



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

N° d'ordre. **SIOD27/SIOD/M2/2022**

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **système d'information, optimisation et décision (SIOD)**

Sélection de services web par l'algorithme d'optimisation de colonie d'abeilles

Par :

SEMAHI IKRAM

Soutenu le 19/06/2023 devant le jury composé de :

Kerdoudi Mohamed Lamine	MCA	Président
Torki Fatima Zohra	MAA	Rapporteur
Moussaoui manel	MAA	Examineur

Année universitaire 2022-2023

Remerciements

Au nom d'Allah, le Miséricordieux, le très Miséricordieux.

Je veux d'abord remercier Dieu qui m'a donné la foi et la volonté, qui m'a mis au monde afin que je puisse ramener une abondance bénéfique dans ma vie.

Je voudrais remercier ceux qui ont été la raison de ma vie, la source de tendresse et l'exemple de dévouement, qui n'ont jamais cessé de m'encourager et de prier pour moi. Ceci est pour toi maman, à mon cher père qui n'a jamais cessé de m'aider et de m'encourager tout au long de ma carrière

J'adresse également mes sincères remerciements à mon encadreur, Mm TORKI Fatima Zohra.

Enfin, je tiens à remercier tous ceux qui m'ont aidé de loin ou directement afin que je puisse faire cet humble travail.

Merci à tous

Dédicace

A mes chers parents, Que nulle dédicace ne puisse exprimer ce que je leurs dois, pour leur bienveillance, leur affection et leur soutien Trésors de bonté, de générosité et de tendresse, en témoignage de mon profond amour et ma grande reconnaissance « Que Dieu vous garde ».

A mes chers frères et ma grande sœur.

Résumé

L'approche par service est devenue la plus utilisée dans le monde du développement software. Ceci a impliqué un nombre important de service web dans le monde. Pour le client choisir le bon service adapté aux besoins est une tâche énorme car il est confronté à l'embarras du choix et il sera incapable de décider quel service est le plus approprié. La seule différence entre des services Web similaires peut être leur qualité de service (QoS).

Notre travail consiste à développer une application basée sur une approche d'optimisation mono-objective globale pour la sélection des meilleurs services Web en fonction de QoS. Les colonies d'abeilles exploitent les relations de voisinages et les heuristiques de danse pour explorer rapidement un grand espace de recherche pour trouver la meilleur service composite. Les résultats obtenus sont très satisfaisants et confirment l'utilité de cette approche dans la réduction de temps de réponse.

Mots clés : SOA, Services Web, Sélection de compositions de services, optimisation, qualité de service (QoS), Meta heuristique, Colonies d'abeilles.

Abstract

The service approach has become the most used in the world of software development. This involved a significant number of web services worldwide. for the consumer to choose the right service suited to the needs is a huge task as he is faced with the embarrassment of choice and he will be unable to decide which service is the most appropriate The only difference between similar web services may be their quality of service (QoS) .

Our work consists in developing an application based on a global mono-objective optimization approach for the selection of the best Web services according to QoS. Bee colonies exploit neighborhood relationships and dance heuristics to quickly explore a large search space. The results obtained are very satisfactory and confirm the usefulness of this approach in reducing response time.

Keywords: SOA, Web services, Selection of service compositions, optimization, quality of service, meta-heuristics, bee colonies.

المخلص

أصبح نهج الخدمة هو الأكثر استخدامًا في عالم تطوير البرمجيات. تضمن هذا عددًا كبيرًا من خدمات الويب في العالم، بالنسبة لمستهلك أصبح اختيار الخدمة الصحيحة هي مهمة ضخمة لأنه يواجه صعوبة في الاختيار ولن يتمكن من تحديد الخدمة الأكثر ملاءمة. يمكن أن يكون الفرق الوحيد بين خدمات الويب المشابهة هو معايير جودة الخدمة.

إن عملنا هذا يقتضي تطبيق مقارنة لتحسين انتقاء تركيبات خدمات الويب ، حيث أن هذه المقارنة مستوحاة من سلوكيات الحيوانات أو الحشرات التي تعيش في مجموعات. حيث أن أكثرها نظامًا وذكاءً جماعيًا هي مجموعات النحل و التي تتميز بقدرتها على التواصل فيما بينها، وقد أظهرت نتائج العمل التطبيقي ذلك.

الكلمات المفتاحية : الهيكلة الخدمانية ، خدمات الويب، انتقاء خدمات الويب، مقارنة تحسينية ، نوعية الخدمات ، مستعمرات النحل.

Liste d'abréviations

IBM : International Business Machines Corporation

W3C : World Wide Web Consortium

URI : Uniform Resource Identifier

SOA : Service Oriented Architecture

XML : eXtensible Markup Language

RPC : Remote Procedure Call

HTTP : Hypertext Transfer Protocol

HTTP POST : Hypertext Transfer Protocol Post

XML_RPC : XML Remote Procedure Call

SOAP : Simple Object Access Protocol

WSDL : Web Services Description Language

UDDI : Universal Description ,Discovery and Integration

IDL : Interface Definition Language

URL : Uniform Resource Locator

SMTP : Simple Mail Transfer Protocol

B2C : Business to Consumer

B2B : Business to Business

C2C : Customer-to-Customer e-commerce

C2B : Customer-to-Business

SW : Service Web

QoS : Quality of de Service (Qualité de service)

ILP : Integer Lineaire Porogramming

MIP : Mixed Integer Programming

API : Application Programming Interface

ABC : Algorithme de colonies d'abeilles artificielles (Artificial Bee Colonies, ABC)

UML : Unified Modeling Language,

CSW : Composition des services Web

JVM : Java Virtual Machine

EDI : Environnement Développement Intégré

CDDL : Common Development and Distribution License

PHP: Hypertext Preprocessor

Html : Hypertext Markup Language

XAMPP : X Apache MySQL Perl PHP

FTP : File Transfer Protocol

SQL : Structured Query Language

Table des matières

Remerciements	i.
Résumé	ii.
Liste des abréviations	iii.
Table des matières	iv.
Liste des figures	v.
Liste des tableaux	vi.
Introduction générale	1
Chapitre 1 : Les Services Web	
1. Introduction.....	4
2. Définition de service Web.....	4
2.1. Service Web (selon IBM)	4
2.2. Service Web (selon W3C (World Wide Web Consortium))	4
3. Architecture orientée service.....	4
4. Technologies des services web.....	5
4.1. XML (eXtensible Markup Language)	6
4.2. SOAP (Simple Object Access Protocol)	6
4.3. UDDI (Universal Description, Discovery and Integration)	6
4.4. WSDL (Web Services Description Language)	6
5. Fonctionnement et architecture des services web.....	7
5.1. Fonctionnement des services web.....	7
5.2. Architecture en couche de service web.....	8
5.2.1. La couche de publication.....	9
5.2.2. La couche description.....	9
5.2.3. La couche communication.....	9
5.2.4. La couche transport.....	9
6. Composition et services web composites.....	10
6.1. Définition de composition de service Web.....	10
6.2. Classification de composition des services web.....	10
6.3. Types de composition de services Web.....	11
6.3.1. Orchestration.....	12
6.3.2. Chorégraphie.....	12

7. Quelques domaines d'application des Services Web.....	12
8. Les avantages et inconvénients des services web.....	13
8.1. Les avantages des services Web.....	13
8.2. Les inconvénients des services web.....	13
9. Conclusion.....	14

Chapitre 2 : La Sélection des Services Web

1. Introduction.....	16
2. Propriétés utilisées pour les méthodes de sélection des services web.....	16
2.1. Propriétés fonctionnelles.....	16
2.2. Propriétés non fonctionnelles.....	16
3. Qualité de Service (QoS)	16
4. Motivation de la sélection des services Web.....	17
5. Techniques de sélection de services web.....	18
5.1. Sélection mono-objective.....	19
5.1.1. Sélection locale.....	19
5.1.2. Sélection globale.....	20
5.1.3. Sélection Hybride.....	19
5.2. Sélection multi-objectives.....	19
5.3. Sélection mono et multi-objective.....	20
6. Conclusion.....	20

Chapitre 3 : Conception du système

1. Introduction.....	24
2. Conception du système.....	24
2.1. Architecture générale du système.....	25
2.2. Exemple illustratif et formulation.....	25
2.3. Architecture détaillée du système	26
2.3.1. Données d'entrée du système (la base de services)	26
2.3.2. Description de la requête.....	27
2.3.3. Critères de QoS considérés	27
2.3.4. Calcul de la qualité de composition de service web (SW).....	28

2.3.4.1. Fonctions d'agrégation.....	28
2.3.4.2. Fonctions objective.....	29
3. Optimisation par Colonie d'abeilles	30
3.1. Algorithme de colonies des abeilles artificielles (ABC)	30
3.2. Caractéristiques de l'algorithme ABC	30
3.3. Principe général de l'algorithme ABC.....	31
4. Modélisation UML.....	32
4.1.Diagramme de cas d'utilisation.....	32
4.2.Diagramme de classes.....	33
5. Conclusion.....	34

Chapitre 4 : Implémentation du système

1. Introduction.....	36
2. Outils et environnement de développement.....	36
2.1. Le Langage JAVA.....	37
2.2. L'IDE NetBeans.....	37
2.3. Gestion de base de données (phpMyAdmin)	38
3. Présentation des interfaces de notre système.....	40
3.1. Interface d'accueil.....	40
3.2.Interface du chargement de la base de services.....	41
3.3.Interface de contraintes QoS.....	43
3.4. Interface de sélection des services Web (SW)	44
4 . Conclusion.....	48
Conclusion Générale et Perspectives.....	49
Bibliographie.....	50

Liste des figures

Figure 1.1 : Modèle fonctionnel de l'architecture SOA [4].	5
Figure 1.2 : Architecture des composants des services web	6
Figure 1.3 : Fonctionnement d'un Service Web [6]	7
Figure 1.4 : La pile des couches des Services Web.....	9
Figure 2.1 : Techniques de sélection de services web à base de QoS.....	19
Figure 3.1 : Exemple de motivation	24
Figure 3.2 : Architecture générale du système.....	25
Figure 3.3 : Pseudo code de l'algorithme ABC.....	32
Figure 3.4 : Diagramme de cas d'utilisation.....	33
Figure 3.5 : Diagramme de classes.....	34
Figure 4.1 : Environnement logiciel utilisé.....	36
Figure 4.2 : Langage JAVA.....	37
Figure 4.3 : Interface principale de NetBeans.....	38
Figure 4.4 : XAMPP.....	39
Figure 4.5 : Interface Serveur phpMyAdmin.....	40
Figure 4.6 : Interface d'accueil.....	41
Figure 4.7 : la Requête demandée par l'utilisateur.....	42
Figure 4.8 : Liste des services candidats de la classe reserverhotel.....	43
Figure 4.9 : Message d'erreur de connexion à la base de données.....	43
Figure 4.10 : Contraintes globales de qualités de services.....	44
Figure 4.11 : La Sélection.....	44
Figure 4.12 : Sélection le meilleur service.....	45
Figure 4.13 : Résultat de l'algorithme ABC pour la classe reserverhotel.....	46
Figure 4.14 : Sélection le meilleur service de la classe reservavion.....	47
Figure 4.15 : Résultat de l'algorithme ABC pour reservavion.....	47
Figure 4.16 : Sélection le meilleur service de classe reservevoiture.....	48

Liste des Tableaux

Tableau 1.1 : Description en couche des services Web.....	8
Tableau 3.1 : Les services Web de la classe reserverhotel	27
Tableau 3.2 : Les fonctions d'agrégation des critères de QoS	29

Introduction générale

Les dernières décennies ont été marquées par le développement rapide des systèmes d'information distribués, et la diffusion de l'accès à Internet. Cette évolution du monde informatique a entraîné le développement de nouveaux paradigmes d'interaction entre les applications. Un de ces paradigmes qui a pris de l'ampleur au cours de ces dernières années est l'architecture orientée services (SOA).

L'architecture orientée service est basée sur l'utilisation de composants réutilisables et distribués appelés *services web*. Le concept de "**Web services**" regroupe un ensemble de technologies basées sur XML, permettant de créer des composants logiciels distribués, de décrire leurs interfaces et de les utiliser en mettant en œuvre des standards simples tels que SOAP, WSDL, UDDI. En effet, On passe d'une architecture à base de composants à une architecture de systèmes d'information orientée services. Cela facilite l'intégration d'applications dans l'entreprise et les échanges B2B (Business to Business) ou B2C (Business to Customer) via le web.

Les services web sont des systèmes logiciels qui répondent généralement à des besoins fonctionnels. Les contraintes temps réel expriment des exigences en termes de temps sur le comportement de l'application. Ces exigences sont totalement indépendantes des besoins fonctionnels de l'application. Ainsi les contraintes temps réel seront exprimées à travers des besoins non fonctionnels. Les besoins non fonctionnels des services web s'expriment généralement à travers la qualité de service (QoS) [18].

La QoS est un ensemble de propriétés et caractéristiques d'une entité ou d'un service qui lui confèrent l'aptitude à satisfaire des 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 les contraintes de temps imposées.

Nous nous intéressons dans ce mémoire plus particulièrement à la phase de sélection des services qui permet de fournir une composition de services Web. La sélection à base de Qualités de Services consiste à choisir parmi les services Web découverts de chaque tâche, ceux qui répondent au mieux aux exigences de l'utilisateur sur la base des besoins non fonctionnels de QoS. En général, trouver un plan de configuration qui répond aux exigences de qualité de service du client est un problème d'optimisation qui prend beaucoup de temps, ce qui nous oblige à adopter une méthode méta heuristique pour choisir le meilleur.

La méta-heuristique est un algorithme avancé pour résoudre les problèmes d'optimisation. Elle est définie officiellement comme un processus de génération itérative qui dirige de manière guidée à travers l'intelligence combinée de différentes stratégies d'exploration et d'exploitation de l'espace de recherche, en utilisant des stratégies d'apprentissage pour structurer les informations afin de trouver des solutions quasi-optimales de manière efficace.

Dans ce travail, on propose une approche d'optimisation pour sélectionner des meilleurs services Web composites à base de colonies d'abeilles orientée qualité de service. L'algorithme d'abeille est une classe d'inférences descriptives récemment proposée pour des problèmes d'optimisation difficiles, et ont donné la majorité des problèmes résolus par ces algorithmes de très bons résultats en termes de valeur de la fonction objectif et de temps d'exécution qui était acceptable.

Notre mémoire est organisé, en quatre chapitres, comme suit :

Chapitre I : (les services web) : présente des concepts fondamentaux pour les services web, technologies utilisées dans les services Web, leurs avantages et inconvénients, ainsi que nous présentons quelques domaines d'application, et ensuite nous donnons des notions sur la qualité de service (QoS ou QoS en anglais : Qualité of Service).

Chapitre II : (la sélection des services Web) : ce chapitre présente un aperçu général sur des approches existantes dans la littérature pour la sélection des services web.

Chapitre III : (Conception) : ce chapitre décrit la conception de notre système représentée en deux niveaux (la conception globale et détaillée) et ainsi que la modélisation UML.

Chapitre IV : (Implémentation) : ce chapitre est destiné à l'évaluation du modèle proposé en conception. Nous commençons par la description des outils utilisés, ainsi que les interfaces et les résultats obtenus.

Enfin nous terminons ce mémoire par une conclusion générale et des perspectives attendues.

Chapitre 1 : Les Services Web

1. Introduction

Les services Web sont des technologies émergentes et prometteuses pour le développement, le déploiement et l'intégration des applications Internet.

Dans ce chapitre, nous décrivons les Services Web en mettant en évidence la définition, Service-Orienté Architecture, les technologies des services web, fonctionnement et architecture des services Web, composition et services web composites.

2. Définition de service Web

On retrouve plusieurs définitions des services Web :

2.1. Service Web (selon IBM)

Les services Web sont la nouvelle génération des applications web. Ce sont des applications auto-contenues, auto-descriptives et modulaires qui peuvent être publiées, localisées et invoquées depuis le web. Les services Web effectuent des actions allant de simples requêtes à des processus métiers complexes. Une fois qu'un service web est déployé, d'autres applications (y compris des services Web) peuvent le découvrir et l'invoque [1].

