République Algérienne Démocratique et Populaire
Université Abou Bekr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

pour l'obtention du diplôme de Master en Informatique

Option :**S**ystèmes d'**I**nformation et de **C**onnaissance (**SIC)**

Thème

La sélection des services Web À base d'algorithme de la recherche Harmonique

Réalisé par :

Mr ALILI Sofiane

Mr ZIANI CHERIF Bassim

Soutenu le 22 Juin 2015 devant le jury composé de :

- Mr.HADJILA Fathallah (President)
- Mr. MERZOUG Mohammed (Encadreur)
- Mr. SMAHI Ismail (Examinateur)
- Mr. Etchiali Abdelhak (Examinateur)

Année universitaire : 2014 - 2015

Table des matières

Table des figures

Introduction Générale

1. **Contexte**

Le Web devient à bien des égards un outil puissant pour soutenir la stratégie d'affaires des entreprises actuelle, ses dernières sont en train de devenir de plus en plus dépendante des nouvelles technologies du Web dont l'information est produite sur commandepour profiter d es n ouvelles o pportunités c ommerciales e t r éagir r apidement à l 'évolution d es exigences du marché. Pour conquérir ces atouts, les organisations doivent assurer l'intégration et l'interopérabilité de leurs applications au niveau interne (EAI) ou Enterprise application Intégration et au niveau partenaire (B2B)i.e. Business To Business, est cela par l'adoption de l'architecture SOA (Service Oriented architecture) et en particulier la technologie des services web.

L'Architecture Orientée Services (SOA – Service-Oriented Architecture en anglais) permet aux concepteurs de systèmes d'information d'organiser un ensemble de logiciels isolés en un ensemble de services interconnectés, accessibles par une interface et des protocoles standard [Papazoglou, 2003]. un service est une unité discrète qui remplit une collection de tâches répondant au(x) même(s) objectif(s). L'SOA est un paradigme architectural qui peut être utilisé pour concevoir des infrastructures permettant aux clients et aux fournisseurs d'échanger des services, malgré la disparité des domaines, des technologies, et des fournisseurs [Nickull2005]. Les services Web sont la plate-forme de mise en œuvre prédominant pour la SOA et il utilise un ensemble de normes, SOAP (Simple Object Access Protocol), UDDI (Universal Description Discovery and Integration), WSDL(Web Services Description Language), qui permettent d'une manière flexible p our l es a pplications d 'interagir l es uns avec les autres sur les réseaux.

L'avantage principal de l'architecture orientée services en général et Web en particulier, est

que ce paradigme pourrait répondre aux besoins en termes de flexibilité et d'adaptabilité rapide exprimés par les concepteurs [Crusson2003]. Une application à base de services est flexible puisqu'un service défaillant peut être remplacé par un autre sans modifier l'ensemble de l'application. De plus, elle est adaptable du fait que le service sélectionné est celui choisi comme étant le meilleur dans un contexte donné. Avec l'intégration de ce paradigme(SOA)au domaine Business To Business (B2B), le nombre de services web devient très important. En raison de cette augmentation, la découverte et la sélection de services on fonction des exigences du client devient une opérationtrès difficile. En outre, les propriétés de qualité de service'QoS' doivent être prennent en compte dans la sélection de service Web. Le problème de sélection de service web devient un problème N-P complet, dont la nécessité d'adopter une méta heuristique approche pour résoudre ce problème.

2. **Problématique**

Avec le nombre croissant de services Web mis à disposition sur Internet et dont la fonctionnalité est similaire, comment distinguer et sélectionner le meilleur service Web à partir d'autres devient un problème majeur à résoudre. La sélection de service Web est généralement basée sur les exigences fonctionnelles du client, mais ces services Web ne peut pas être en mesure de fournir la qualité de service que le consommateur attend ! La sélection de service web est basée essentiellement sur la qualité de service (QoS), cette dernière se mesure à l'aide de plusieurs métriques tel que le Cout, le temps d'exécution, la fiabilité, la disponibilité, et la réputation. Le but de notre travail et de pouvoir sélectionner d'une façon optimale les services Web qui répondent aux exigences fonctionnelles des demandeurs qui doivent être localisé et délimité de façon dynamique à partir d'un nombre important et en constante évolution des fournisseurs de services en fonction de leur qualité de service (QoS).

3. **Contribution**

Pour une sélection optimale des services en fonction de leur (QoS), nous avons proposéune approche d'optimisation mono objective basée sur Harmony search Algorithme. Harmony search est un algorithme d'optimisation récemment développé qui imite le processus musical de recherche d'un meilleur état d'harmonie,dont le résultat est plus rapide, par rapport à d'autres approches.

4. **Plan du mémoire**

Le manuscrit est composé de trois chapitres et une conclusion générale, qui sont organisé comme suit :

Chapitre I : Nous avons présenté la technologie des services Web et les principaux standards qu'elle supporte,pourmieux comprendre les concepts de base de cette technologie, il montre aussi leurs avantages et leurs inconvénients.

Chapitre II : Dans ce chapitre, Nous avons présenté un état de l'art détaillé sur la sélection des compositions de service, tous d'abord, nous commençons par une introduction au sélectiondes services web. Ensuite, nous détaillons les différents critères de qualités de service(QoS). De plus, nous décrivons un exemple de motivation afin de conclure ce que nous avons fait.

Chapitre III : Ce chapitre est consacré à la conception, l'implémentation et l'expérimentation du prototype. Tout d'abord, nous présentons notre algorithme : Harmony search Algorithme. Après nous expliquons une application à partir d'un scénario et mentionnons les paramètres, les fonctions et la base de données utilisé, et enfin nous avons présenté la conception de notre système et l'implémentation pour évaluer la performance de l'approche proposée.

Chapitre

Les Services Web

Introduction

Les services web sont des mots tendance, et sont actuellement promus par, entre autre , SUN,ORACLE,GHP,Microsoft et IBM. C'est un mot nouveau de concept ancien car il s'agit ni plus ni moins que de déporter un traitement de données d'un poste client, vers un poste serveur sur lequel tourne l'application.

Cette évolution du monde informatique a entrainé le développement de nouveaux paradigmes d'interaction entre application tels que la SOA, cette dernière a été mise en avant afin de permettre des interactions entre application distantes.

L'architecture orientée service SOA est une méthode de conception basée sur des standards (SOAP, WSDL,...) , permettant de créer une infrastructure informatique intégrée capable de répondre rapidement au nouveau besoin d'un utilisateur.

Elle fournit les principes et directives permettant de transformer un réseau existant de ressources informatique hétérogènes, distribuées, complexes et rigides en ressources intégrées, simplifiées et particulièrement souple pouvant être modifiées et combinées afin de mieux satisfaire les objectifs de l'utilisateur.

L'objectif de ce chapitre est d'introduire le concept de "Service Web". Nous commencerons par étudier l'architecture SOA, ensuite nous présenterons le modèle des web services au travers des différents standards.

Les services web

L'adoption des **SOA** dans le monde informatique a été renforcée par plusieurs implantations. Dans le cadre de ce mémoire nous nous intéressons aux services Web, qui représentent

l'implantation la plus largement répandue.

Définition :

Un service web est un composant logiciel indépendant, qui rassemble un ensemble de standards web et de langages dérivés du langage XML, et qui permettent sa publication, sa découverte et enfin son invocation à distance. Il expose des fonctionnalités via une interface publique et permet de communiquer avec les autres applications et services web en utilisant des messages XML transportés par des protocoles internet comme le HTTP.

Définition d'IBM « *Les services web sont la nouvelle vague des applications web. Ce sont des applications modulaires, auto-contenues et auto-descriptives 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'invoquer* »[Ponge, 2004].

Définition de W3C « *Un service web est un système logiciel identifié par un URI dont les interfaces publiques et les incarnations sont définies et décrites en XML.Sa définition peut être découverte dynamiquement par d'autres systèmes logiciels. Ces derniers peuvent ensuite interagir avec le service web d'une façon décrite par sa définition, en utilisant des messages XML transportés par des protocoles Internet* » [Ricardo, 2004].

Caractéristiques des services web :

Selon [Cerami,2002],« plusieurs acteurs définissent les services web par des caractéristiques technologiques distinctives, les plus importantes sont :

- Un service web 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 service web est donc accessible en spécifiant son URI, c'est-à-dire que le service web est caractérisé par un seul objet et une seule fonctionnalité, c'est un prolongement de la programmation orientée objet et à 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.
- Capacité des interfaces et liaisons (bindings) d'être publiées, localisées et invoquées via le langage XML : les principales tâches d'un service web sont : la publication dans un

registre, la localisation en interrogeant ce registre qui l'héberge et l'invocation par un ou plusieurs web services après sa localisation. Ces tâches sont réalisées en utilisant le langage XML.