2.2. Service Web (selon W3C (World Wide Web Consortium))

Un service Web est un composant logiciel identifié par une URI (Uniform Resource Identifier), et conçu pour supporter l'interaction interopérable de machine à machine sur un réseau. Il possède une interface décrite dans un format exploitable par la machine (WSDL : Web Service Description Language) [2].

3. Architecture orientée service

Une architecture orientée service (Service Oriented Architecture ou SOA en anglais) est un Framework stratégique de technologie qui permet à tous les systèmes intéressés, à l'intérieure à l'extérieur d'une organisation, d'exposer et d'accéder à des services bien définis, et à des informations liées à ces services, qui peuvent être encore abstraites pour traiter les couches et les applications composées pour le développement de solutions. En substance, SOA ajoute l'aspect d'agilité à l'architecture, ce qui nous permet de faire face aux changements dans les systèmes en utilisant une couche de configuration plutôt que d'avoir constamment à redévelopper ces systèmes [3].

Chaque bloc de construction SOA peut jouer un ou plusieurs des rôles suivants [4] :

- **Fournisseur de services** : il désigne l'entité propriétaire du service. D'un point de vue technique, un fournisseur est constitué par la plate-forme d'accueil du service.
- **L'annuaire de services** : correspond à un registre de descriptions de services offrant des facilités de publication de services à l'intention des fournisseurs ; et des facilités de recherche de services à l'intention des clients. En d'autres termes, l'annuaire joue le rôle d'intermédiaire entre les clients et les fournisseurs de services
- **Demandeur de services** : c'est le consommateur de service. D'un point de vue technique, le service client est constitué de l'application qui va rechercher et invoquer un service.

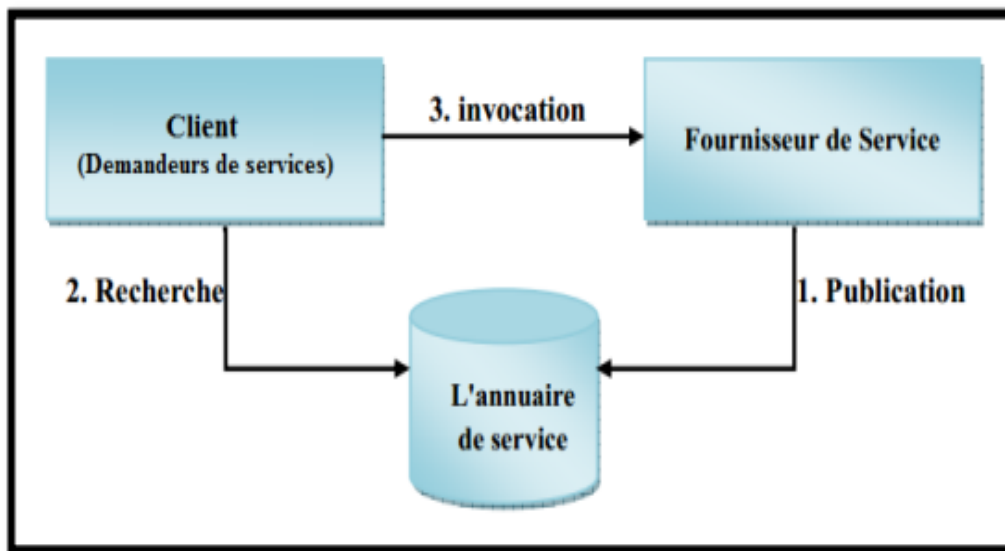


Figure 1.1 : Modèle fonctionnel de l'architecture SOA.[4]

4. Technologies des services web

XML, SOAP, WSDL, et UDDI sont les technologies dominantes des services web. Une pléthore d'autres technologies viendront aux fils du temps pour enrichir l'architecture de ces derniers. Sans entrer dans les détails techniques, nous vous exposons dans le paragraphe suivant les grandes lignes de ces technologies.

On présente les technologies utilisées pour les services web illustré dans la Figure 1.2 comme suit [5] :

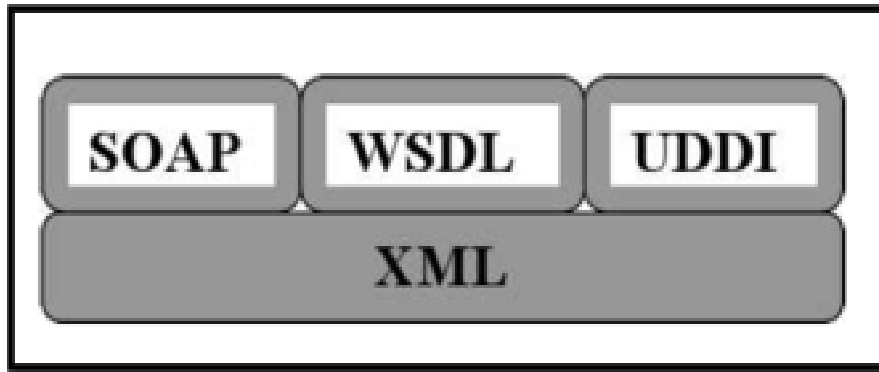


Figure 1.2 : Architecture des composants des services web.

4.1. XML (eXtensible Markup Language)

XML 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.

4.2. SOAP (Simple Object Access Protocol)

- Est un protocole d'échange inter-applications indépendant de toute plate-forme, basé sur le langage XML
- Un appel de service SOAP est un flux ASCII encadré dans des balises XML et transporté dans le protocole http.

4.3. UDDI (Universal Description, Discovery and Integration)

- Normalise une solution d'annuaire distribué de web services, permettant à la fois la publication et l'exploration (recherche) de web services.
- UDDI se comporte lui-même comme un web service dont les méthodes sont appelées via le protocole SOAP.

4.4. WSDL (Web Services Description Language)

- Donne la description au format XML des web services en précisant les méthodes pouvant être invoquées, leurs signatures et le point d'accès (URL, Port, etc...).
- C'est, en quelque sorte, l'équivalent du langage IDL pour la programmation distribuée CORBA.

5. Fonctionnement et architecture des services web

5.1. Fonctionnement des services web

Le fonctionnement des services Web s'articule autour de trois acteurs principaux illustrés par le schéma suivant [6] :

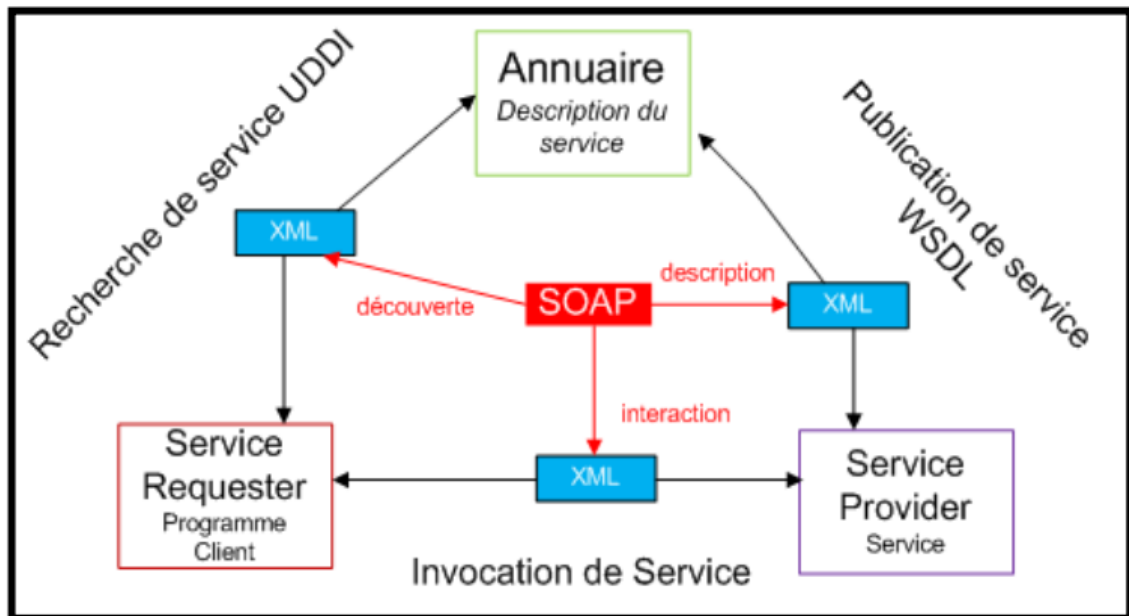


Figure 1.3 : Fonctionnement d'un Service Web [6].

- **Service provider service** : Le fournisseur de service met en application le service Web et le rend disponible sur Internet.
- **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 (REST, XML-RPC, SOAP).
- **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 [7] :

- **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 l'utilise pour l'invoquer auprès du fournisseur de services.

5.2.Architecture en couche des services web

L'exposition et l'utilisation des services se fait dans un contexte particulier qui définit clairement les interactions entre le service et ses utilisateurs. Les services web communiquent via un ensemble de technologies fondamentales qui partagent une architecture commune. Ils ont été conçus pour être réalisés sur de nombreux systèmes développés et déployés de façon indépendante.

Une architecture standard est constituée de plusieurs couches se superposant les unes sur les autres, d'où le nom de pile des Web services. La Figure 1.4 décrit un exemple d'une pile qui 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 illustrer dans le Tableau 1.1.

Couche	Technologie	Rôle
Découverte de services	UDDI	La publication et la découverte des services web sont assurées par le biais du référentiel UDDI, un référentiel UDDI est un catalogue de services Web.
Description de services	WSDL	La description d'un service Web se fait en utilisant le langage WSDL. Il expose l'interface du service.
Communication	SOAP	SOAP prévoit la couche de communication basée sur XML pour accéder à des services Web.
Transport	HTTP	Le transport de messages SOAP est assuré par le standard HTTP.

Tableau 1.1 : Description en couche des services Web.

Les différentes couches de l'architecture d'un service web s'interfacent avec des standards, comme suit [8] :

5.2.1. La couche de publication

Cette couche repose sur le protocole UDDI (Universal Description, Discovery and Integration) qui assure le regroupement, le stockage et la diffusion des descriptions des services web.

5.2.2. La couche description

Cette couche est prise en charge par le langage WSDL (Web Service Description Language) qui décrit les fonctionnalités fournies par le service web, les messages reçus et envoyés pour chaque fonctionnalité, ainsi que le protocole utilisé pour la communication.

5.2.3. La couche communication

La couche de communication des messages propose différents mécanismes liés à l'acheminement des messages (format de communication des messages, adressage, routage...etc.). Cette couche utilise des protocoles reposants sur le langage XML, qui résout les conflits syntaxiques lors de l'encodage des données et cela grâce à sa syntaxe unique. Actuellement SOAP (Simple Object Access Protocol) est le protocole le plus utilisé pour cette couche.

5.2.4. La couche transport

Le protocole le plus utilisé dans cette couche est l'HTTP (Hyper Text Transfert Protocol). Cependant, d'autres protocoles peuvent être utilisés, tels que le SMTP (Simple Mail Transfer Protocol) ou le FTP (File Transfer Protocol), permettant ainsi aux services web de rester indépendants du mode de transport utilisé.

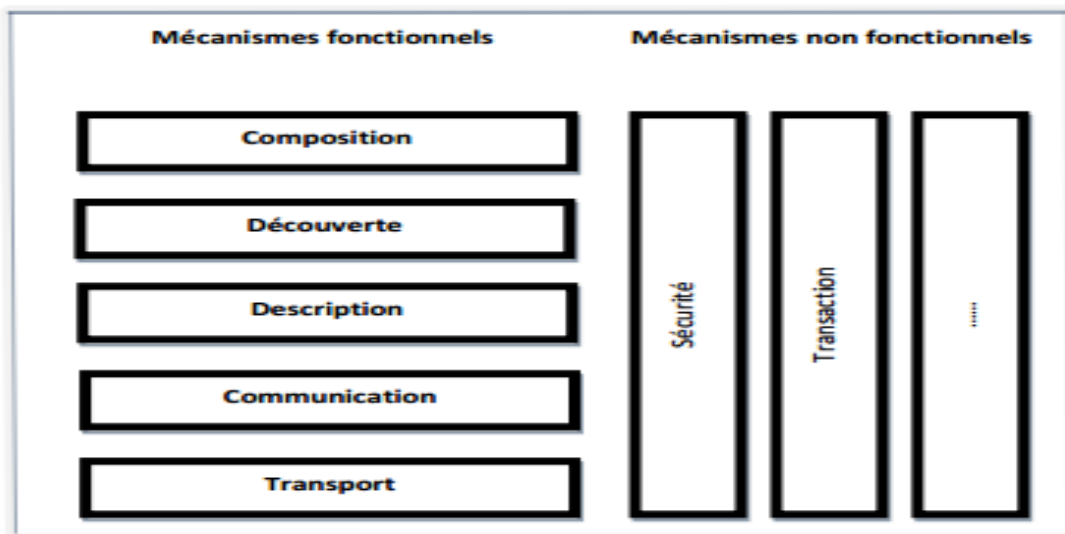


Figure 1.4 : La pile des couches des Services Web

6. Composition et services web composites

La composition de services Web est la plus importante fonctionnalité assurée par une architecture SOA. Celle-ci offre un environnement homogène pour la composition dans la mesure où toutes les parties de la composition sont des services idéalement décrits de la même façon et communiquant par les mêmes standards d'échange de messages [9].

D façon générale, la composition de services Web est défini comme étant un processus complexe qui consiste à trouver des composants appropriés, de les sélectionner, de les combiner, et de fournir des informations d'ordre comportemental renseignant sur leur orchestration ou chorégraphie en vue de les rendre exécutables. Le résultat de cette composition est un service web composite faisant interagir un ensemble de services web afin de satisfaire la requête de l'utilisateur qui ne peut l'être par un service atomique [10].

6.1. Définition de composition de service Web

La composition des services Web est le processus de construction de nouveaux services Web à valeur ajoutée, à partir de deux ou plusieurs services Web déjà présents et publiés sur le Web. Un service Web est dit composé ou composite lorsque son exécution implique des interactions avec d'autres services Web, et des changements des messages entre eux afin de faire appel à leurs fonctionnalités.

La composition de services Web spécifie quels services ont besoin d'être invoqués, dans quel ordre et comment gérer les conditions d'interaction [11].

6.2. Classification de composition des services web

L'idée de la composition de services web consiste à définir comment les services web vont être rassemblés selon certaines règles, pour atteindre le but demandé par un utilisateur. Une fois que, la description de la composition est réalisée, il est possible de savoir facilement quels services web appartiendront à cette composition. Les solutions proposées peuvent être classifiées selon deux axes :

En fonction du degré de participation de l'utilisateur dans la définition du schéma de composition, ces propositions sont **manuelles**, **semi-automatiques** ou **automatiques** [11].

- **Composition manuelle**

La composition manuelle des services web suppose que l'utilisateur gère la composition à la main via un éditeur de texte et sans l'aide d'outils dédiés.

- **Composition semi-automatique**

Les techniques de composition semi-automatiques sont un pas en avant en comparaison avec la composition manuelle, dans le sens qu'elles font des suggestions sémantiques pour aider à la sélection des services web dans le processus de composition.

- **Composition automatique**

La composition totalement automatisée prend en charge tout le processus de composition et le réalise automatiquement, sans qu'aucune intervention de l'utilisateur ne soit requise. Selon que, la sélection des services web et la gestion du flot soient faites ou non a priori, une autre approche sera dite **statique** ou **dynamique** :

- **Composition statique des services web**

Composition statique qui consiste à générer manuellement les flux de composition. Dans une composition statique, l'agrégation des services a lieu au moment de la conception. Les composants de service requis pour la composition sont choisis, liés ensemble, puis déployés. La composition statique n'est pas flexible et adaptable dans les cas où il y a des changements fréquents d'exigences ou des services qui ne peuvent pas être prévus au moment de la conception. Une fois qu'un service de liaison anticipée devient indisponible, ou s'il existe un meilleur service alternatif, la composition statique ne sera pas en mesure de fournir un meilleur support pour l'exécution du service composite en temps réel [12].