- Capacité d'interagir avec les composantes des logiciels via des éléments XML avec l'utilisation des protocoles Internet standards : un service web est créé pour être interrogé par d'autres logiciels contrairement à une page web, ou à une autre application qui n'utilise pas les services web. L'interopérabilité est basée sur l'utilisation du XML et des protocoles Internet standards, tels que, le HTTP qui est le protocole du web, le SMTP qui est le protocole du courrier électronique,. . . etc.

- Composante logicielle légèrement couplée à interaction dynamique : un service web avec un programme qui l'invoque est appelé le consommateur de service web, et qui sont indépendants l'un de l'autre. Si une modification est à faire sur le consommateur, on n'a pas besoin de connaître la machine, le langage de programmation, le système d'exploitation ou autres paramètres, afin d'établir à nouveau une communication entre le service web et son consommateur. Le consommateur possède une fonctionnalité qui consiste à faire une localisation et une invocation du service web.

Architecture des services Web :

L'interopérabilité –aptitude de deux ou plusieurs systèmes à fonctionner ensemble en utilisant des standards communs– est l'objectif premier des services web. Pour permettre cet échange d'information entre des applications distantes, indépendamment des systèmes d'exploitation et des langages de programmation, les services web sont composés de couches standards . Nous mettons en relief dans cette section deux types architectures. La première est l'architecture dite de référence et traditionnellement utilisée pour les services web isolés. La seconde architecture est plus complète et est fréquemment utilisé lors de composition de services web. Elle est appelée architecture étendue ou encore en pile.

Architecture de référence :

Cette architecture vise trois objectifs importants [H.Kreger, 2001] :

1. identification des composants fonctionnels,
2. définition des relations entre ces composants et

3. établissement d'un ensemble de contraintes sur chaque composant de manière à garantir les propriétés globales de l'architecture.

L'architecture de référence des services web (Figure 1.1) s'articule autour des trois rôles suivants :

- fournisseur de service : correspond au propriétaire du service. D'un point de vue technique, il est constitué par la plate-forme d'accueil du service.
- Le client : correspond au demandeur de service. D'un point de vue technique, il est constitué par l'application qui va rechercher et invoquer un service. L'application cliente peut être elle-même un service web.
- L'annuaire des services : correspond à un registre de descriptions de services offrant des facilités de publication de services à l'intention des fournisseurs ainsi que des facilités de recherche de services à l'intention des clients.

Les interactions de base entre ces trois rôles incluent les opérations de publication, de recherche et de liens (Bind) d'opérations. Pour garantir l'interopérabilité de ces trois opérations, des propositions de standards ont été élaborées pour chaque type d'interaction. Nous citons, notamment les standards émergents suivants :

- **SOAP** définit un protocole de transmission de messages basé sur XML.
- **WSDL**Introduit une grammaire commune pour la description des services.
- **UDDI** fournit l'infrastructure de base pour la publication et la découverte des services.

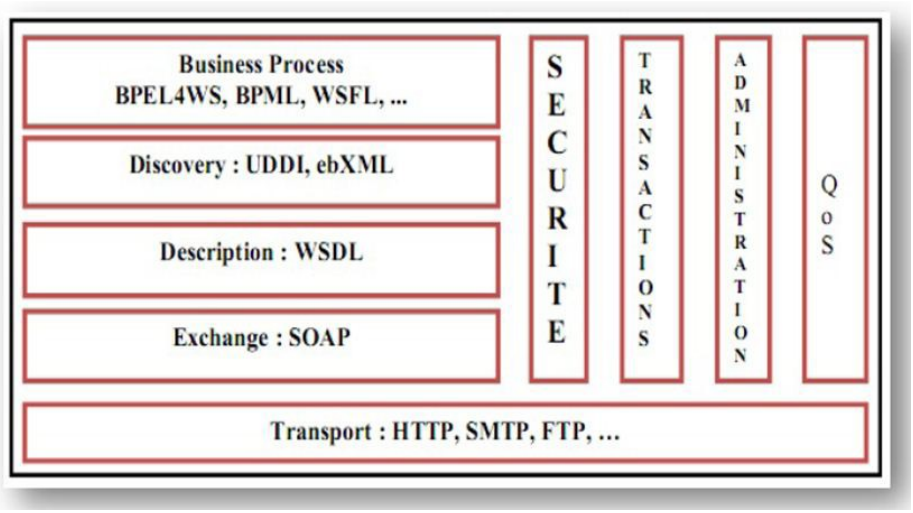

Figure 1.1 – Architecture de référence des services Web.

Cependant, cette infrastructure n'est pas suffisante pour permettre une utilisation effective des services web dans les domaines dont les exigences vont au-delà de la capacité d'interactions simples via des protocoles standards, par exemple, dans le domaine du e-business, d'ou la nécessité d'introduire une nouvelle architecture suffisante et complète.

Architecture étendue :

Une architecture étendue est constituée de plusieurs couches se superposant les unes sur les autres, d'où le nom de pile des services web. 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.

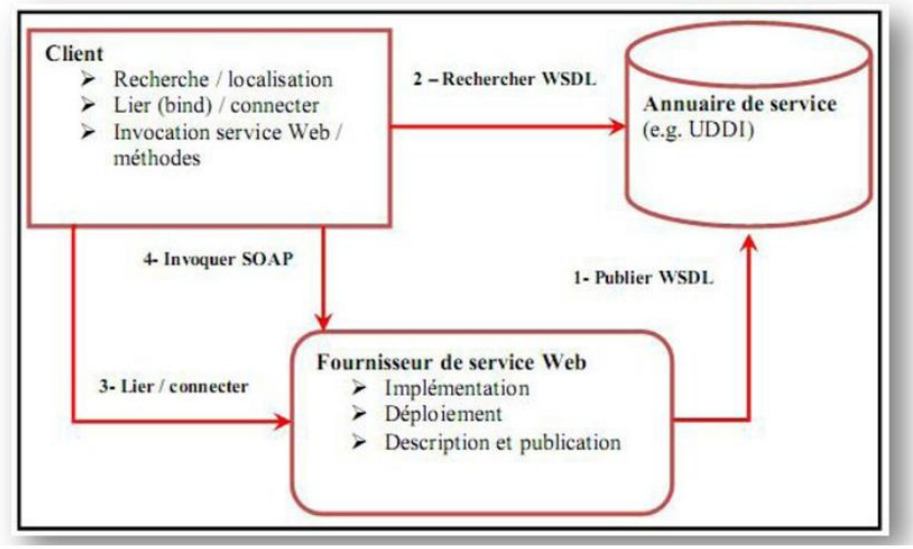

Figure 1.2 – Architecture en Pile des services Web.

Nous apportons une explication de la mise en relief des trois types de couches (Voir la Figure 1.1) : **L'infrastructure de base (Discovery,Discription, Exchange) :** ce sont les fondements techniques établis par l'architecture de référence. Nous distinguons les échanges des messages établis par SOAP, la description de service par WSDL et la recherche de services Web que les organisations souhaitent utiliser via le registre UDDI ; **Couches transversales (Security, Transactions, Administration, QoS) :** Ce sont ces couches qui rendent viable l'utilisation effective des services Web dans le monde industriel ; **La couche Business Processus (BusinessProcess) :** Cette couche supérieure permet l'intégration de services Web, elle établit la représentation d'un « BusinessProcess » comme un ensemble de services Web. De plus, la description de l'utilisation de différents services composant ce service est disponible par l'intermédiaire de cette couche.

Cycle de vie d'un service web :

L'emploi d'un service web connaît le cycle de vie suivant [J.Garcia,] :

- D'abord, on effectue le déploiement du service web, en fonction de la plateforme.
- Ensuite, on enregistre le service web à l'aide de WSDL (Web Services Description Language), dans l'annuaire UDDI.
- L'étape suivante est la découverte du service web par le client, par l'intermédiaire d'UDDI qui lui donne accès à tous les services web inscrits. Pour ce, on utilise SOAP.
- Enfin, Le client invoque le service web voulu, ce qui termine le cycle de vie de ce service web.

Les langages et protocoles utilisés par les services web :

Nous avons choisi de décrire les différentes technologies des services web en nous basant sur leur architecture. Toutes ces technologies sont basées sur XML.
Nous avons choisi de décrire les différentes technologies des services web en nous basant sur leur architecture. Toutes ces technologies sont basées sur XML.

Echange avec SOAP-Simple Object Access Protocol : [Ram-pacek, 2006]

Les communications entre les différentes entités impliquées dans le dialogue avec le service web se font par l'intermédiaire du protocole SOAP (Simple Object Access Protocol). Ce protocole est normalisé par le W3C [Nilo.Mitra, 2002], [M.Gudgin and Henrik.F, 2002]. Le protocole SOAP est une sur couche de la couche application du modèle OSI des réseaux. Le protocole applicatif le plus utilisé pour transmettre les messages SOAP est HTTP, mais il est également possible d'utiliser les protocoles SMTP ou FTP, la norme s'impose pas de choix.
Le choix de l'utilisation de protocoles applicatifs, comme par exemple HTTP, est lié aux problèmes d'interconnexion connus des réseaux. Le but des services web étant d'être réutilisables, après publication, à travers tout le réseau Internet, il faut alors donner les moyens

de passer les protections telles que les pare-feux (firewalls). Ces derniers autorisent généralement sans aucune restriction le trafic sur les ports liés aux protocoles tels que HTTP, permettant ainsi le passage sans problème à travers les différents réseaux des messages générés par l'utilisation du protocole SOAP.

Structure d'un message SOAP :

La technologie des services web repose principalement sur le protocole SOAP qui est indépendant des langages de programmation ou des systèmes d'exploitation. Le standard SOAP définit trois éléments composants un message : [GARDIEN, 2002] L'enveloppe (Enveloppe), l'en-tête du message (Header) et le corps du message (Body) (voir Figure 1.3)

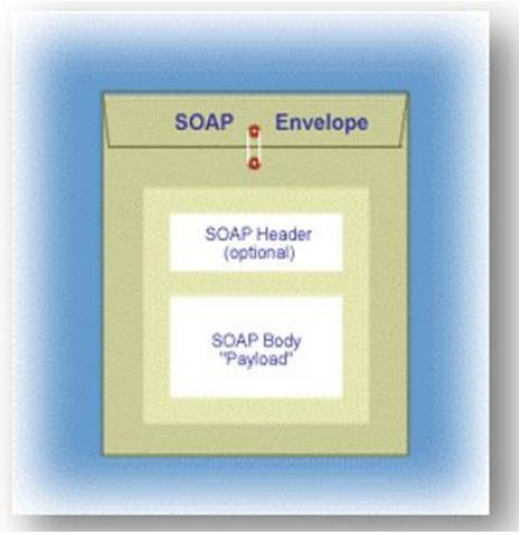

Figure 1.3 – Structure d'un message SOAP.

- Enveloppe SOAP : L'enveloppe SOAP est l'élément obligatoire d'un message SOAP qui englobe les deux autres parties de ce message (Header et Body). Elle contient le nom du message et le nom du domaine, et permet de préciser la version de SOAP utilisée.

- L'En-tête SOAP : L'en-tête est un élément facultatif dans un message SOAP, marqué par la balise <Header>. Il peut avoir plusieurs fils. Ces fils sont utilisés pour ajouter des fonctionnalités au message SOAP comme l'authentification et la gestion des transactions [Ricardo.R, 2004]. L'en-tête peut utiliser les attributs mustUnderstandet /ou SOAP actor pour déterminer comment le destinataire d'un message doit traiter ce dernier.

  – L'attribut acteur de SOAP peut être utilisé pour indiquer le destinataire d'un élément d'en-tête. La valeur de l'attribut d'acteur de SOAP est un URI.

- L'attribut mustUnderstand peut être utilisé pour indiquer si une entrée d'entête est obligatoire ou facultative pour être traitée par le destinataire. La valeur de l'attribut de mustUnderstand est "1" ou "0". L'absence de cet attribut est sémantiquement équivalente à sa présence avec la valeur "0".

- Le corps SOAP : [GARDIEN, 2002] Le corps du message SOAP est obligatoire, marqué par la balise <Body>. Il permet de transmettre les requêtes et les réponses entre les systèmes, il est composé d'un ou plusieurs sous éléments, qui sont :

  - *L'élément FAULT :* il permet d'indiquer les défaillances de transmission des messages SOAP. Il renvoie des informations sur le type d'erreur, une description de l'erreur et l'adresse du serveur SOAP qui a généré l'erreur.
  - *L'élément MESSAGE :* il contient les données à transmettre via le protocole SOAP.

Modèle d'échange de messages en SOAP :

Les messages SOAP sont des transmissions fondamentalement à sens unique d'un expéditeur à un récepteur. Lorsqu'une transmission d'un message commence (ex. invocation d'un service web), un message SOAP (ou document XML) est généré. Ce message est envoyé à partir d'une entité appelé le SOAP Sender, localisé dans un SOAP noeud. Le message est transmis à zéro ou plusieurs noeuds intermédiaires (SOAP intermédiaires) et le processus fini lorsque le message arrive au SOAP receiver. Le chemin suivi par un message SOAP est nommé message path, [Ricardo.R, 2004].La Figure 1.4 montre les éléments qui participent au processus.

En résumé, le SOAP est un protocole de communication entre les applications fondé principalement sur le protocole HTTP et sur XML, visant à satisfaire un double objectif : servir de protocole de communication sur Internet, dans une optique d'intégration d'application, et permettre la communication entre les applications et les services web. Il définit un ensemble de règles pour structurer les messages envoyés. Mais SOAP ne fournit aucune instruction sur la façon d'accéder aux services web. C'est le rôle du langage WSDL.

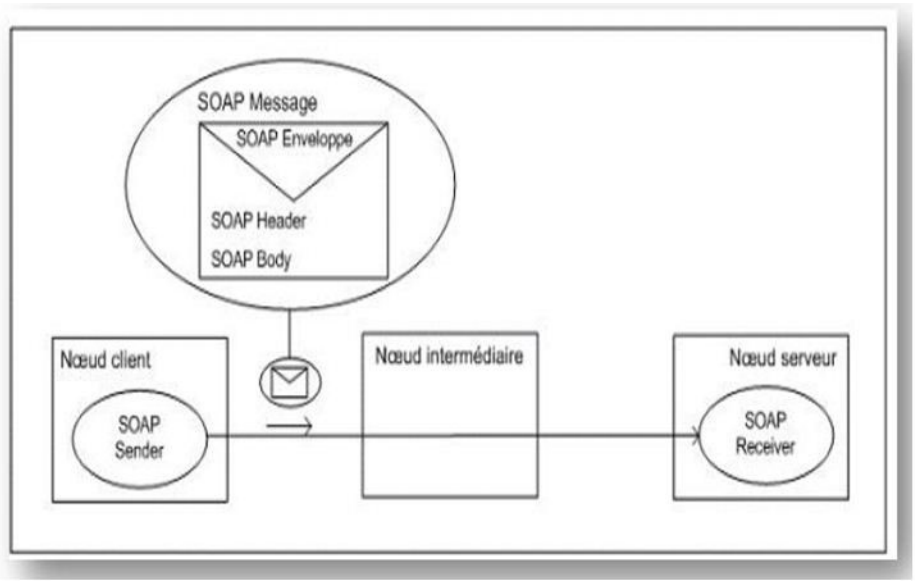

Figure 1.4 – figure
Modèle d'échange de message en SOAP.

Description avec WSDL-Web Services Description Language : [Rampacek, 2006]

Le WSDL (Web Services Description Language) est un langage basé sur XML, crée dans le but de fournir une description unifiée des services web. Il se présente comme un standard actuel dans ce domaine, et il est, de plus, normalisé par le W3C. Il est issu d'une fusion des langages de descriptions NASSL (Network Accessibility Service SpecificationLanguage - IBM) et SCL (Service Conversation Langage - Microsoft).Son objectif principal est de séparer la description abstraite du service de son implémentation même.

La structure d'un document WSDL ; [Ricardo.R, 2004]

Le langage de description WSDL se comporte donc comme un langage permettant de décrire l'interface visible (ou publiée) du service web. Il décrit, à l'aide du langage de balises XML, les différents éléments du service, [Rampacek, 2006]. Une description WSDL d'un service web est faite sur deux niveaux, un niveau abstrait et un niveau concret. Au niveau abstrait, la description du service web consiste à définir les éléments de l'interface du service web tel que les types de données (Data Types), les messages (Message), les types de ports (Port Type). Ces parties décrivent des informations abstraites indépendantes au contexte de mise en oeuvre. On y trouve les types de données envoyées et reçues, les opérations utilisables et le protocole qui sera utilisé.

- **Data types :** est l'élément qui définit les types de données utilisées dans les messages

échangés par le service web.

- **Message :** spécifie les types d'opérations supportées par le service web, il permet d'incorporer une séquence de messages corrélés sans avoir à spécifier les caractéristiques du flux de données.

- **Port Type :** est un groupement logique ou une collection d'opérations supportées par un ou plusieurs protocoles de transport, il est analogique à une définition d'un objet contenant un ensemble de méthodes. Au niveau concret, le service web est défini grâce aux éléments Bindings et Service et Port. Ces deux derniers décrivent des informations liées à un usage contextuel du service web. On y trouve l'adresse du fournisseur implémentant le service, et le service qui est représenté par les adresses des fournisseurs.