- **Composition dynamique des services web**

Dans cette approche, les services sont sélectionnés et composés à la volée en fonction des besoins formulés par l'utilisateur ce type de composition pour offrir le potentiel de réaliser des applications flexibles et adaptables en sélectionnant et en combinant les services de manière appropriée sur la base de la requête et du contexte de l'utilisateur. Ce type de composition peut engendrer de nombreuses applications utiles, qui n'ont pas été prévues à l'étape de conception. Par conséquent, la composition dynamique de services Web est propice dans un environnement, tel que le Web et l'informatique pervasive où les composants disponibles sont dynamiques et les attentes des utilisateurs sont variables et personnalisées [11].

6.3.Types de composition de services Web

La plupart des travaux portant sur la composition de services Web reconnaissent deux types de composition : l'orchestration et la chorégraphie de services.

6.3.1. Orchestration

Définissent l'orchestration comme un ensemble de processus exécutés dans un ordre prédéfini afin de répondre à un but. Ce type de composition permet de centraliser l'invocation des services Web composants. Chaque service est décrit en termes d'actions internes. Les contrats entre deux services sont constitués selon le processus à exécuter. L'orchestration de services Web consiste en la programmation d'un moteur qui appelle un ensemble de services Web selon un processus prédéfini. Ce moteur définit le processus dans son ensemble et appelle les services Web (tant internes qu'externes à l'organisation) selon l'ordre des tâches d'exécution. L'orchestration peut être vue comme une composition ascendante : les services Web utilisés dans la composition existent au préalable et sont appelés selon un enchaînement prédéfini afin de réaliser un processus précis [12].

6.3.2. Chorégraphie

La chorégraphie permet de décrire la composition comme un moyen d'atteindre un but commun en utilisant un ensemble de services Web. La collaboration entre chaque service Web de la collection (faisant partie de la composition) est décrite par des flots de contrôle. Pour concevoir une chorégraphie, les interactions entre les différents services doivent être décrites. La logique de contrôle est supervisée par chacun des services intervenant dans la composition. L'exécution du processus est alors distribuée [12].

7. Quelques domaines d'application des Services Web

- Les services Web peuvent être utiles dans la plupart des scénarios applicatifs lorsque la communication peut être établie sur un modèle bidirectionnel (requête/réponse).
- L'application des services Web est multiple, autant dans les domaines du B2B, B2C que pour les domaines de gestion de stock, 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 professionnel à professionnel [13].
- *Customer-to-Customer* e-commerce (C2C) qui se réfère aux activités commerciales entre plusieurs clients à travers des sites - parties-tiers - (ex:ebay.com) [13].
- Customer-to-Business (C2B) qui se réfère, quant à lui, à un regroupement de clients intéressés par l'achat d'une entreprise. Les clients ont, dans ce cas, ou bien un intérêt

économique commun (ex: demande d'achat de Mercata.com) ou encore un intérêt commun social (ex: le groupe d'avocat intéressé par Voxcap.com) [13].

8. Les avantages et inconvénients des services web

8.1. Les avantages des services Web

Aujourd'hui, les services Web jouent un rôle majeur dans les secteurs d'administration de serveur et de développement car ils offrent beaucoup d'avantages tels que [14] :

- ✓ Les services Web permettent des programmes écrits dans différentes langues et sur des plateformes différentes de communiquer entre elles en utilisant certaines normes. En d'autres termes, les services Web permettent une meilleure interopérabilité entre les logiciels.
- ✓ Les services Web utilisent des normes et des protocoles ouverts tel que SOAP et HTTP.
- ✓ Basés sur le protocole HTTP, les services Web peuvent fonctionner via de nombreux pare-feu sans nécessiter de modifications des règles de filtrage.
- ✓ Les outils de développement, basés sur ces standards, permettent la création automatique de programmes utilisant les services web existants.
- ✓ Grâce aux services web, les coûts sont réduits par l'automatisation interne et externe des processus commerciaux.
- ✓ Les services sont conçus de façon à ce qu'ils puissent être réutilisés ultérieurement pour minimiser la redondance et rendre le service réutilisable par les différentes applications du système d'information.

8.2. Les inconvénients des services web

La technologie des services web comporte plusieurs inconvénients dont [13] :

- Problèmes de performance : Les services web sont encore relativement faibles par rapport à d'autres approches de l'informatique répartie telles que CORBA ou RMI.
- Fiabilité : Il est difficile de s'assurer de la fiabilité d'un service car on ne peut garantir que ses fournisseurs ainsi que les personnes qui l'invoquent travaillent d'une façon fiable

- Syntaxe et sémantique : On se concentre beaucoup sur comment invoquer des services (syntaxe) et pas assez sur ce que les services web offrent (sémantique)
- Disponibilité : Les services web peuvent bien satisfaire un ou plusieurs besoins du client. Seront-ils pour autant toujours disponibles et utilisables ? Ça reste un défi pour les services web.
- Confiance : Les relations de confiance entre différentes composantes d'un service web sont difficiles à bâtir, puisque parfois ces mêmes composantes ne se connaissent même pas.
- Technologies connexes : SOAP, technologie primordiale dans le domaine des services web, manque de mécanismes d'authentification, d'autorisation et de chiffrement.

9. Conclusion

Dans ce chapitre nous avons fourni une vue générale des services web en termes de concept, de fonctionnalité et de technologie, et nous avons présenté une vue de composition et services web composites et dans ce cœur nous avons défini et décrit la définition et la classification de composition des services web et leur type.

Chapitre 2 : Sélection des services Web

1. Introduction

Etant donné qu'il y a plusieurs services Web publiés sur Internet, beaucoup d'entre eux, ne peuvent satisfaire les différents besoins d'un utilisateur. Il sera donc nécessaire, d'ordonner et de classer plusieurs services Web en se basant sur différents critères de qualité. En effet, en disposant d'une bibliothèque de services Web similaires, nous pouvons faire une sélection à la main, en faisant l'écoute sur les services les plus populaire, ou guidée par un outil d'aide à la décision. La sélection des services Web a pour objectif de déterminer le service le plus adéquat.

Dans ce chapitre nous allons voir les propriétés utilisées pour les méthodes de sélection de services web et dans cœur nous allons définir la qualité de service (QoS) et ses paramètres, ensuite nous allons présenter les différents techniques de sélection de services web avec les 03 grandes classes : la sélection mono objective, la sélection multi objective, et la sélection hybride.

2. Propriétés utilisées pour les méthodes de sélection des services web

2.1. Propriétés fonctionnelles

Les propriétés fonctionnelles d'un service Web désignent les opérations qu'il peut fournir. Les propriétés fonctionnelles sont décrites dans la description de service en termes d'opérations et reflètent le fonctionnement du service [15].

2.2. Propriétés non fonctionnelles

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

3. Qualité de Service (QoS)

La QoS est un ensemble de propriétés et caractéristiques d'une entité ou d'un service qui lui confèrent l'aptitude à satisfaire des 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 les contraintes de temps imposées [14].

De nombreux paramètres ont été ainsi proposés dans la littérature. Les paramètres de QoS les plus importants sont les suivants [16] :

- **Le coût (ou le prix) :** désigne le prix que le client doit payer pour chaque invocation du service. Ce paramètre apparaît fréquemment dans les services Web commerciaux

- **La latence (aussi connue comme le temps de réponse) :** ce paramètre est utilisé pour désigner le temps pris par un service pour répondre à une requête. Cela peut comprendre le retard causé par le réseau lors de l'appel au service Web.
- **Le débit :** le débit d'un service représente le nombre de demandes que le service est en mesure de traiter dans un intervalle de temps donné. Il peut inclure le débit maximal ou une fonction décrivant comment le débit varie en fonction de l'intensité de charge.
- **La fiabilité (aussi connue comme le taux d'exécution avec succès) :** ce paramètre désigne la capacité du service à accomplir sa fonction correctement pendant une période de temps spécifiée. Il peut être mesuré par le temps moyen entre pannes.
- **La Disponibilité :** La probabilité d'accessibilité d'un service.
- **La sécurité :** la sécurité d'un service Web comporte différents aspects assurant que les échanges de messages entre le client et le service soient sécurisés. Plus précisément c'est un regroupement d'un ensemble de qualités à savoir : la confidentialité, le cryptage des messages et le contrôle d'accès.
- **La réputation :** c'est une mesure de la crédibilité du service. Elle dépend principalement des expériences d'utilisateurs finaux. Ces paramètres peuvent être classés, selon leur mesure, en trois catégories.

4. Motivation de la sélection des services web

La disponibilité des fournisseurs de services avec des caractéristiques différentes rend la tâche de choisir un fournisseur de service approprié pour un utilisateur de plus en plus complexe qui nous motive à considérer de nouvelles solutions pour le problème de la sélection des services [36].

- Il existe différents types de services (calcul, stockage, etc.).
- Il existe un grand nombre de services avec fonctionnalités similaires qui se traduit par une prolifération du nombre de services qui offrent des fonctionnalités similaires avec QoS différents critères (prix, disponibilité).
- Il existe un grand nombre de fournisseurs de services qui sont émergente en continu sur Internet tels que Google App Engine.
- Enfin, il y a un large éventail entre la performance de service et le prix. Lorsque différents fournisseurs offrent leurs services avec les prix et les valeurs de performance différentes

On peut conclure que le processus de la sélection de service Web a besoin de cinq issues cruciales :

- (a) Précision : l'algorithme devrait éviter la perte de Web services qui peuvent correspondre à la demande de l'utilisateur, mais leur interface est pas la même que la demande de l'utilisateur.
- (b) Flexibilité : de nouveaux mécanismes évolutifs devraient être flexible pour supporter un grand nombre de fournisseurs de services.
- (c) Evolutivité : l'algorithme de sélection devrait être évolutive pour soutenir un certain nombre de QoS exigences.
- (d) Généralités : l'algorithme de sélection devrait être aussi générique que possible pour soutenir les différents utilisateurs et divers exigences des utilisateurs, plutôt que de types spécifiques d'utilisateurs.
- (e) la personnalisation de l'utilisateur : l'algorithme doit être capable de fournir le bon service à la demande des utilisateurs; l'idéale préférence des utilisateurs doivent être automatiquement capturées.

5. Techniques de sélection de services web

La sélection automatique de services web, a fait l'objet de plusieurs travaux, de façon générale on distingue 03 grandes classes : la sélection mono objective, La sélection multi objective, et la sélection hybride (mono et multi objective), ces classes forment une hiérarchie comme montrée sur la figure 2.1, par la suite on va détailler chacune des approches en donnant quelques algorithmes utilisés par ces dernières :

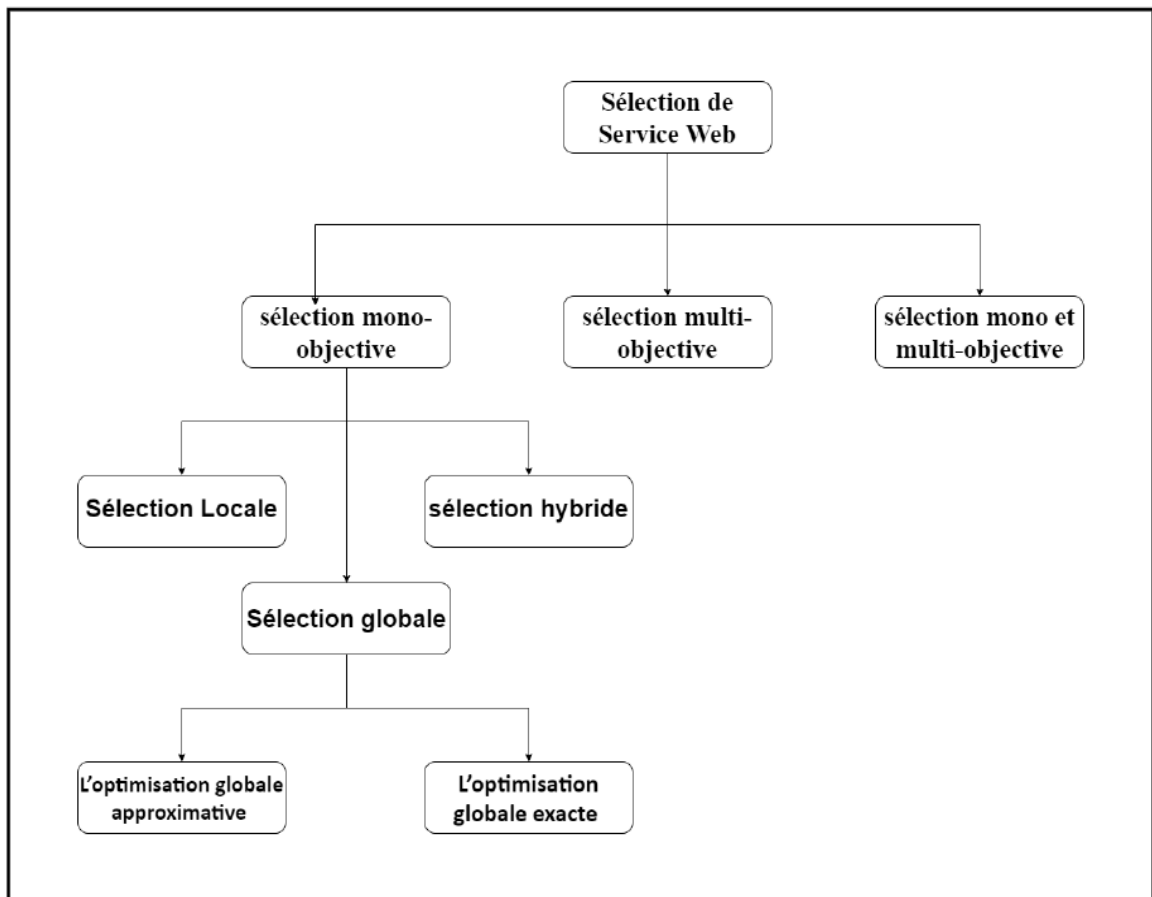


Figure 2.1 : Techniques de sélection de services web à base de QoS.

5.1. Sélection mono-objective

Cette catégorie suppose que les m valeurs de qualité de service sont agrégées en un seul score, i.e. on considère une seule fonction objective qui associe des poids aux différents attributs de QoS. Elle est divisée en 03 sous classes : la sélection locale, globale et hybride (globale et locale) [17] :

5.1.1. Sélection locale

A pour objectif de choisir le meilleur service web pour chaque tâche individuelle à part entière en considérant des contraintes de QoS relatives à chaque tâche plutôt qu'en considérant des contraintes de QoS globales exprimées pour l'ensemble des tâches. Pour le cas d'une application temps réel, il s'agit de sélectionner pour chaque tâche un service web apte à l'exécuter en prenant en compte les contraintes temps réel locales et d'autres préférences imposées pour chaque tâche. Par exemple, la durée d'exécution d'une tâche devra avoir un temps de réponse qui ne dépasse pas quelques minutes, le service web exécutant une tâche doit être disponible à 100%, etc [18].

5.1.2. Sélection globale

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 [18]. D'après la Figure 2.1, on distingue deux sous classes d'optimisation globale : exacte et approximative.

5.1.3. Sélection Hybride

Cette approche est un compromis des deux précédentes approches, en commençant la recherche par une optimisation globale, puis continuant le travail avec une optimisation local, sa complexité temporelle est inférieure à celle de l'optimisation globale, l'approche peut également manipuler des contraintes globales [19].

5.2. Sélection multi-objectives

Un grand nombre d'approches existent pour résoudre les problèmes d'optimisation multi-objective. Certains d'entre eux utilisent la connaissance qu'ils ont au sujet du problème pour donner des préférences à quelques objectifs, de ce fait dévient l'aspect multi-objective. D'autres donnent à tous les objectifs le même niveau d'importance,... Parmi ces approches, nous devrions distinguer deux catégories : approches non-Pareto et Pareto. Les approches de Non-Pareto ne traitent pas réellement le problème comme problème multi-objective. Elles essayent de le convertir en problème mono-objectif. D'autre part, les approches de Pareto ne transforment pas les objectifs du problème, mais essayent de les optimiser simultanément [19].

5.3. Sélection mono et multi-objectives

C'est une approche hybride qui combine les deux sélections mono et multi-objective, telle que elle effectue une recherche (ou filtrage) multi objective pour chaque classe, après elle groupe hiérarchiquement les services de chaque classe en choisissant un représentant de ces groupes, ensuite elle continue avec une recherche mono-objective (par niveau) sur les représentants des groupes (ex. la sélection clonale) [20].

6. Problème d'optimisation

L'optimisation est une branche des mathématiques qui permet de résoudre des problèmes en déterminant le meilleur élément d'un ensemble selon certains critères

prédéfinis. De ce fait, l'optimisation est omniprésente dans tous les domaines et évolue sans cesse depuis Euclide. 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 [36].

6.1. Approches d'optimisation

Une méthode d'optimisation permet de trouver la solution optimale, différentes méthodes d'optimisation existent et en sont présentées. Les méthodes d'optimisation peuvent être réparties en deux catégories : Méthodes Approximatives et Méthodes Exactes [36].

6.1.1. Approches Exactes

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

6.1.2. Approche heuristique (Approximative)

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 heuristiques prennent beaucoup plus moins de temps pour trouver la solution optimale par rapport au Méthode exacte.

6.1.3. Approche Méta-heuristique

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).

6.1.3.1. 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'itération s'en 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 classe deux familles d'algorithmes de recherche locale : Pour une optimisation locale : recherche locale simple (la descente). Pour une optimisation globale : Recuit simulé.

6.1.3.2. Méta-heuristiques à base de population de solution

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

Catégories de cette classe :

1. Les algorithmes évolutionnaires.
2. Les algorithmes génétiques.
3. Algorithmes à base d'intelligence par essaims : Colonies de fourmis, Colonies d'Abeilles Artificielles (Artificial Bee Colonies, ABC). Dans notre mémoire on va l'utiliser pour résoudre notre problématique comme une méthode d'optimisation et de sélection de Top services composite.

7. Conclusion

Nous avons présenté dans ce chapitre, les propriétés utilisées pour les méthodes de sélection de services web. Nous avons parlé aussi sur les techniques de sélection de services web. Plus précisément, nous avons étudié la sélection mono objective avec toutes ses variantes la sélection multi objective, et la sélection hybride.

Le prochain chapitre présente une approche pour la sélection de services web basée sur l'algorithme des abeilles artificielles.

Chapitre 3 : Conception du système

1. Introduction

Le nombre de services sur internet devient de plus en plus important ce qui nous a invités à concevoir des approches de sélection plus efficaces, surtout pour les requêtes composées. En général, nous pouvons trouver un ensemble de services qui fournissent la même fonctionnalité mais différent en terme de qualité, dans telle situation que nous devons choisir les meilleurs services, on a besoin d'appliquer un certain algorithme d'optimisation.

Pour atteindre cet objectif et pour obtenir la meilleure performance possible, nous avons suivi la méthodologie présentée au-dessous, présentant la conception de notre système en commençant par sa conception générale puis sa conception détaillée. Puis nous donnons les différents diagrammes UML.

2. Conception du système

2.1. Exemple illustratif et formulation

On suppose qu'il y a un utilisateur qui veut planifier un voyage, pour cela il a besoin de consommer 03 types de services, une réservation d'hôtel, une réservation de billet d'avion et une location de voiture, on note aussi qu'on doit sélectionner un seul service (ou entreprise) de chaque catégorie, en utilisant les critères de QoS (réputation, fiabilité, coûts, temps d'exécution...), en plus le client exige des contraintes globales sur chaque critère de QoS, par exemple le coût total des 03 services, ne doit pas excéder une certaine limite [].

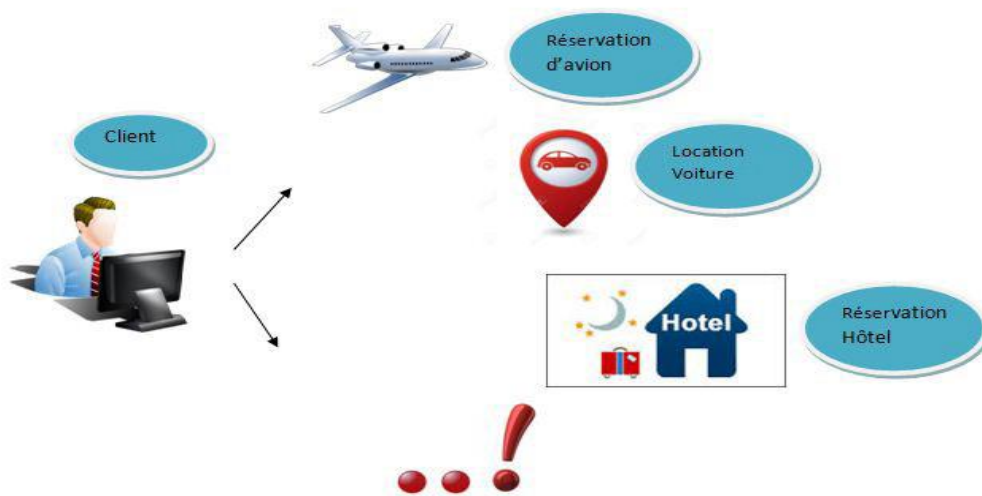


Figure 3.1 : Exemple de motivation.

De façon plus formelle, nous modélisons le problème comme suit :

$$CA = \{S1, \dots, Sn\} \quad (2.1)$$

Une composition abstraite qui représente la requête de l'utilisateur, i.e., les n classes de services à consommer Si .

$$CONS = \{c1, \dots, cm\} \quad (3.1)$$

Un ensemble de contraintes globales définies par l'utilisateur.

$$C = \{s1, \dots, sn\} \quad (3.2)$$

Une composition concrète, i.e., nous remplaçons chaque classe Si par un service concret $si \in Si$.

Nous devons rechercher une composition concrète C telle que la ou les fonctions objectives. Si on agrège les m attributs de QoS en une seule valeur alors nous aurons besoin d'une seule fonction objective $U(C)$. Les contraintes globales sont vérifiées, i.e. $Q^k(C) \geq ck, \forall ck \in CONS$ Avec $Q^k(C)$, est la valeur du K^{eme} critère de QoS de la composition c . Si nous supposons que le nombre de candidats par classe est L , alors le nombre global de compositions possibles est Ln .

2.2. Architecture générale du système

L'objectif de la conception globale est de présenter le processus proposé pour la sélection des services web composites à partir des plusieurs services web. La figure suivante illustre l'architecture globale proposée pour notre système.

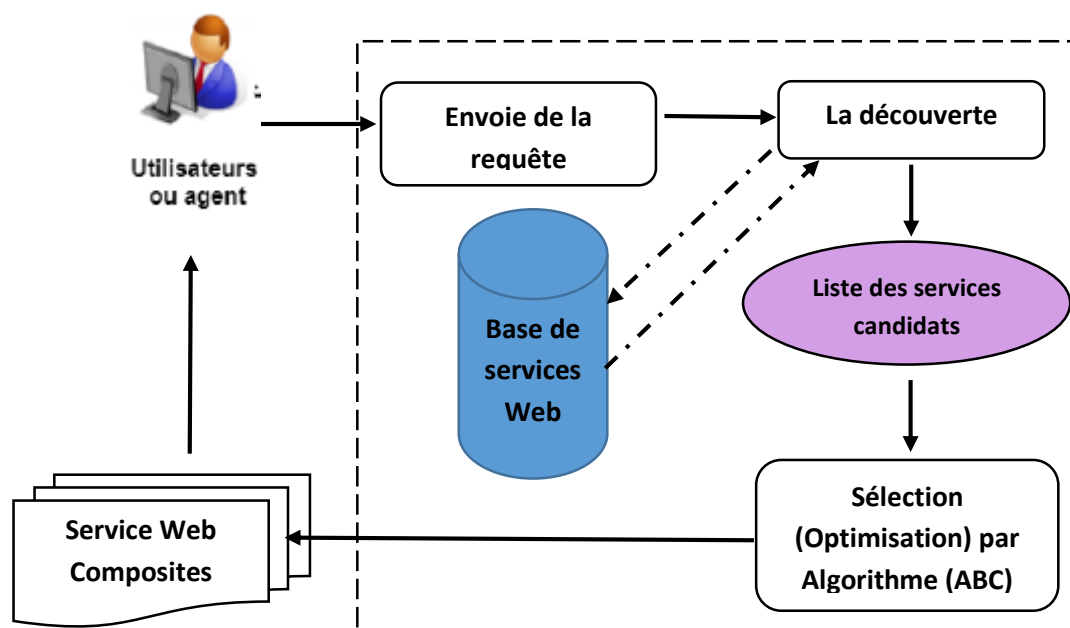


Figure 3.2 : Architecture générale du système.

Tout d'abord, le client (i.e., utilisateur ou agent) envoie une requête, puis une phase de découverte est faite pour analyser la demande et de découvrir les services aptes à réaliser cette requête en faisant une recherche dans la base des services qui ont une relation avec la demande du client et les retournés comme une liste des services candidats. Ensuite, la base de services se compose de différentes classes ou chaque classe contient des services qui offrent les mêmes fonctionnalités avec des valeurs QoS différentes.

La sélection consiste à choisir un service composite composé à partir des services candidats. Chaque service du service composite possède les QoS les plus convenable et satisfiable aux exigences de l'utilisateur. La sélection faite par l'algorithme d'optimisation d'abeilles artificielles (ABC).

Cette méthode va être étudiée en détail dans les prochaines sections de ce chapitre.

2.3. Architecture détaillée du système

Dans cette section, nous discutons des principaux composants de l'architecture détaillée de notre système.

2.3.1. Données d'entrées du système (la base de services)

Nous avons créé un schéma de service qui contient trois classes de services web, où chaque classe contient dix d'instances de service web. Chaque fournisseur se caractérise par quatre critères qualités de service : **latence**, **fiabilité**, **cout** et **débit**. Les valeurs de ces paramètres sont générées en se basant sur une distribution uniforme..

On considère la classe *reserverhotel* contenant plusieurs services qui fournit des taches de réservations d'hôtels par exemple :*HotelerumBooking Engine*, *Campass*, etc. mais avec différents QoS. Le Tableau 3.1 donne une description sur les services utilisés dans cet exemple.

Service	Nom	Fournisseur	q1(ms)	q2(hits/sec)	q3(%)	q4(%)
1	HotelerumBooking Engine	Hotelerum	110.38	45.3	65.5	26.45
2	ComfyHotelReservation	OrgBuisnessSoftware	509.2	28.3	41.3	78.3
3	ResAvenue	Avenue	481	41.11	49.8	98.4
4	HotelReservation System	CMSplaza	1024.1	18.6	32.6	32.1
5	Campass	OnPeak	363.4	60.7	19.9	27.5
6	Booking Engine	ReservHOTEL	563.45	54.9	21.78	11.5
7	HotelReservation System	StivaSoft	771.34	38.7	30.2	34.73
8	Weltraum PMS	Weltraum Software	192.5	60	72.2	100
9	RoomScope	RoomScope	300.67	43.33	46.69	42.1
10	Trawex Cloud Suite	TrawexTechnologie	204.92	71.4	53.2	60.3

Tableau 3.1 : Les services Web de la classe reserverhotel.

Chaque service Web possède un identificateur, un nom, le nom du fournisseur, et ses caractéristiques QoS. On a choisi quatre paramètres QoS, $q1$, $q2$, $q3$ et $q4$, qui représentent respectivement le Temps de Réponse, Débit, Fiabilité et le Cout.

2.3.2. Description de la requête

La requête de l'utilisateur consiste à vérifier la qualité de services de la composition sélectionnée par le system, en considérant les valeurs minimales ou maximales des quatre critères de QoS suivantes :

- La borne minimale de la fiabilité.
- La borne minimale du débit.
- La borne maximale du cout.
- La borne maximale de temps latence.

2.3.3. Critères de QoS considérés

Dans ce qui suit nous nous focalisons sur les cinq critères de QoS suivants pour calculer la qualité de la composition de services Web sélectionnée par notre système :

- **Temps de réponse** : est une performance qui représente la vitesse avec laquelle un service Web répond à une requête.
- **Débit** : c'est le Nombre total d'invocations / période de temps.
- **Fiabilité** : est la capacité d'un service à remplir ses fonctions requises dans les conditions indiquées pour une période de temps déterminée.

- **Cout d'un service:** est le coût à payer pour consommer le service, ce coût peut être fourni par le fournisseur du service.

2.3.4. Calcul de la qualité de composition de service web (SW)

2.3.4.1. Fonctions d'agrégation

En effet, certains critères de qualité de service (QoS) peuvent être évalués de manière qualitative, par exemple l'expérience utilisateur, la convivialité ou l'ergonomie d'une application. Cependant, la plupart des critères de QoS sont des attributs quantitatifs, tels que la disponibilité, la fiabilité, la sécurité, la bande passante, le temps de réponse, etc...

Il est vrai que les systèmes ne sont jamais totalement disponibles, fiables et sécurisés en raison de la présence inévitable d'erreurs et de défaillances. C'est pourquoi ces attributs sont généralement exprimés en probabilité. Par exemple, la disponibilité d'un système peut être exprimée comme le pourcentage de temps où le système est opérationnel par rapport au temps total.

Dans l'ensemble, les critères de QoS peuvent être divisés en deux catégories : les attributs négatifs et les attributs positifs. Les valeurs des attributs négatifs doivent être minimisées, tels que le temps de réponse et le coût, tandis que les valeurs des attributs positifs doivent être maximisées, tels que la disponibilité, la fiabilité et la sécurité. En d'autres termes, les attributs négatifs peuvent être considérés comme des coûts ou des pénalités, tandis que les attributs positifs peuvent être considérés comme des bénéfices ou des récompenses.

Enfin, pour simplifier l'analyse des critères de QoS, il est possible de considérer uniquement les attributs positifs et de multiplier les valeurs des attributs négatifs par (-1). De cette manière, tous les attributs peuvent être exprimés en termes de maximisation plutôt que de minimisation, ce qui facilite la comparaison et l'évaluation des différents systèmes ou services.

Maintenant, nous avons besoin de calculer les paramètres de QoS pour la composition des services Web, en se basant sur la signification de chaque critère de QoS, nous définissons un ensemble de fonctions d'agrégation permettant de calculer les valeurs agrégées des critères de QoS pour la composition des services Web.

Les fonctions d'agrégation de chaque critère sont données par le Tableau 3.2 :

Critère de QoS	Fonction d'agrégation
Suret� (fiabilit�)	$\prod_{i=1}^n sur(opi)$
D�bit	$\prod_{i=1}^n dis(opi)$
Latence	$\sum_{i=1}^n lat(opi)$
Cout	$\sum_{i=1}^n cou(opi)$

Tableau 3.2 : Les fonctions d'agr gation des crit res de QoS .

Alors la qualit  du web service composite peut  tre d crite par un vecteur de QoS

2.3.4.2. Fonction objective

La fonction objective est donn e comme suit :

$$F(c) = U(c)+p(c),$$

U est d fini comme suit :

$$U(c) = \sum_{k=1}^R wk * ((Qk(c) - Qminl(k))/(Qmaxl(k) - Qminl(k))) \tag{3.3}$$

La fonction est une agr gation de R crit res de QoS normalis s et agr g s, cette agr gation utilise des pond rations (ou priorit s) wk, par d faut  quitables (ie, si R=5 wk = 0.2). C repr sente une composition concr te, et Q'k(c) est la valeur de son K me crit re de QoS.

$$Qmin'(k) = \sum_{j=1}^n Qmin(j, k) \tag{3.4}$$

Ce terme repr sente la valeur minimale du crit re de QoS K associ    une composition

$$Qmax'(k) = \sum_{j=1}^n Qmax(j, k) \tag{3.5}$$

Ce terme repr sente la valeur maximale du crit re de QoS K associ    une composition. $Qmin(j, k) = \min \forall sji \in Sj Qk(sji)$ (3.6)

Ce terme repr sente la valeur minimale du crit re de QOS k de la classe j (i.e un service simple).

$$Qmax(j, k) = \max \forall sji \in S j Qk(sji) \tag{3.7}$$

Ce terme repr sente la valeur maximale du crit re de QoS k de la classe j (i.e un service simple).