- **Bindings :** Décrit la façon dont un type de port est mis en oeuvre pour un protocole particulier (HTTP par exemple), et un mode d'invocation (SOAP par exemple). Cette description est faite par un ensemble donné d'opérations abstraites. Pour un type de port, on peut avoir plusieurs liaisons, pour différencier le mode d'invocation ou de transport de différentes opérations.

- **Port : Spécifie** une adresse URL qui correspond à l'implémentation du service web par un fournisseur et identifie une ou plusieurs liaisons aux protocoles de transport pour un Port Type donné.

- **Service :** Spécifie l'adresse complète du service web, et permet à un point d'accès d'une application distante de choisir à exposer de multiples catégories d'opérations pour divers types d'interactions. En résumé le document WSDL est indispensable au déploiement de services web et décrit deux documents essentiels : un document pour l'interface du service web et un autre pour son implémentation. La publication et la recherche d'un service web au sein de l'annuaire UDDI se font via ces deux types de documents WSDL.

Recherche avec UDDI(Universal Discovery Description and Integration) :

Un service web doit être référencé afin de pouvoir être retrouvé et utilisé par une autre organisation. Pour cela, il existe des annuaires pouvant être soit internes à l'organisation, soit universels. Il existe de nombreux annuaires, mais nous choisissons de décrire seulement

le registre UDDI, annuaire dit universel. L'UDDI définit les mécanismes permettant de répertorier des services Web. Ce standard gère donc, l'information relative à la publication, la découverte et l'utilisation d'un service Web. En clair, l'UDDI définit un registre des services Web sous un format XML. Les organisations publient les informations décrivant leurs services Web dans l'annuaire, et l'application client ayant besoin d'un certain service, consulte cet annuaire pour la recherche des informations concernant le service Web qui fourni le service désiré, pour une éventuelle interaction ; ainsi l'UDDI a été créé pour faciliter la découverte de services Web en plus de leurs publications. Par une API SOAP, on peut interagir avec l'UDDI au moment de la conception et l'exécution des applications afin de découvrir des données techniques, et administratives sur les entreprises et leurs services Web.
L'annuaire UDDI repose sur le protocole SOAP, les requêtes et les réponses sont des messages SOAP [Scott, 2002] [GARDIEN, 2002]. L'UDDI est subdivisé en deux parties principales : *partie publication* ou *inscription*, et *partie découverte*. *La partie publication* regroupe l'ensemble des informations relatives aux entreprises et à leurs services. Ces informations sont introduites via une API d'enregistrement. *La partie découverte* facilite la recherche d'information contenue dans UDDI grâce à l'API SOAP. L'UDDI peut être vu comme un annuaire contenant [Newcomere, 2004] :

- **Les pages blanches :** noms, adresses, identifiants et contacts des entreprises enregistrées. Ces informations sont décrites dans des entités de type « BusinessEntity ». Cette description inclut des informations de catégorisation permettant de faire des recherches spécifiques dépendant du métier de l'entreprise.

- **Les pages jaunes :** donnent les détails sur le métier des entreprises et les services qu'elles proposent.Ces informations sont décrites dans des entités de type « BusinessService ».

- **Les pages vertes :** contiennent des informations techniques du service offert, la manière d'interagiravec le service, une définition du « BusinessProcess » et aussi un pointeur vers le fichier WSDL etune clé unique identifiant le service.

La structure de données du registre UDDI contient cinq types de données :

- **BusinessEntity :** Les informations concernant l'entreprise qui publie ses services Web dans l'annuaire et le type de services qu'elle offre sont contenues dans la structure« BusinessEntity »et correspondent notamment aux pages blanches concernant l'entreprise.

- **BusinesService :** L'élément « BusinessService »contient des informations sur les services métiers. Il s'agit des informations de description des services web référencés : nom du service, sa description, son code. Une entreprise peut enregistrer plusieurs services. Le type d'informations contenu dans l'élément « BusinessService »correspond aux informations pages jaunes d'une entreprise.

- **BindingTemplete** (informations de liaison) : Les informations de liaison contiennent des informations techniques sur un service Web. Elles servent également de pages vertes de l'annuaire UDDI. Il s'agit des informations indiquant les références aux « tModels »désignant les spécifications de l'interface pour un service Web, ainsi le point de terminaison (adresse Internet) de ce dernier. Ces informations aident le client à se connecter puis à invoquer le service désiré. Un service peut avoir plus d'un modèle de liaison.

- **tModel (informations techniques) :** La structure « tModèle »a pour rôle de décrire les spécifications des services Web à enregistrer. Elle comporte les informations permettant de connaître les normes auxquelles le service est conforme afin d'avoir une description suffisamment précise du service. Le nom doit présenter suivant le format URI et doit pointer vers l'emplacement du fichier correspondant, généralement au format WSDL.

- **PublishAssertion (Assertion contractuelles entre partenaires) :** met en correspondance deux ou plusieurs structures « BusinessEntity »selon le type de relation (filiale de, département de).Cette structure comporte les assertions contractuelles entre organismes pour les services publiés. Ces assertions représentent un ensemble de règles contractuelles d'invocation de services représentées sous la forme de protocoles entre deux partenaires métiers. Chaque rôle entre partenaires est défini à travers ces assertions. En résumé, UDDI est un standard pour faciliter la collaboration entre partenaires dans le cadre d'échanges commerciaux, le coeur d'UDDI est un annuaire qui contient des informations techniques et administratives sur les entreprises et les services Web qu'elles publient. Donc l'annuaire UDDI permet de publier et découvrir des informations qui concernent une entreprise et ses services Web.

Avantages et Inconvénients des services web :

Avantages :

La technologie des services web est populaire et couramment utilisée car elle offre des avantages intéressants pour les utilisateurs des systèmes distribués :

- Les services web réduisent le temps de mise en marché des services offerts par les diverses entreprises.
- Les services web permettent à des programmes écrits en des langages différents et sur des plateformes différentes de communiquer entre eux par le biais de certaines normes. En d'autres termes, les services web permettent une meilleure interopérabilité entre les logiciels.
- Les services web utilisent des normes et protocoles ouverts.
- Grâce au protocole HTTP, les services web peuvent fonctionner malgré les pare-feu sans pour autant nécessiter des changements sur les critères de filtrage.
- Les protocoles et les formats de données sont offerts, le plus possible, en format texte pour que la compréhension du fonctionnement des échanges soit plus intuitive.
- Grâce aux services web, les coûts sont réduits par l'automatisation interne et externe des processus commerciaux.

Inconvénients :

La technologie des services web comporte plusieurs inconvénients dont :

- Problèmes de sécurité : Il est facile de contourner les mesures de sécurité mises en place par les pare-feu -l'utilisation du protocole HTTP (tel que mentionné ci-haut) n'a pas que des avantages -car les normes de sécurité des services web laissent encore à désirer. CORBA, par exemple, qui est une technologie plus mûre, est plus sécuritaire.
- 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.
- 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.

- Syntaxe et sémantique : On se concentre beaucoup sur comment invoquer des services (syntaxe) et pas assez sur ce que les services web offrent (sémantique).
- 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.
- 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.

1.6 Conclusion

La technologie des services Web offre de fortes potentialités pour surmonter les problèmes d'interopérabilité des systèmes. Elle constitue un cadre prometteur pour l'intégration des applications, et pour la gestion des interactions entre divers partenaires dans un environnement distribué, hétérogène, ouvert et versatile qui est le Web. La plupart des travaux existants qui s'intéressent à l'intégration fonctionnelle évitent le problème fondamental de l'automatisation des différentes étapes liées à la fourniture d'un service web (par exemples, découverte et composition) puisqu'ils limitent l'usage des services web aux utilisateurs humains plutôt qu'aux machines.

Chapitre 2

La sélection des service web

Introduction :

Les services Web sont aujourd'hui parmi les groupes les plus largement utilisés pour Service Oriented Architecture (SOA). La Sélection de service est l'un des débats actuels les plus importants dans la SOA, qui évalue les services découverts et choisit le meilleur. La sélection de service Web apparaît quand il y'a un ensemble de services Web découverts qui peut remplir les besoins des utilisateurs, et l'un de ces services devrait être sélectionné pour être retourné au consommateur de services [R.Chinnici2003,]. Il est essentiel que cette sélection soit adaptée aux préférences de l'utilisateur en raison du fait qu'un utilisateur peut exiger de haute qualité tandis que l'autre peut exiger bas prix. Nous montrons le processus de découverte de service, choix et la composition dans [Figure 2.1].