$P(c)$, représente la pénalité associée à une composition, elle nulle si aucune contrainte n'est violée et elle négative si au moins une contrainte n'est pas respectée :

$$p(c) = \sum_{j=1}^n (Dk)^2(c) \quad (3.8)$$

$$\text{et } Dk(c) = 0 \quad (3.9)$$

$$\text{si } Q^k(c) \geq \text{cons}(k) \quad \text{et } |Q^k(c) - \text{cons}(k)| \quad (3.10)$$

$\text{Cons}(k)$: représente la K^{eme} contrainte globale de l'utilisateur.

Nous utilisons le score de la fonction objective «fitness» pour évaluer la qualité de la composition des services Web. La meilleure Web service composite sera celle avec le score maximal de fitness.

2.3.5. Sélection des services composites

C'est une étape très importante dans notre travail qui sert à trouver un service Web composite qui répond au besoin du client (i.e., l'utilisateur). Cela consiste à choisir ou sélectionner parmi les services Web de chaque classe ce qui répond aux exigences du client pour former le service composite ; c.-à-d. de déterminer l'ensemble des top-services de toutes les classes qui forme le service composite le plus adéquat à la requête fournit selon les QoS.

3. Optimisation par Colonie d'abeilles Artificielles

L'optimisation par colonie d'abeilles est une famille très récente des métaheuristiques. Son principe est basé sur le comportement des abeilles réelles dans la vie. Cette approche de résolution fait l'objet de notre étude [24].

3.1. Algorithme de colonies des abeilles artificielles (ABC)

L'algorithme de colonie d'abeilles artificielles (ABC) a été introduit par Dervis Karaboga et développe depuis 2005 [25]. Cet algorithme simule le comportement d'alimentation d'un essaim d'abeilles. L'algorithme de colonie d'abeilles artificielles est l'un des algorithmes par essaim le plus largement étudié et appliqué pour résoudre des problèmes du monde réel.

3.2. Caractéristiques de l'algorithme ABC

L'algorithme ABC a montré une efficacité remarquable dans la recherche des solutions optimales. Il surmonte le problème de l'optimum local. Il est facile à implémenter. Il utilise plusieurs paramètres réglables. En outre, l'algorithme a une robustesse remarquable, produisant des taux de réussite de 100% dans beaucoup de cas qu'il a traité. Sa technique c'est l'une des meilleures, concernant la rapidité de trouver l'optimum et la précision des résultats obtenus.

3.3. Principe général de l'algorithme ABC

Dans cet algorithme, les abeilles artificielles sont classifiées en trois groupes : les abeilles employées qui recherchent des sources de nourriture, les spectatrices (les abeilles observatrices) qui évaluent les sources trouvées et les scouts (éclaireuses) sont chargées de trouver de nouvelles sources nourritures. Les abeilles employées et spectatrices suivent le procédé d'exploitation dans l'espace de recherche d'une part et les scouts commandent le procédé d'exploration d'autre part. Une position de source de nourriture représente une solution possible au problème à optimiser dans notre cas c'est le service web candidat [27].

L'algorithme ABC commence par n solutions (sources de nourriture, i.e., les services web candidats) de dimension d qui sont modifiées par les abeilles artificielles. Chaque solution $\vec{x} = [x_{i1}, x_{i2}, \dots, x_{id}]$ est évaluée par une fonction de fitness $f(\vec{x})$, $i = 1, \dots, n$.

Les abeilles visent à découvrir des endroits contenant des sources de nourriture (régions de l'espace de recherche) avec une grande quantité de nectar (une bonne fitness). Si la quantité de nectar d'une nouvelle source est supérieure à celle de la précédente dans la mémoire, ils mettent à jour la nouvelle position et oublient la précédente (greedy selection). Si une solution n'est pas améliorée par un nombre prédéterminé d'essais, contrôlé par le paramètre de la limite, alors la source de nourriture est abandonnée par l'abeille employée correspondante et celle-ci devient une abeille scout [37].

Le critère d'arrêt qui est généralement le nombre d'itérations.

```

Placer chaque abeille employée sur une position aléatoire dans l'espace de recherche.
Tant que le critère d'arrêt n'est pas atteint faire
    Pour toutes les abeilles employées faire
        Si le nbr d'étapes sur la même position = la limite alors
            Choisir une position aléatoire dans l'espace de recherche ;
        Sinon
            tenter d'améliorer la position ;
            Si la meilleure position est trouvée, alors
                changer la position ;
                Réinitialisation le nbr d'étapes sur la même position ;
            Fin Si ;
        Fin Si ;
    Fin Pour ;
    Pour toutes les abeilles spectateurs faire
        Choisir la position d'abeille employée ;
        tenter d'améliorer la position ;
    Fin pour ;
Fin tant que

```

Figure 3.3 : Pseudo-code de l'algorithme ABC [37].

4. Modélisation UML

Le langage UML (Unified Modeling Language, ou langage de modélisation unifié) a été pensé pour être un langage de modélisation visuelle commun, et riche sémantiquement et syntaxiquement. Il est destiné à l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que leur comportement. L'UML a des applications qui vont au-delà du développement logiciel, notamment pour les flux de processus dans l'industrie.

L'UML n'est pas un langage de programmation, mais il existe des outils qui peuvent être utilisés pour générer du code en plusieurs langages à partir de diagrammes UML. L'UML a une relation directe avec l'analyse et la conception orientées objet [21].

4.1. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation permet de structurer les besoins des utilisateurs et les objectifs correspondants de notre système d'abeilles. Il permet aussi d'identifier les possibilités d'interactions entre le système et ces utilisateurs (intervenants extérieurs au système). Il part du principe que les objectifs du système sont tous motivés. La Figure 3.3 représente le diagramme de cas d'utilisation de notre système [22]

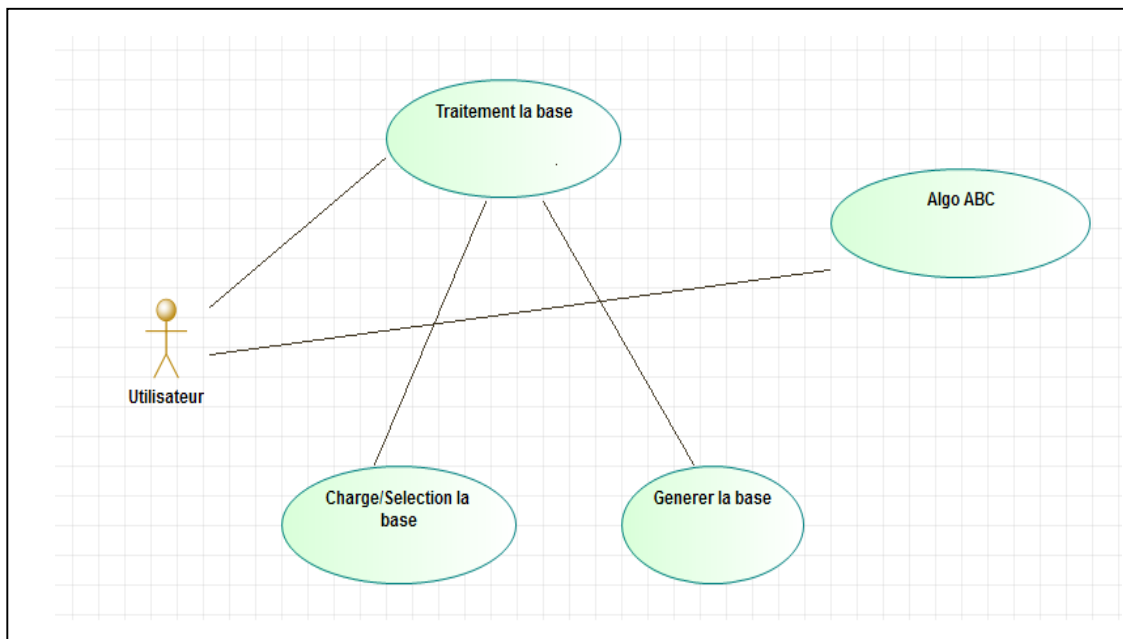


Figure 3.4 : Diagramme de cas d'utilisation

4.2. Diagramme de classes

Le diagramme de classes permet de représenter les classes intervenant dans le système. Il constitue un élément très important de la modélisation et permettre de définir les classes qui sont les composantes de notre système final, donc nous avons l'interface qui fait appelle à la classe Lecture qui concerne tous ce qui est traitements de la base des SW et la classe de l'algorithme d'abeilles (ABC) qui permette la sélection des meilleures abeilles.

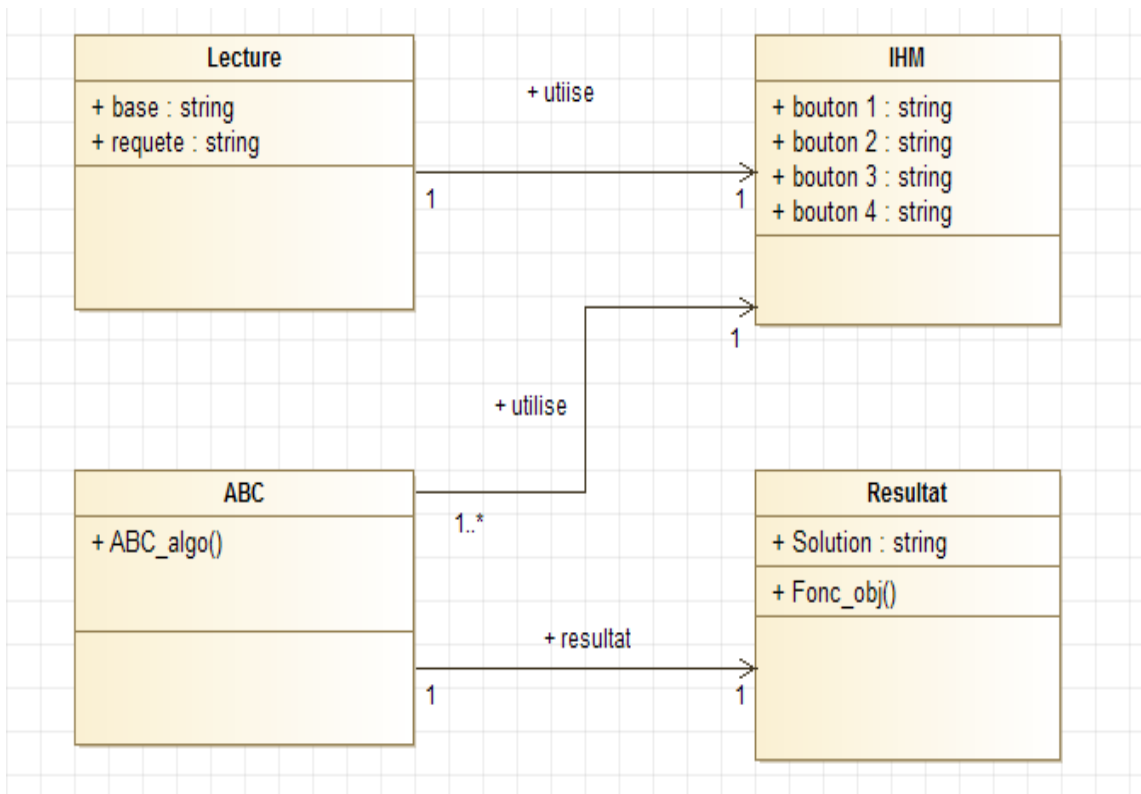


Figure 3.5 : Diagramme de classes

5. Conclusion

Dans ce chapitre nous avons présenté une approche pour la sélection et l'optimisation de services Web on se basant sur des critères de qualités de service (QoS), utilisant l'algorithme des abeilles artificielles. Nous avons expliqué comment cet algorithme fonctionne et dans le prochain chapitre, nous allons concentrer sur la mise en œuvre de cet algorithme et la présentation des résultats obtenus.

Chapitre 4 : Implémentation du système

1. Introduction

Après avoir la conception générale de notre projet dans le chapitre précède, nous devons passer à l'étape suivante du projet qui est l'implémentation.

Dans ce chapitre final, nous allons présenter les différents outils de développement ainsi que les langages appris et exploités dans la réalisation de notre projet.

Nous exécutons un exemple illustratif du choix du meilleur service trouver un service Web Composite (i.e., top-services) qui répond au besoin de l'utilisateur et le plus adéquat à la requête fournit selon les QoS utilisant l'algorithme de colonie d'abeilles artificielles (ABC).

2. Outils et environnement de développement

Avant de commencer l'implémentation de notre application, nous allons tout d'abord spécifier les langages de programmation et les outils utilisés qui nous ont semblé être un bon choix vu les avantages qu'ils offrent. Pour réaliser notre système, nous avons un PC Asus de Windows 7 (64bits) qui est décrit avec la figure suivante :

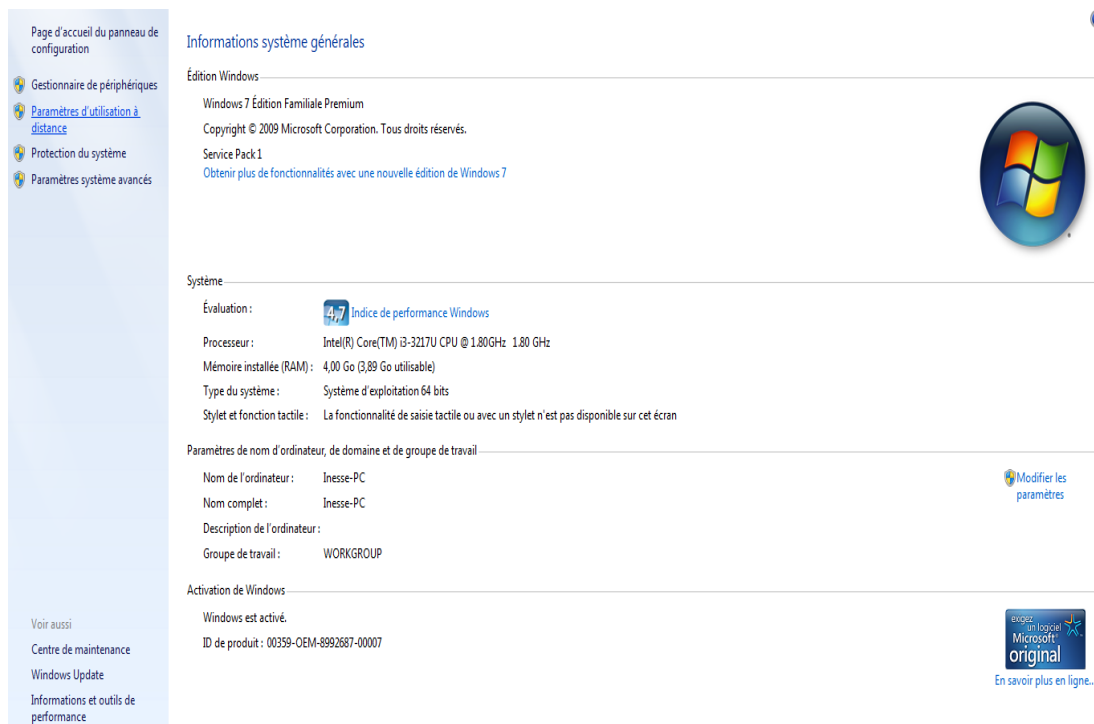


Figure 4.1 : Environnement logiciel utilisé

2.1. Le Langage JAVA

Le langage Java est un langage de programmation informatique orienté objet créé par James Gosling et son équipe chez Sun Microsystems [29].

Le Java introduit une couche intermédiaire supplémentaire entre le code et le langage machine, ce qu'on appelle la Java Virtual Machine (JVM) [30], qui doit être installée sur tout ordinateur voulant exécuter un programme écrit en Java, et qui est spécifique à cette configuration matérielle. Il s'agit donc d'un langage compilé dont le code machine sera interprété, traduit au travers de la JVM dans le langage de la machine hôte. La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation [31].



Figure 4.2 : Langage JAVA

2.2. L'IDE NetBeans

Est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License) et GPLv2. En plus de Java, NetBeans permet la prise en charge native de divers langages tels le C, le C++, le JavaScript, le XML, le Groovy, le PHP et le HTML, ou d'autres (dont Python et Ruby) par l'ajout de *greffons*. Il offre toutes les facilités d'un IDE moderne (éditeur avec coloration syntaxique, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web). Compilé en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Développement Kit JDK est requis pour les développements en Java.

NetBeans constitue par ailleurs une plateforme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plateforme [32].

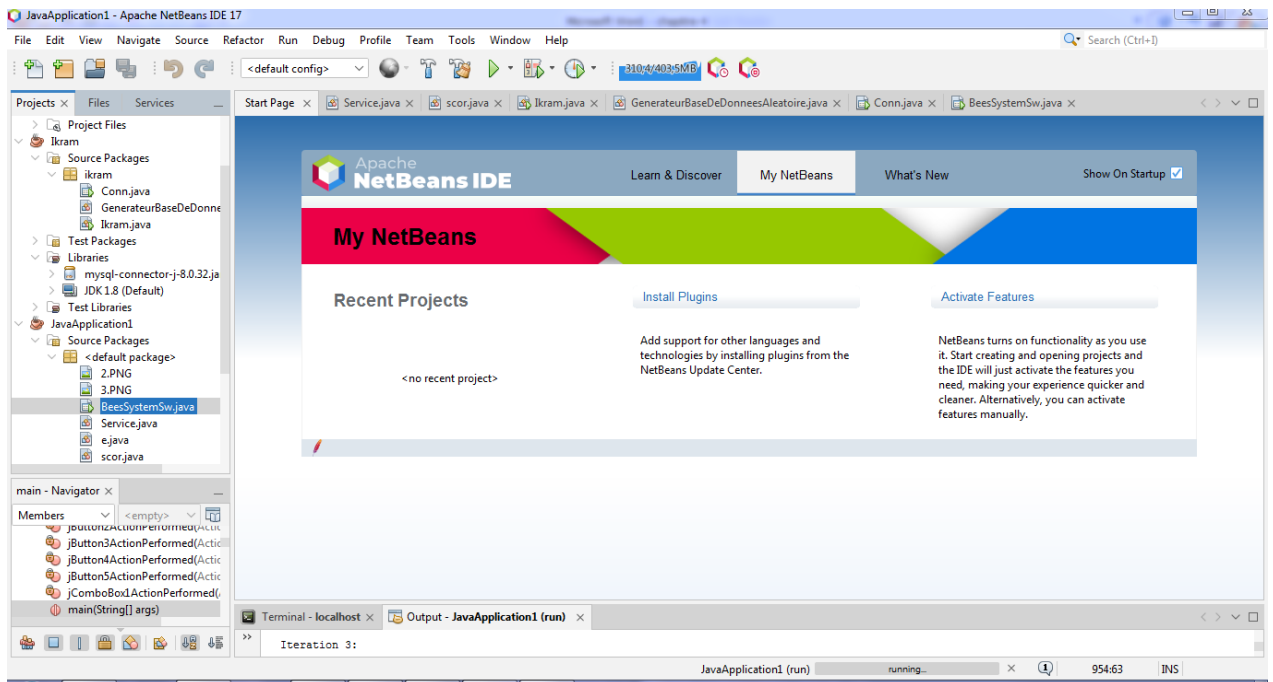


Figure 4.3 : Interface principale de NetBeans.

2.3. Gestion de base de données (phpMyAdmin)

- **XAMPP**

Un ensemble de logiciels permettant de mettre en place facilement un service web et un serveur FTP.

Il s'agit d'une distribution de logiciels libres (Apache MySQL Perl PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide.

Ainsi, il est à la portée d'un grand nombre de personnes, puisqu'il ne requiert pas de connaissances particulières et fonctionne de plus sur les systèmes d'exploitation les plus répandus [33].

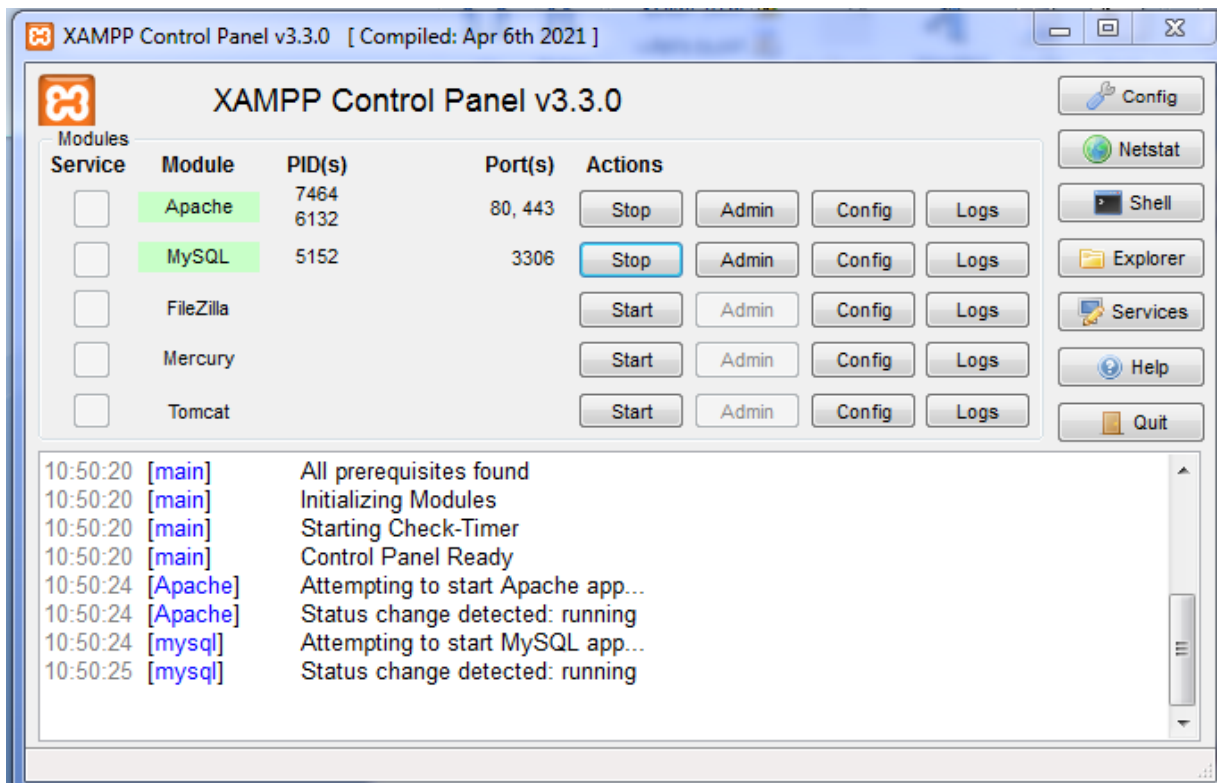


Figure 4.4 : XAMPP.

- **phpMyAdmin**

Est une interface conviviale qui permet de gérer très facilement une base de données, sans nécessiter une connaissance avancée de requêtes SQL.

Avec le gestionnaire de base de données PHPMyAdmin, on pourra rapidement créer et supprimer des base de donnée, assurer mise à jour (ajouter, modifier et supprimer...) des tables et permet d'exécuter des requête SQL [34].

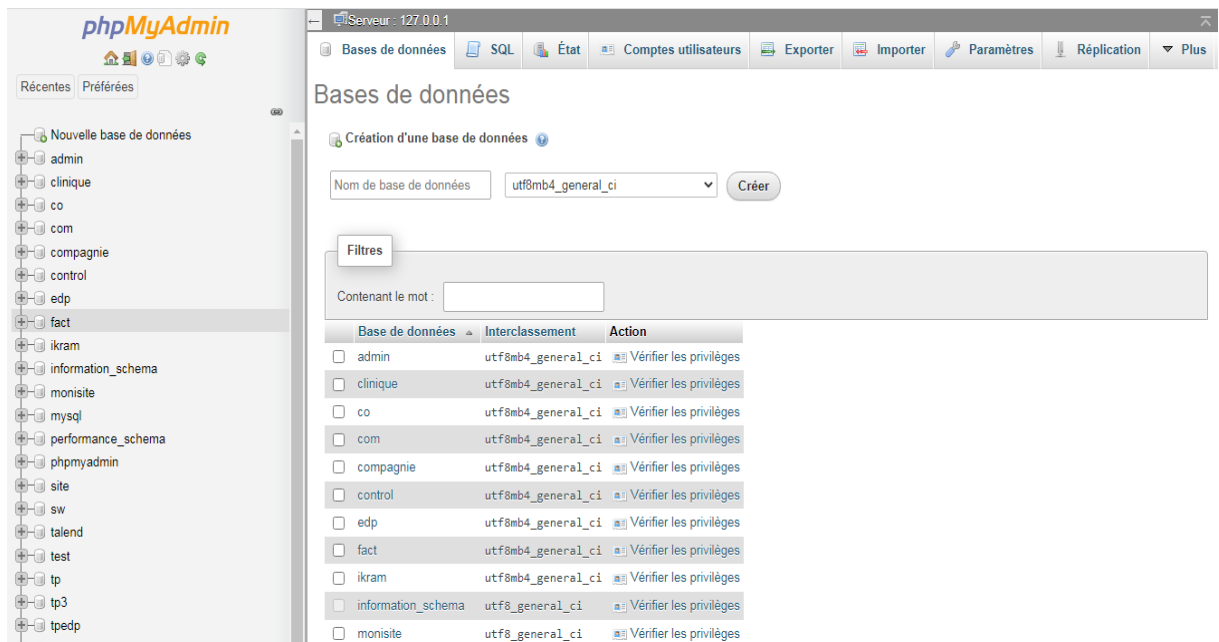


Figure 4.5 : Interface Serveur phpMyAdmin.

- **MySQL**

MySQL est un serveur de bases de données relationnelles Open Source. Un serveur de bases de données stocke les données dans des tables séparées plutôt que de tout rassembler dans une seule table. Cela améliore la rapidité et la souplesse de l'ensemble. Les tables sont reliées par des relations définies, qui rendent possible la combinaison de données entre plusieurs tables durant une requête. Le SQL dans "MySQL" signifie "Structured Query Language" : le langage standard pour les traitements de bases de données [35].

3. Présentation des interfaces de notre système

3.1. Interface d'accueil

La fenêtre principale de l'interface donne un aperçu global du système, offrant une présentation claire de l'objectif du système afin de faciliter la compréhension des opérations et des traitements ultérieurs.



Figure 4.6 : Interface d'accueil.

3.2. Interface du chargement de la base des services

Tout d'abord nous avons commencé par choisir les classes comme : réservation, reserverhotel et reservervoiture.

Si on sélectionne une classe comme par exemple reserverhotel on obtient les rubriques où les différents services qui répondent à la requête donnée par l'utilisateur contenant leurs différentes qualités de QoS.

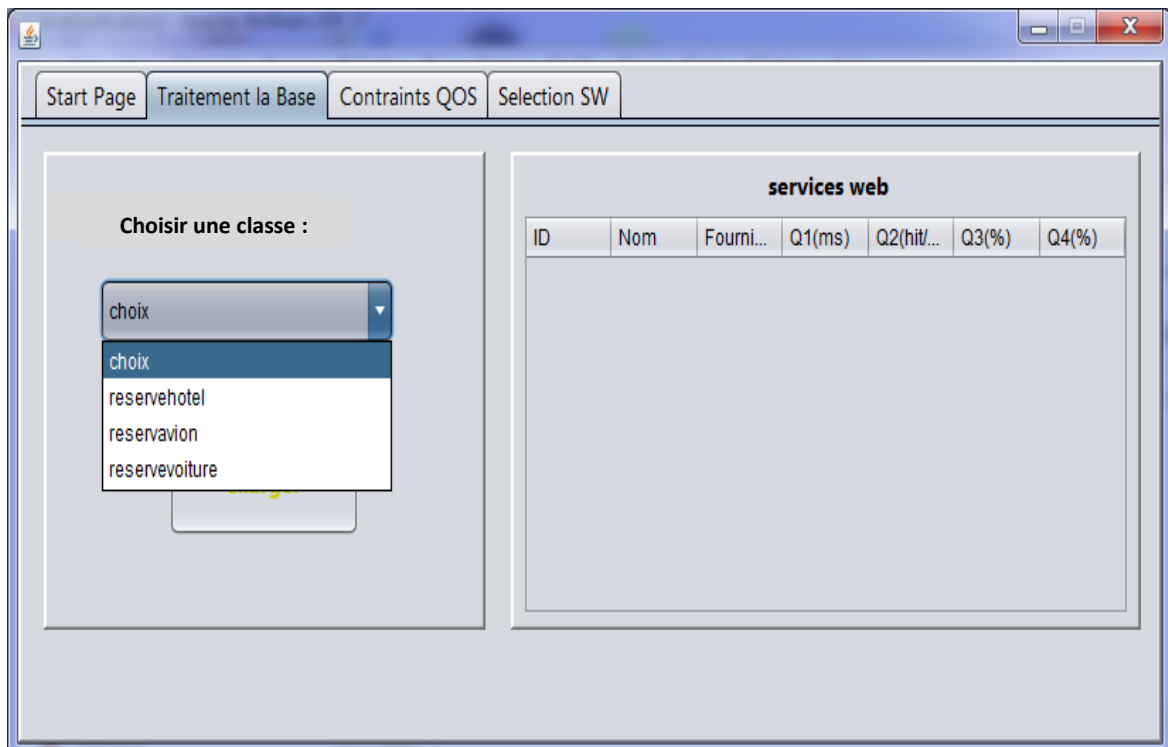


Figure 4.7: la Requête demandée par l'utilisateur.

Ensuite, après avoir sélectionné la classe, nous appuyons sur le bouton charger les données de la base des services Web, alors chaque service est caractérisé par quatre critères de QoS.

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

- **Temps de réponse (Q1)** : est une performance qui représente la vitesse avec laquelle un service Web répond à une requête.
- **Débit (Q2)** : c'est le Nombre total d'invocations / période de temps.
- **Fiabilité (Q3)** : est la capacité d'un service à remplir ses fonctions requises dans les conditions indiquées pour une période de temps déterminée.
- **Cout d'un service (Q4)** : est le coût à payer pour consommer le service, ce coût peut être fourni par le fournisseur du service.

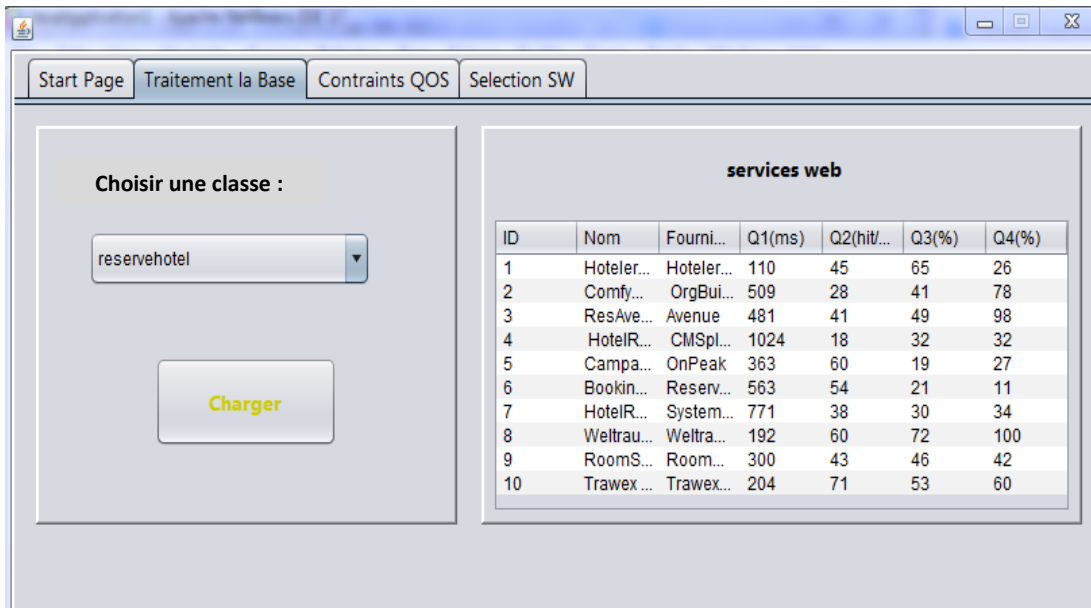


Figure 4.8 : Liste des services candidats de la classe reservehotel.