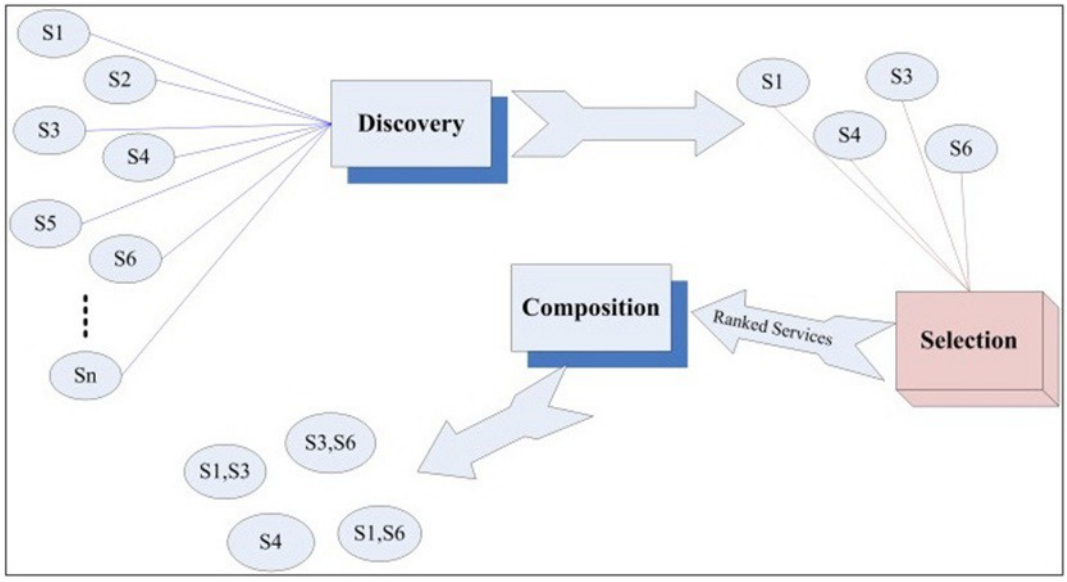

Figure 2.1 – Discovery, selection and composition of services..

La découverte. Les services sont conçus afin d'être sélectionnés via des mécanismes derecherche. La découverte est permise par la description préalable des services et leur publicationau sein d'un registre. La découverte est un point clé nécessaire à la réutilisation des servicesélémentaires dans une SOA.

La composition. Des collections de services peuvent être coordonnées et assemblées afin deformer une composition de services. Cette possibilité de construire de nouveaux systèmes àpartir de services existants constitue un des avantages de l'SOA.

Définition :

La sélection de service Web est l'une des discussions les plus importantes dans la SOA, ce qui signifie pour identifier les meilleurs services parmi un groupe de services ayant des fonctions similaires, mais ayant différents Qualité de Service (QoS) [R.Chinnici2003,]. QoS est important que les mesures de qualité doivent être atteintes par le biais de service exigée. Ces indicateurs sont mesurables et comprennent ce que le service est offert [Geem, 2008].

Sélection de services web basée sur la qualité de service :

La sélection consiste à choisir, parmi les services web découverts, ceux qui répondent au mieux aux exigences de l'utilisateur sur la base des besoins fonctionnels et/ou non fonctionnels. Les besoins non fonctionnels des services web sont généralement exprimés à l'aide des critères de QoS. Dans une sélection de services web basée sur la QoS deux cas se présentent [Jaeger and Mühl, 2006] : 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 répondent mieux aux besoins des clients. Pour le deuxième cas nous sommes face à un problème de sélection de services web composants basée sur la QoS. La modélisation de ce problème dépend de la nature de la stratégie de sélection : locale ou globale. Au cours des dernières années, il ya un intérêt croissant dans la sélection de service basé sur QoS. Problèmes de sélection de services voix sur QoS peuvent être résolus par des méthodes telles

que la programmation linéaire, MCDM et logique floue .Cependant, dans plusieurs études, les méthodes hybrides ont été utilisées pour résoudre les problèmes de sélection de services. La majorité des techniques de sélection de services ont été appliquées, et les caractéristiques de sélection de service basé sur le QoS permettre aux chercheurs de faire face aux problèmes de sélection de service par le multi-critères prise de décision (MCDM). Un certain nombre d'approches a utilisé la méthode de MCDM pour la sélection de service. Par exemple, AHP , l'ANP , et PROMETHEE ,ont été appliqués pour la sélection de service. Le problème de sélection de service dans SOA a été résolu de diverses manières, grâce à quoi MCDM est l'une des solutions. Cependant, les différentes méthodes de MCDM créent souvent des résultats différents, en particulier lorsque les différences entre les solutions de rechange sont ensemble intrinsèquement proche pour le classement d'un ensemble de décisions alternatives impliquant de multiples critères. Plusieurs chercheurs ont suggéré d'appliquer différentes méthodes de MCDM simultanément pour fournir un outil plus efficace afin d'améliorer la précision de la décision finale. Par conséquent, il est nécessaire de développer une procédure scientifique plus systématique et logique pour aider les concepteurs de services Web pour réaliser la conception de Web optimale.

Critères pour les méthodes de sélection des services :

Les approches utilisées pour la sélection de service Web sont étudiées par rapport à un ensemble de caractéristiques, et sont brièvement décrites comme suit :

- **Qualité de service :** Ceci se rapporte à des approches qui tiennent compte de la qualité de service que le critère de prise de décision. Les QoS sont répandues Durée (temps d'exécution), la disponibilité, fiabilité et de coût. Ceci est un sujet de la plus haute importance, qui doit être pris en considération.

- **La préférence de l'utilisateur :** Ceci fait référence aux approches qui traitent de préférences de l'utilisateur afin de rendre compte de la priorité des consommateurs de services. Par exemple, l'importance relative des critères dans une matrice de décision peut être obtenue par la préférence du demandeur de service.

- **Evolutivité :** Ceci se rapporte à des approches qui tiennent compte de nombreuses propriétés et les processus de classement qui se produisent en même temps, tout en conservant la précision des résultats. Dans certains procédés, la précision du procédé

est influencée par le nombre de services de remplacement ou l'augmentation du nombre de critères.

- **Automatique :** L'élément essentiel dans la sélection automatique de service réside dans l'étape finale. Quand un service est disponible, le concepteur de service spécifie les données pour le service et l'utilisateur spécifie les exigences. Cependant, l'intervention humaine est absente lors de la sélection de service.

Qualité de Service (QoS) :

La définition des attributs de QoS dans le processus de sélection de service sont prévues comme suit :

Temps de latence : également appelé Temps d'exécution ou de performance, est la vitesse à laquelle une demande de service peut être complétée, ce qui est un élément essentiel pour les services Web. La durée du temps d'attente et d'exécution sont nécessaires pour estimer P. Le temps d'attente est la durée des activités, telles que le transfert d'un message, et le temps d'exécution est la durée d'exécution de la fonctionnalité d'un service. Il est :

- Quantifié par : Seconde.
- Formulé par : (Le temps de réception de la réponse par le client) – (Le temps d'envoi de la requête par le client).

Disponibilité : est le taux se réfère à la façon d'accéder au service en tout temps. Supposons qu'un service est sélectionné comme la sélection finale, mais quand l'utilisateur veut accéder à certains services qui ne sont pas disponibles. Par conséquent, tous computing à propos de la sélection doit être répété à l'amende l'autre service Web. Cet attribut est appelé disponibilité.

Fiabilité : est la capacité d'un service pour réaliser ses tâches et fonctions demandées. La capacité de la SP à demander de livrer la fonctionnalité du service avec succès est la fiabilité du service web. La probabilité de réussite lors d'une exécution de service définit la quantité de cette capacité. Toutefois, le taux d'un service de défaillance détermine généralement la fiabilité. Le taux est évalué comme le rapport entre le temps d'exécution pour le temps moyen entre défaillances. Le temps d'exécution peut être en conflit parce qu'il est le temps nécessaire pour effectuer un service et aussi le temps nécessaire pour fournir un résultat du point de vue du demandeur de service ; cependant, parce que la SP est pas capable de

supporter le problème de réseau, temps d'exécution est considéré comme le temps nécessaire d'effectuer un service. Elle est :

- Quantifié par : Pourcentage.
- C'est le nombre de fois qu'une opération a été exécutée avec succès ; le nombre total d'appels.

Coût : également appelé financière ou le cours, concerne le coût et les frais liés à un service. Le coût de la demande et l'utilisation de chaque service est le prix de service Web. Le prix des services est affecté par la valeur de la fonctionnalité. Fournir des fonctions plus complexes augmente le coût du service.

Réputation : est une mesure de la crédibilité de l'entreprise.

Motivation

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.

- Il existe différents types de services (calcul, stockage, etc.).
- Il existe un grand nombre de services avec fonctionnalités similaire qui se traduit par une prolifération du nombre de des 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 AppEngine.
- 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 différente .

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.