Si aucune base ne correspond à la requête donnée ce message est apparaître.

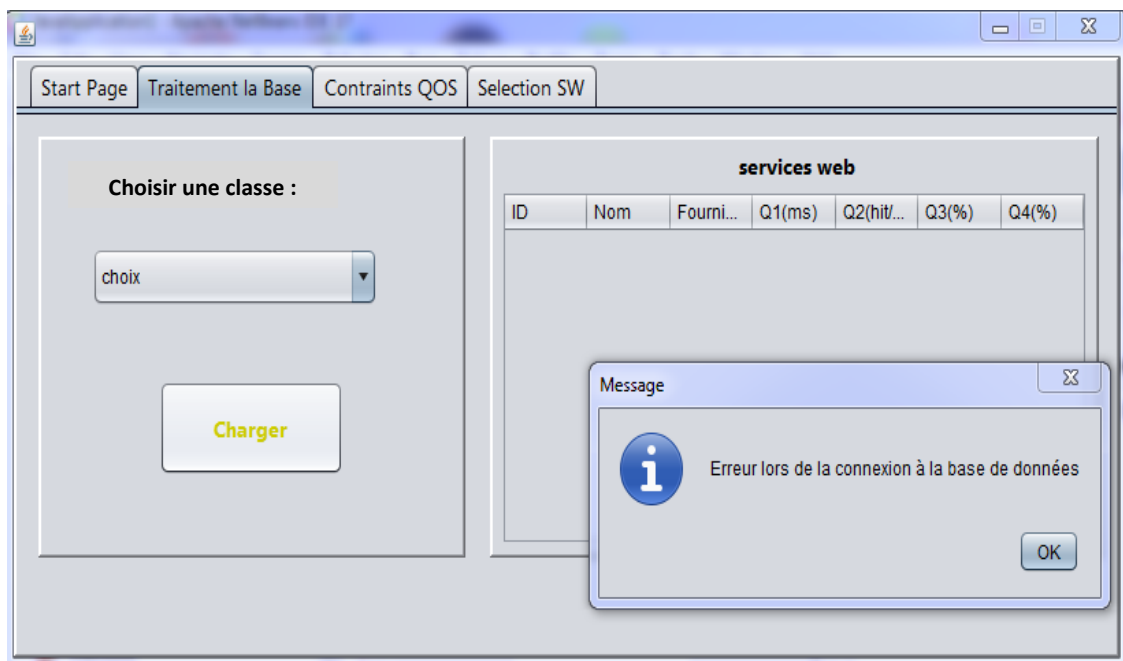


Figure 4.9 : Message d'erreur de connexion à la base de données.

3.3. Interface de contraintes QoS

Dans cette interface, nous spécifions des exigences de qualité de service (QoS) qui doivent être respectées lors de la sélection des compositions de services Web. Nous veillons à ne pas dépasser un certain coût total pour l'ensemble des services de la composition, ainsi qu'une

limite de latence. De plus, nous garantissons un niveau minimum de sécurité, de réputation et de disponibilité pour les services sélectionnés.

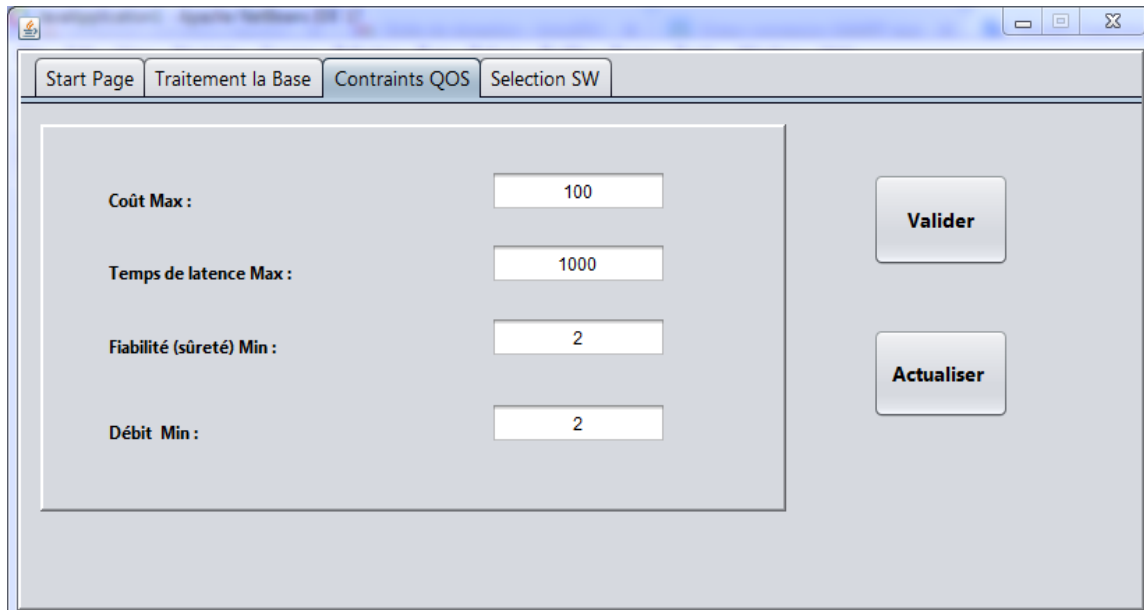


Figure 4.10 : Contraintes globales de qualités de services

3.4. Interface de sélection des services Web (SW)

Maintenant nous pouvons lancer la sélection de services Web.

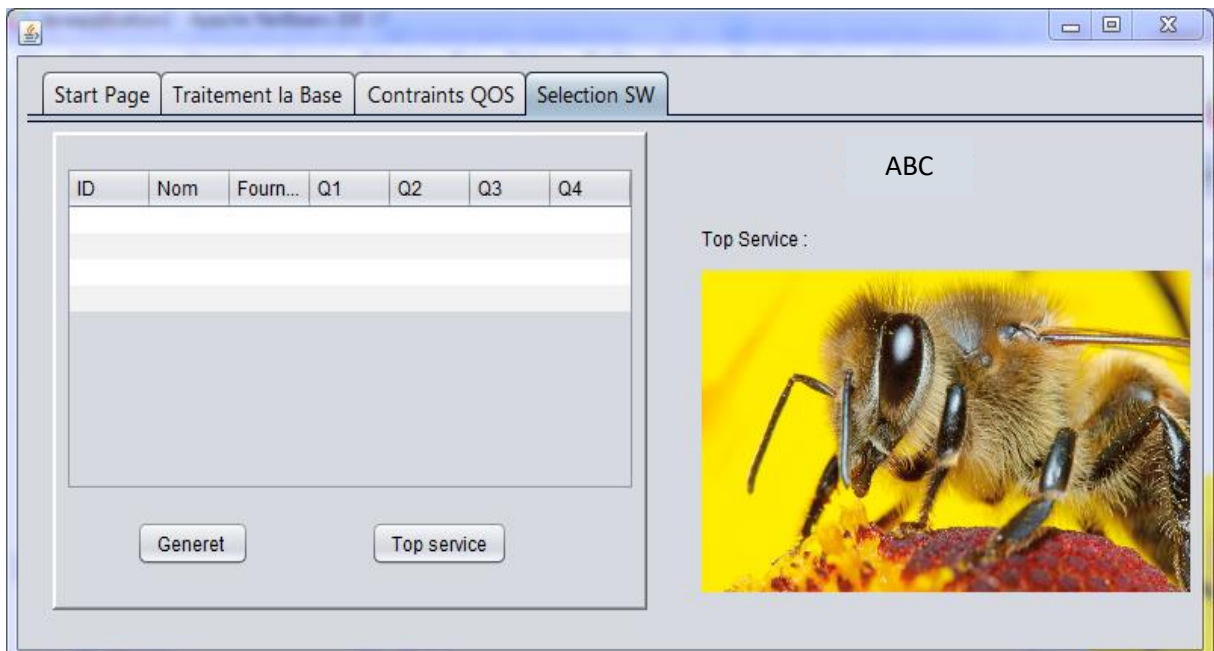


Figure 4.11 : La phase de la Sélection.

Le bouton Générer affiche les lignes de la base de données, et le bouton Top service sert à sélectionner le meilleur service pour chaque classe choisi.

Maintenant nous pouvons lancer la sélection de Top service web, en précisant le nombre maximum des itérations = 10.

L’algorithme d’optimisation « ABC » permet de faire une optimisation mono objective qui améliore les résultats obtenus en chaque itération, jusqu’à l’obtention d’une solution proche de l’optimum c.-à-d., un service optimal (abeille) qui satisfait au mieux les besoins de l’utilisateur.

La figure suivante représente les différentes étapes ainsi les résultats de la sélection d’un service dans une seule classe : par exemple si on choisit la classe 1 « reserverhotel »

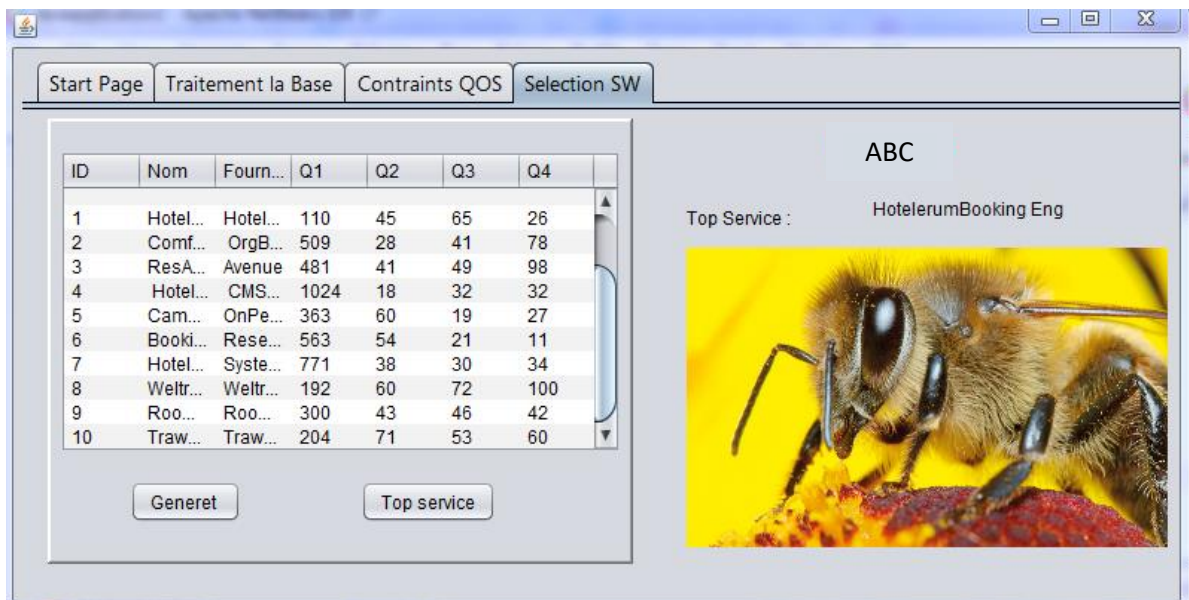
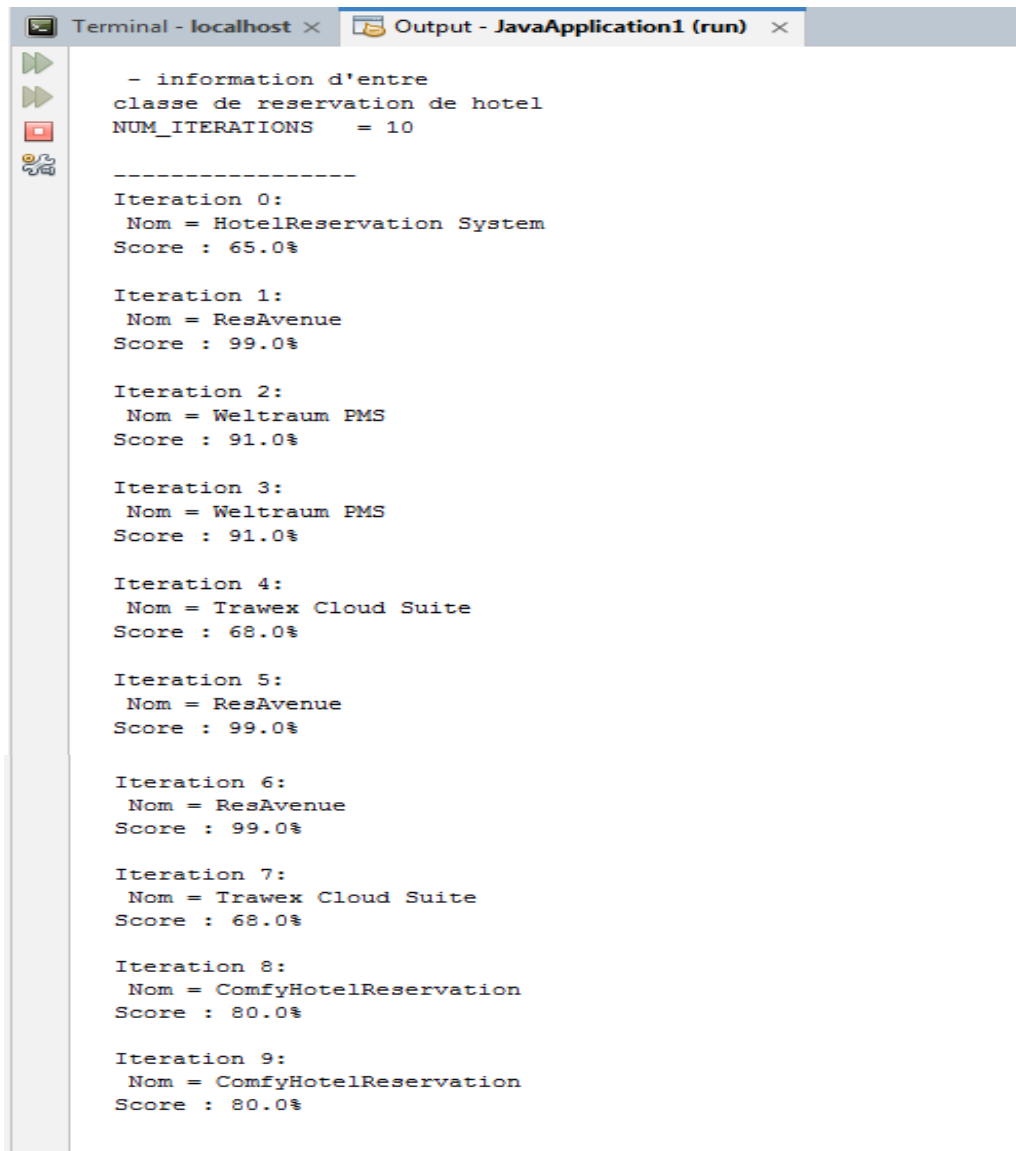


Figure 4.12 : Sélection le meilleur service.



```
Terminal - localhost x Output - JavaApplication1 (run) x
- information d'entre
classe de reservation de hotel
NUM_ITERATIONS = 10
-----
Iteration 0:
  Nom = HotelReservation System
  Score : 65.0%

Iteration 1:
  Nom = ResAvenue
  Score : 99.0%

Iteration 2:
  Nom = Weltraum PMS
  Score : 91.0%

Iteration 3:
  Nom = Weltraum PMS
  Score : 91.0%

Iteration 4:
  Nom = Trawex Cloud Suite
  Score : 68.0%

Iteration 5:
  Nom = ResAvenue
  Score : 99.0%

Iteration 6:
  Nom = ResAvenue
  Score : 99.0%

Iteration 7:
  Nom = Trawex Cloud Suite
  Score : 68.0%

Iteration 8:
  Nom = ComfyHotelReservation
  Score : 80.0%

Iteration 9:
  Nom = ComfyHotelReservation
  Score : 80.0%
```

Figure 4.13 : Résultat de l’algorithme ABC pour la classe reserverhotel.