Exemple de motivation :

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 l'utilisateur exige des contraintes globales sur chaque critère de QOS, par exemple le coût total des 03 services, ne doit pas excéder une certaine limite.[Hadjila 2014]

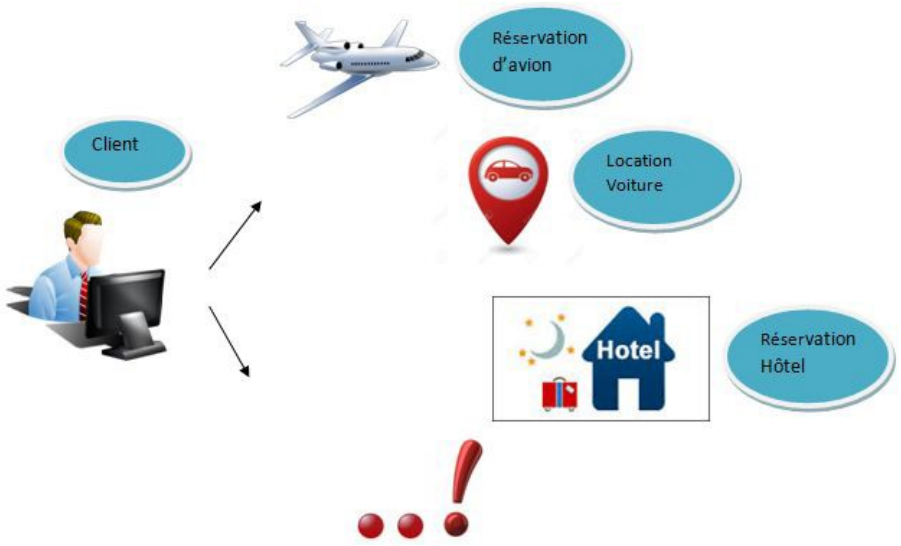

Figure 2.2 – Scenario de motivation. [Liu et al., 2011]

Il est clair que ce problème est NP-Hard, en effet il est équivalent au MMKP(multi-dimensional multiple choiceknapsackproblem) [Pisinger, 1995]. Les solutions exactes de ce problème ont une complexité exponentielle, et de ce fait nous ne pouvons plus garantir les exigences temps réelles, ou le passage à l'échelle.

Formalisation du Problème :

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

$$CA = \{S1, \ldots, Sn\} \tag{2.1}$$

une composition abstraite qui représente la requête de l'utilisateur, cad les n classes de services à consommer Si.

$$CONS = \{c1, \ldots, cm\} \tag{2.2}$$

Un ensemble de contraintes globales définies par l'utilisateur.

$$C = \{s1, \ldots, sn\} \tag{2.3}$$

une composition concrète, cad 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 Uk(C) sont optimisées. (Uk représente la fonction objective

du K eme attribut de QOS. Si on agrège les m attributs de QOS en une seule valeur alors nous aurons besoin d'une seule fonction objective U.) les contraintes globales sont vérifiées,i.e. $Q'k(C) \geq ck$, $\forall ck \in CONS$ Avec Q'k(C), est la valeur du Keme 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. L'énumération totale de ces compositions ne peut être faite en temps raisonnable, en plus la présence des contraintes globales, impose un temps exponentiel pour avoir une solution exacte.

État de l'Art

La sélection de services, a fait l'objet de plusieurs travaux, de façon générale on distingue 03 grandes classes (voir la figure V.2. [F.Hadjila, 2013].[Hadjila 2014] : La sélection mono objective, La sélection multi objective, La sélection hybride (mono et multi objective)

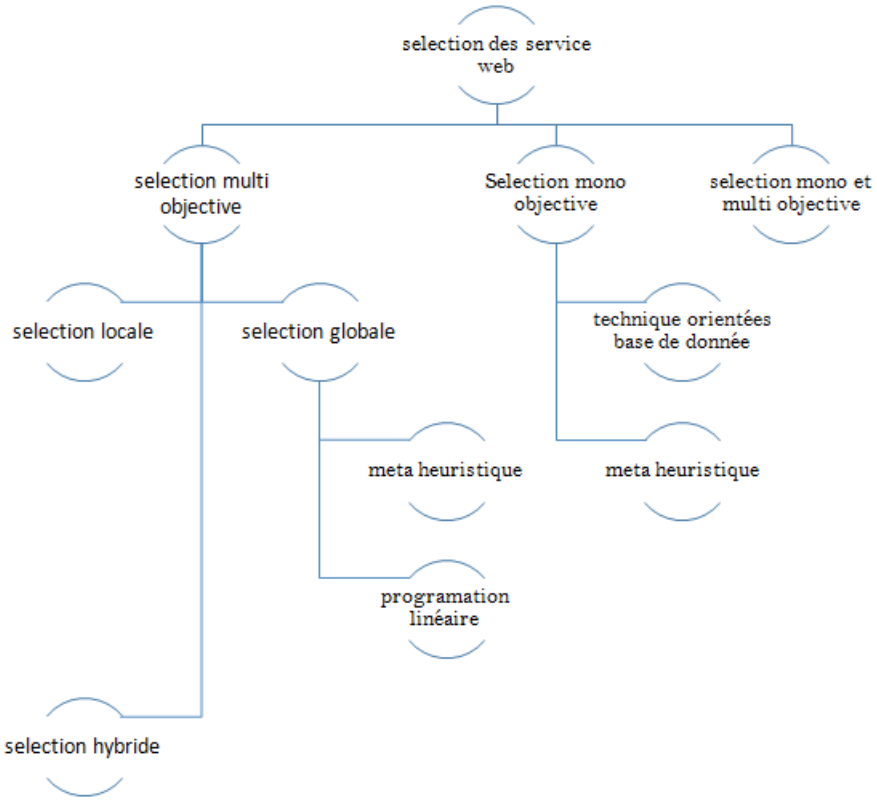

Figure 2.3 – Les différentes classes de sélection de services..

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).

- Sélection Globale Ces approches explorent un espace de recherche dont les nœuds sont des compositions complètes (c.à.d. contenant toutes les tâches), on distingue deux sous classes d'optimisation globale : exacte et approximative.

- Optimisation Exacte Elle se base sur la programmation entière ou la programmation par contrainte, ou les énumérations exhaustives, ces méthodes donnent des résultats optimaux mais elles ont un temps d'exécution exponentiel. [Zeng et al., 2003] ; [L.Zeng, 2004] ; [Ardagna and Pernici, 2005] ; [Ardagna and Pernici, 2007] ; [Kritikos and Plexousakis, 2009],. Les travaux de Zeng et ses collègues [L.Zeng, 2004] utilisent les techniques de programmation entière et mixtes (MixedIntegerProgramming ou MIP) [] pour trouver la composition optimale des services. Dans la même optique [Ardagna and Pernici, 2005], étendent le modèle de programmation linéaire afin d'inclure les contraintes locales. Dans ce modèle, les contraintes globales sont exprimées sur la composition entière, et par l'utilisateur final, tandis que les contraintes locales peuvent être spécifiées par le concepteur de la composition (au niveau des classes). [L. Xu,] Proposent une méthode de sélection de services web à base QOS en considérant une variation de valeurs de QOS en fonction de l'intervalle de temps (les fluctuations sont données par les fournisseurs de services). Pour cela l'utilisateur introduit les poids et l'intervalle d'intérêt qui peut englober plusieurs intervalles ayants des valeurs de QOS différentes, mais ils ne gèrent pas les contraintes globales, Pour cela, ils utilisent la programmation dynamique qui découpe la composition en une série de paires de services (02 classes consécutives), l'algorithme cherche la paire concrète qui minimise la fonction objective.

- Optimisation Approximative

  Cette catégorie consiste à explorer une partie de l'espace de recherche, en adoptant des recherches heuristiques ou des Meta heuristiques, les Meta- heuristiques [Dréo et al., 2006] sont des algorithmes d'optimisation génériques, qui adoptent une recherche locale ou globale, elles permettent de donner des résultats quasi-optimaux, tout en ayant un temps d'exécution abordable. Par opposition, les recherches heuristiques ne peuvent être généralisées pour le reste des problèmes. Plusieurs travaux sont inspirés des algorithmes heuristiques. propose l'un des premiers algorithmes pour la résolution du problème de sac à dos (et sa version MMKP12). L'auteur propose une heuristique