La figure suivante représente les différentes étapes ainsi les résultats de la sélection d’un service dans une seule classe : par exemple si on choisit la classe 2 «reservavion»

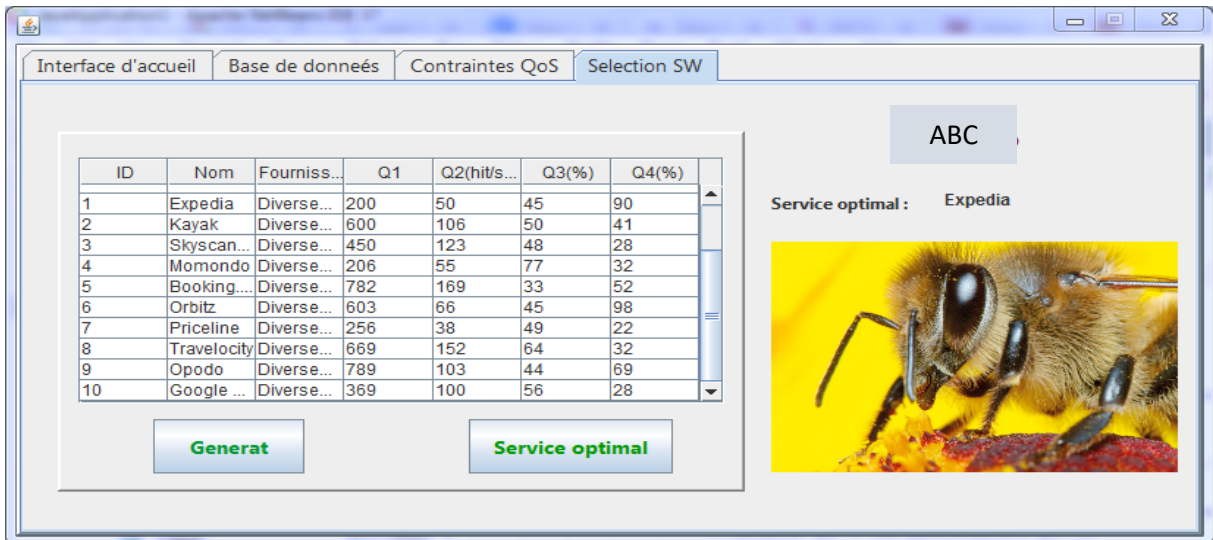


Figure 4.14 : Sélection le meilleur service de la classe reservavion

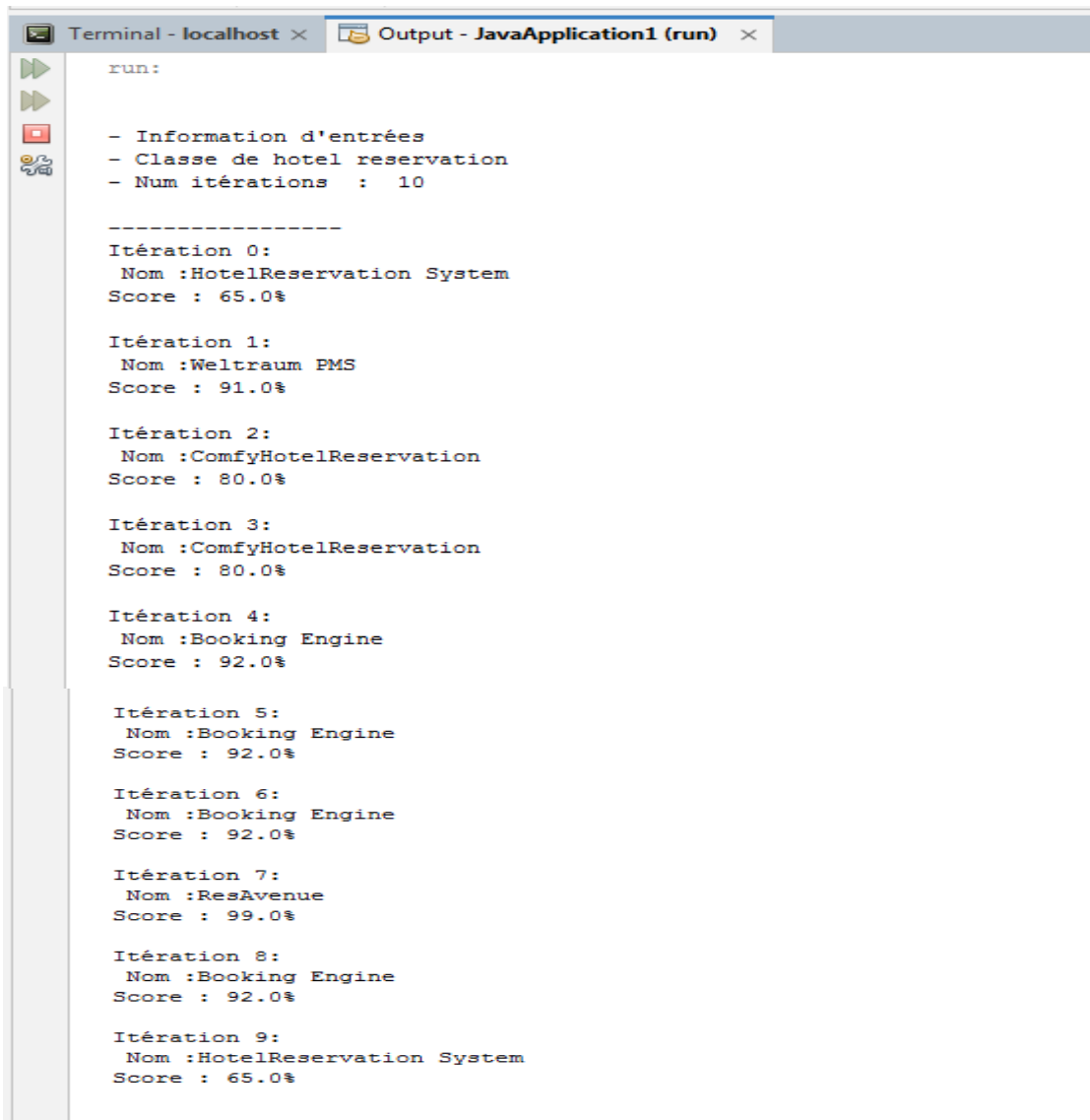


Figure 4.15 : Résultat de l'algorithme ABC pour reservavion.

La figure suivante représente les différentes étapes ainsi les résultats de la sélection d'un service dans une seule classe : par exemple si on choisit la classe 3 «reservoiriture».

ID	Nom	Fourniss...	Q1(ms)	Q2(hit/s...	Q3(%)	Q4(%)
1	CarGo	Hertz	110	48	72	300
2	DriveNow	BMW	50	38	400	48
3	Zipcar	Avis	110	90	30	99
4	Turo	Particuli...	90	200	60	74
5	Getarou...	Particuli...	60	40	300	102
6	CarShare	Enterpri...	50	400	514	777
7	Sixt Share	Sixt	400	28	23	60
8	BlaBlaCar	Particuli...	500	82	60	40
9	Aurore C...	Europcar	230	50	38	22
10	Car2Go	Mercede...	770	90	40	29

Figure 4.16 : Sélection le meilleur service de classe reservevoiture

Finalement le Top-services composite inclut : HotelrumbookingEng, Expedia et Zipcar .

4. Conclusion

Ce chapitre a abordé la présentation du langage de programmation et de l'environnement de développement, ainsi que la description de notre application. Nous avons discuté de l'interface et de la mise en œuvre de la méthode choisie pour résoudre le problème. Nous avons proposé un algorithme d'optimisation basé sur les colonies d'abeilles, appelé "ABC", pour résoudre le problème de sélection de services Web. Cet exemple d'application nous a permis d'illustrer notre travail et de présenter les résultats de la validation de notre approche, afin d'évaluer son efficacité.

Conclusion Générale et Perspectives

La sélection des services web est une étape primordiale dans la prise en charge des compositions dynamiques. Ils constituent la technologie de base pour le développement d'architectures orientées services. Ces architectures sont de plus en plus répandues sur le Web. La composition de services Web est utilisée pour composer les services Web existants afin d'obtenir une nouvelle fonction à valeur ajoutée, qui est devenue l'une des fonctionnalités les plus importantes des services Web. Avec le nombre croissant de services web offrant des fonctionnalités similaires, la QoS devient un critère important de sélection des meilleurs services disponibles.

Dans ce mémoire, nous avons présenté le problème de sélection de services web comme un problème d'optimisation combinatoire où les méthodes méta heuristiques sont nécessaires pour relever le défi de Sélection de service Web basé QoS. Nous avons présenté un travail qui a pour but de faire une optimisation mono objective basée sur l'algorithme de colonie d'abeilles artificielles (ABC) pour la sélection des Tops services Web.

D'après notre propre cas d'application et les résultats obtenus, les algorithmes à base colonies d'abeilles artificielles « ABC » fournissent de bons résultats pour les problèmes d'optimisations difficiles, on peut dire que cette méthode est mieux adaptée aux problèmes d'optimisation combinatoire. Tout travail est amené à être amélioré, en ce sens, notre application peut encore évoluer et se voir améliorer, Il est possible de faire une optimisation multi objective ou hybride à base de colonies d'abeilles pour résoudre le même problème proposé dont le but d'améliorer et de varier les résultats finaux.

Bibliographie

- [1] : <http://www-106.ibm.com/developerworks/webservices/>
- [2] : www.w3c.org
- [3] : BARKAT Abdelbasset, Composition de service web dans le cloud computing, Thèse doctorat. Université biskra, 2019.
- [4] : KHELLAF Radhia, verification de la compatibilite des services web pour une composition, Thèse Magister, Université Constantine 2 .
- [5] : Cour de Professeur Mohamed Yousfi, Université, Morocco.
- [6] : <http://sdz.tdct.org/sdz/les-services-web.html>
- [7] : DERRADJI HAZAR, Mémoire Master « Une Approche Effective Pour La Sélection Des Services Web Composites » , Université Mohamed Khider – BISKRA 2019.
- [8] : <https://www.rapport-gratuit.com/les-services-web/>
- [9] : DEHANE Aicha Djihad, et Melle KEBIR Zohra, « Evaluation des techniques de codage d'ontologies sur les performances de la composition de services web », Thèse Master, Université Abou Bekr Belkaid Tlemcen, 2012.
- [10] : HALFAOUI Amal, La sélection des services web dans une composition à base de critères non fonctionnels. Doctorat en science par.
- [11] : Abdelkrim Amirat , Sélection par la composition des services web basé sur la logique floue » présenté par :.
- [12] : Sélection et composition de services Web pour la génération d'applications adaptées au contexte d'utilisation, Article · Janvier DOCTEUR DE L'UNIVERSITE JOSEPH FOURIER Céline LOPEZ-VELASCO
- [13] : DEHMECHE Abdelbaqi, Diplôme de master Service web Embarqué (Technologie SOAP sous Android) pour la gestion d'une Bibliothèque MI, UNIVERSITE MOHAMED BOUDIAF - M'SILA.
- [14] : AOUCHE Ahmed, Sélection des services Web dans les systèmes distribués : étude comparative, UNIVERSITE LARBI BEN MHIDI D'OUM EL BOUAGHI.

- [15] : Tahar Moulay SAIDA, BENKHALED Hamid Naceur, REGRAGUI Sid Ahmed Tidjani, sélection de web services à base de la qualité de service, 2016 .
- [16] : <https://www.rapport-gratuit.com/les-services-web>.
- [17] : [Bekaddour Hassen](#) , La sélection multi objectifs des services web à base de recuit simulé .livre auteur , [Éditions universitaires européennes](#)
- [18] : TURKI KOSENTINI Hazar, Leila BACCOUCHE, Henda HAJJAMI BEN GHEZALA, Etude de Problème de Sélection de Services Web en tenant Compte des contraintes temps réel .Laboratoire RIADI-GDL, Ecole Nationale des Sciences de l'Informatique, Campus Universitaire de la Manouba, 2010 Manouba, Tunisie ;
- [19] : Composition et interopération des services web sémantiques A L'UNIVERSITE DE TLEMCEM .Par Mr Hadjila FethAllah
- [20] : La sélection de services Web À base de l'algorithme d'abeilles présenté par Melle BOUROUIS Meriem Et Melle DJERRIRI Sabah pour l'obtention du diplôme de Master en Informatique 2013
- [21] : Diplôme master : Une approche parallèle pour la sélection des services web basée sur la logique floue . Présenté par : Bekakra Aicha
- [22] : <https://www.lucidchart.com/pages/fr/langage-uml>
- [23] : <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/diagramme-de-cas-de-utilisation/>
- [24] : <https://www.lucidchart.com/pages/fr/diagramme-de-classes-uml>
- [25]: Souhil MOUASSA, Optimisation de l'écoulement de puissance métaheuristique (technique des abeilles) en présence d'une source renouvelable (éolienne) et des dispositifs FACTS, Faculté de Technologie Présenté au département d'Electrotechnique Pour obtenir le diplôme De Magister En Electrotechnique, UNIVERSITE Ferhat ABBAS SETIF (ALGERIE).
- [26]: Françoise Rebecca a Montréal déploiement de passerelles pour l'intégration du réseau capillaire dans le réseau cellulaire. Université du Québec
- [27]: Hamid BOUALI, Bachir BENHALA, Hamid BOUYGHF, Performance study of MultiObjective Artificial Bee Colony (MOABC) Algorithm by Numerical Problems Benchmark, Conference Paper · April 2020, Université Moulay Ismail LEAB, Department of physics FS, Meknes, Moulay Ismail, Université, Morocco.

- [28]: Boudouaoui yassine « contribution à la synthèse automatiques des systèmes flous par approche meta_heuristique ». Université. m'hamed bougara boumerdès;
- [29]: James Gosling, Bill Joy, Guy Steele, Gilad Bracha, Alex Buckley. "The Java Language Specification, Java SE 8 Edition." Oracle, 2015.
- [30]: David Flanagan. "Java in a Nutshell: A Desktop Quick Reference." O'Reilly Media, 2018
- [31]: Patrick Naughton, Herbert Schildt. "The Complete Reference Java 2, Fourth Edition." McGraw-Hill Education, 2002.
- [32]: <https://www.techno-science.net/glossaire-definition/NetBeans.html>
- [33]: <https://desgeeksetdeslettres.com/xampp-plateforme-pour-heberger-son-propre-site-web/>
- [34]: Eric Daspét, Cyril Pierre de Geyer, PHP 5 Avancé 4e édition Eyrolles. (livre) .phpmyadmin ;
- [35] : Frédéric Moitry, Jean-Marie CochetEAU, Emmanuel Friedmann, Adobe Dreamweaver, DUNOD, Paris, 2000 ;
- [36] : BERRABAH Samir, Mémoire Master « Sélection de service web basée sur la FOA (fruit Fly optimization algorithm) », Université Abou Bakr Belkaid– Tlemcen, 2017.
- [37] : Hassina SERIDI, These de doctorat « Approches Bio-inspirées pour la Sélection d'Attribut », 2015.