nommée UHE pour sa résolution, UHE utilise une mesure appelée la consommation des ressources agrégés, pour mettre à jour un élément de chaque groupe à chaque tour de sélection. Dans [Akbar et al., 2001] les auteurs modifient l'heuristique en créant M-UHE, ils proposent une étape de prétraitement de trouver une solution réalisable et une étape de post-traitement pour améliorer la valeur totale de la solution, pour cela ils font des améliorations (upgrades) en mettant à jour un élément qui augmente l'utilité totale, ensuite ils dégradent la solution avec un ou plusieurs coups (downgrades), i.e. ils sélectionnent un élément qui diminue la valeur de l'utilité totale). [Mostofa Akbar et al., 2006] proposent une autre heuristique, C-UHE, et évaluent sa performance et son optimalité contre plusieurs heuristiques, y compris la M-UHE. Les résultats montrent que la C-UHE est meilleure par rapport à M-UHE en termes de temps d'exécution. Cependant, les expériences montrent également que M-UHE est la meilleure en termes de degré d'optimalité, tandis que l'optimalité de C-HEU diminue à mesure que le nombre de candidats par classe augmente. Les expériences montrent que l'algorithme C-UHE fonctionnemieux dans le cas où l'objectif à maximiser (par exemple, la valeur d'utilité de la composition de services) n'est pas proportionnel aux besoins en ressources (i.e., les valeurs de QoS des services Web). Et de ce fait elle ne peut être appliquée pour la sélection de services composés à base de qualité (puisque l'utilité dépend des qualités de service). Dans [Zhang and Ren, 2011], les auteurs proposent une normalisation des valeurs de QOS légèrement différente par rapport à [Alrifai et al.,], [Alrifai et al., 2010], [Alrifai et al., 2012], ils utilisent un algorithme génétique adaptative, i.e. que les taux de mutations et de croisement dépendent de l'affinité des compositions (leur fitness). Plus l'affinité est grande, plus les taux de croisements et de mutation sont faibles, en plus les auteurs introduisent quatre constantes pour contrôler ces taux. Les expérimentations montrent que le temps d'exécution est moins de 40 sec mais la base considérée est très petite (03 classes et moins de 500 services).

Dans [G. Canfora, 2005] et [J. Wang, 2005] les auteurs présentent un algorithme génétique qui instancie les services abstraits avec des composants concrets. Dans les deux travaux, Les points de mutation sont choisis aléatoirement .Le codage des chromosomes utilise la représentation entière pour [Canfora et al., 2005] et la représentation binaire pour [J. Wang, 2005].

Les algorithmes génétiques souffrent des problèmes de scalabilité, (le codage des chromosomes est fixe), en plus la possibilité de se stagner dans un optimum local est toujours présente.

- Sélection Locale Elle consiste à élire un seul service de chaque classe en utilisant une fonction objective (et indépendamment des autres classes), ensuite elle compose les n résultats qui correspondent aux n classes.
  Cette approche possède une complexité linéaire O(l) [Alrifai et al., 2012], en plus elle est fortement adaptée aux environnements distribués, en effet la gestion de la QOS (mesures, mise à jours . . . est faite par des facilitateurs distribués). [B. Benatallah and Dumas, 2002], [F.Li and S.Su., 2007]. Mais nous notons en contrepartie, que ce type de méthodes ne prend pas en charge les contraintes globales (ex : le coût global de la composition), ce qui la rend obsolète pour les problèmes réels. [Alrifai et al.,] adaptent la sélection locale pour gérer les contraintes globales. Pour cela ils divisent les contraintes globales, en contraintes locales en se basant sur la distribution statistiques des valeurs de QOS. Les auteurs gèrent uniquement la séquence.
  Dans [K. Benouaret, 2011] et [K. Benouaret and Hadjali, 2011] Les auteurs considèrent la sélection de service web en prenant en compte les préférences sur les sorties de services, pour cela ils appliquent un tri local (pour chaque classe) en adoptant la notion de fuzzy dominance, cette dernière permet la comparaison de 02 services de manière plus efficace que la notion de pareto dominance. Enfin ils retiennent les K premières solutions. Les auteurs ne gèrent ni la qualité de service, ni les contraintes globales.

- Sélection Hybride [M.Alrifai, 2009] les auteurs transforment le problème d'optimisation globale (exacte), en un problème d'optimisation approximative. (Moins gourmand en termes de temps).
  En effet les méthodes d'optimisation globales et exactes dépendent fortement du nombre de services de la collection de test, puisque nous associons une variable binaire, pour chaque service (n*l variables sachant que n est le nombre de classes, et l est le nombre de services par classe). Ceci explique le caractère exponentiel et le non scalabilité de ces approches. Pour pallier à ce problème, les auteurs proposent deux phases d'optimisation, une phase globale qui considère des intervalles de valeurs de Qos comme variables à la place des services directs, le problème revient à sélectionner des bornes b'zjk, pour chaque critère de Qos et pour chaque classe de telle sorte que la majorité des effectifs de la classe est retenu après optimisation, en plus de la préservation des contraintes globales de l'utilisateur. Pour cela, les auteurs divisent chaque intervalle de Qos de chaque classe, en d sous intervalles, ensuite ils choisissent un seul sous intervalle pour l'utiliser

au niveau de l'étape suivante. Pour sélectionner les intervalles, l'approche adopte le modèle MIP [Nemhauser et Wolsey, 1988] (la programmation en nombre entiers ou Mixed IntegerProgramming). Le modèle MIP doit maximiser la fonction suivante (i.e. choisir des intervalles d qui ont une forte chance pour avoir des services optimaux et vérifiant les contraintes globales)

- Sélection Multi-Objectives Plusieurs travaux, ont été proposés pour la recherche multi objectives des services skylines, la majorité utilise les meta heuristiques telles que l'algorithme NSGAII (NondominatedSortingGeneticAlgorithm II). NSGA-II [K. Deb and Meyarivan, 2002], et SPEAII (Strengh Pareto elitistalgorithm II) [E. Zitizler, 2005]

Dans [Claro et al, 2005] les auteurs considèrent 04 fonctions objectives :

- La Minimisation du coût .
- La Minimisation de la durée.
- La maximisation du chiffre d'affaire.
- La maximisation de la réputation.

Deux expérimentations ont été menées pour évaluer l'algorithme NSGAII Le nombre de service de la base varie entre 30 et 60, et le nombre de tâches ou classes varie entre 03 et 05. Pour la première expérimentation, la taille de la population des chromosomes varie de 10 à 200 individus et le nombre de générations est compris entre 10 et 500.
Les résultats montrent que la taille de l'ensemble des skylines varie entre 1 (pour 5 chromosomes) jusqu'à 70 (pour 200 chromosomes). Le temps d'exécution est acceptable puisqu'il est toujours inférieur à 20 secs, pour toutes les populations.
Dans le deuxième test effectué, la taille des populations a été fixée à 200 et le nombre de générations est égal à 500. Cette expérimentation, montre une corrélation positive entre le nombre de skylines et le nombre de classes/services.
Pour 05 classes et 50 services le nombre de skylines=161, et le temps d'exécution=25 sec.
Dans [T. Gonsalves, 2009] les auteurs proposent une optimisation multi objectives à base d'essaims particulaires, ils considèrent deux fonctions fitness (le coût et le temps d'attente) et négligent les contraintes globales, ils adoptent la version Gbest de l'algorithme.
Le choix de la meilleure position individuelle et globale est basé sur des règles simples (la dominance), l'expérimentation montre que le front de solutions obtenu, est proche du front de

Pareto-optimal réel, mais les auteurs n'ont pas indiqué le nombre de skylines oubliés par erreur, en plus ils n'ont pas donné des comparaisons avec d'autres algorithmes multiobjectives.

- Sélection Mono et Multi-objective

[Alrifai et al., 2010] proposent des approches combinant la sélection multi-objective (skylines) et mono-objective, En particulier ils proposent 03 algorithmes à base de skylines.
Le premier est nommé « exact skylines », il extrait les services skylines de chaque classe et applique une optimisation globale à base de MIP sur ces résultats.
Nous notons que l'extraction des skylines permet l'élagage de l'espace de recherche, et par la suite, elle assure un gain de temps considérable lors de l'optimisation globale.
Le deuxième algorithme est nommé representative- skylines, il extrait en premier lieu les services skylines de chaque classe, ensuite il réalise un clustering hiérarchique descendant de chaque catégorie de skyline.
L'algorithme de clustering applique un découpage binaire descendant des clusters à l'aide l'algorithme k-means, chaque cluster possède un service représentant qui a la plus grande affinité (i.e. la fonction fitness).

Conclusion

Dans ce chapitre on a présenté une formalisation du problème de sélection automatique de services web avec un exemple de motivation, ensuite on a montré une partie d'état de l'art des différentes approches de sélection de services Web composites proposées dans la littérature pour résoudre ce problème.
Le chapitre suivant est consacré a l'etude et le développement de notre application dans ce domaine c-à-d. la sélection de services Web à base d'Harmony search algorithme.

Chapitre 3

Conception et implémentation du prototype

Introduction :

Les Services Web individuels sont conceptuellement limités à des fonctionnalités relativement simples modélisés à travers une collection d'opérations simples. Cependant, pour certains types d'applications, il est nécessaire de combiner un ensemble de services Web individuels pour obtenir les plus complexes, appelé les services Web composites ou agrégées. Une question importante concernant les services web composites est celle de la sélection de la composition de service la plus appropriée parmi les différentes compositions possibles. Une solution possible est d'utiliser la qualité de service (QoS) pour évaluer, comparer et choisir la composition la plus appropriée. Plusieurs travaux ont été proposé pour résoudre ce problème [M.Alrifai, 2009], [Alrifai et al., 2010], [F.Li and S.Su., 2007]. Dans ce chapitre nous allons présenter la proposition adoptée pour résoudre le problème de sélection des compositions de services web à base de (QoS), en utilisant Harmony search Algorithme. Cette approche a été proposée pour résoudre les problèmes d'optimisation continus et combinatoires [Geem, 2008], [Lee KS, 2005]

Description de la base :

Description de la base :

En se basant sur la collection de texte présentée, par [Q. Yu, 2010] Nous avons créé un schéma de service qui contient N classes (par défaut 10) de services web. Chaque classe contient **L** d'instances de service web (par défaut 40). Chaque fournisseur se caractérise par

R critères qualités de service. Nous adoptant cinq critères pour évaluer les opérations des services : **l**atence, **f**iabilité, **d**isponibilité, **c**out et **r**éputation.

Les valeurs de ces paramètres sont générées en se basant sur une distribution uniforme. Les détails techniques de notre base sont illustrés dans la table suivant.

Critère	Intervalle
Cout	1-30
Temps d'exécution	1-300
Fiabilité	0,5-1
Disponibilité	0,7-1
Réputation	0-5

Table 3.1 – description de base de données. [Q. Yu, 2010]

Pour des raisons d'uniformisation Nous avons multiplié les valeurs des critères de couts et de latence par -1, ainsi la fonction objectif contient uniquement des critères a maximiser (fiabilité, disponibilité, réputation) et (cout, Temps d'exécution).

Description de la requête :

La requête de l'utilisateur comporte cinq éléments :

- La borne minimale pour la fiabilité.
- La borne minimale pour la disponibilité.
- La borne minimale pour la réputation.
- La borne maximale pour le cout.
- La borne maximale pour latence.

Conception :

Présentation de la recherche harmonique :

Recherche Harmonique (HS) est une évolution de la méta-heuristique algorithme d'optimisation, développé en 2001 par [Geem ZW and GV, 2001].

Il imite le processus musical dans la recherche d'un parfait état d'harmonie.

Harmony Search Algorithm est un concept simple et avec peu de paramètres. Il impose

quelques exigences mathématiques et peut facilement être mis en œuvre.
HS est un nouvel algorithme d'optimisation heuristique qui peut produiredе meilleures solutions que les autres algorithmes existants avec un nombre minimum d'itérations. Il ne nécessite pas la mise en valeurs des variables initiales de décision, donc il peut échapper à l'optimisation locale.

Fonctionnalités du Système :

Selon l'approche Harmonique, la mémoire d'harmonies (HM) est un emplacement de mémoire où tous les vecteurs de solutions courantes (ensembles de variables de décision) sont stockés. Dans chaque évolution de Recherche Harmonique, si un vecteur de solution est relativement d'une bonne forme physique est généré, il sera sauvegardé dans la mémoire de l'harmonie et pourrait être utilisée pour les générations à venir.
Les étapes d'Harmony search algorithm sont les suivantes [Geem ZW and GV, 2001] :

Étape 1 : Initialiser les paramètres du problème et de l'algorithme.
Étape 2 : Initialisation de la mémoire de l'harmonie.
Étape 3 : Improviser une nouvelle harmonie.
Étape 4 : Mettre à jour la mémoire de l'harmonie.
Étape 5 : Répétez les étapes 3 et 4 jusqu'à satisfaire critère de terminaison.

A **l'étape 1**, la fonction objective **F** de **N** variables de décision et l'ensemble des valeurs possibles pour chaque variable de décision sont définis.Les paramètres de l'algorithme *« Harmony Search »* sont également spécifiés dans cette étape.

Les paramètres de la Recherche Harmonique incluent : la taille de la mémoire de l'harmonie (HMS), le taux de considération de la mémoire d"harmonies (HMCR), taux d'ajustement des pitchs (PAR),et le critère de terminaison (nombre maximal de recherches).
Dans **l'étape 2**, la matrice des harmonies est remplie, (les harmonies sont des compositions de taille N)elles sont générées de manière aléatoire.
Une harmonie est un simple vecteur d'entier $x = (x1, x2, \ldots, xn)$, telle que chaque case **i** (ou variable de décision) représente un service à appartenant à la classe **i**.
Dans **l'étape 3**, un nouveau vecteur de l'harmonie comme $x^t = (x^t1, x^t2, \ldots, x^tn)$ est généré, ensemble basant sur trois règles :

(1) le choix d'une harmonie aléatoire à partir de la mémoire, (2) l'adaptation de l'harmonie sélectionnée précédemment (3) la sélection aléatoire.

Harmony Search Algorithm

```
Input: Pitchnum, Pitchbounds, Memorysize, Consolidationrate,
       PitchAdjustrate, IterationMax
Output: BestHarmony
Harmonies ← InitializeMemory(PitchNum, PitchBounds, MemorySize) ;
Evaluate(Harmonies);
for i ← IterationMax do
    Harmony ← Nil;
    foreach Pitchi ∈ PitchNum do
        if Rand() ≤ ConsolidationRate then
            RandomPitch = SelectRandomHarmonyPitch(Harmonies, Pitchi);
            if Rand() ∈ PitchAdjustrate then
                pitch = AdjustPitch(RandomPitch)
            else
                pitch = RandomPitch
            end
        else
            pitch = GetRandomPitch(Pitchbounds)
        end
    end
    update(Harmony, pitch, i)
    Evaluate(Harmony);
    if Cost(Harmony) ≤ Cost(Worst(Harmonies)) then
        Harmonies = Harmonies − Worst(Harmonies);
        Harmonies = Harmonies ⊎ Harmony
    end
end
BestHarmony = SelectBest(Harmonies)
returnBestHarmony
```

Algorithm 1: Harmony Search Algorithm

Lors de l'examen de la mémoire (règle 1), la valeur d'une variable de décision pour le nouveau vecteur est choisi parmi l'une des valeurs de variable de décision des harmonies appartenant à la mémoire.

Le HMCR, qui varie entre 0 et 1, est le taux de choisir une valeur à partir des valeurs stockées dans la mémoire.

En outre (1-HMCR) est le taux à sélectionner aléatoirement une valeur de la plage de valeurs possibles. Chaque composante obtenue par la considération de la mémoire est examinée pour

déterminer si elle doit être ajustée.

La probabilité de réglage du pitch (ou la variable de décision)de cette nouvelle harmonie est spécifiée par le paramètre PAR.la valeur courante de cette variable de décision est remplacée par une valeur d'une harmonie voisine (dans la mémoire).

Au cours de l'improvisation (génération d'une nouvelle harmonie), HMCR et PAR sont utilisées pour améliorer le vecteur solution globalement et localement respectivement.

Examen de la mémoire, Le réglage de la variable de décision ou la sélection aléatoire est appliqué à chaque variable du nouveau vecteur de l'harmonie à son tour.

Dans **l'étape 4**, la nouvelle harmonie construite précédemment remplacera la mauvaise harmonie de la mémoire, en termes de la fonction objective **F**. **l'étape 5** spécifie le critère d'arrêt qui est généralement le nombre d'itérations. La fonction objective est donnée comme suit :

$F(c) = U(c) + p(c)$ U est défini comme suit :

$$U(c) = \sum_{k=1}^{R} wk * ((Q^{t}k(c) - Q_{min'}(k))/(Q_{max'}(k) - Q_{min'}(k)) \quad (3.1)$$

La fonction est une agrégation de **R** critères de QOS normalisés et agrégés, cette agrégation utilise des pondérations (ou priorités) w_k , par défaut équitables (ie, si R=5 w_k = 0.2) C représente une composition concrète, et Q'k(c) est la valeur de son K^{eme} critère de QOS.

$$Q_{min'}(k) = \sum_{j=1}^{n} Q_{min}(j,k) \quad (3.2)$$

Ce terme représente la valeur minimale du critère de QOS **K** associé à une composition

$$Q_{max'}(k) = \sum_{j=1}^{n} Q_{max}(j,k) \quad (3.3)$$

Ce terme représente la valeur maximale du critère de QOS K associé à une composition.

$$Q_{min}(j,k) = min \forall sji \in SjQk(sji) \quad (3.4)$$

Ce terme représente la valeur minimale du critère de QOS k de la classe j (i.e un service simple).

$$Q_{max}(j,k) = max \forall sji \in SjQk(sji) \quad (3.5)$$

Ce terme représente la valeur maximale du critère 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 :

$$p(c) = -\sum_{k=1}^{R} (Dk)^2(c) \tag{3.6}$$

et

$$Dk(c) = 0 \tag{3.7}$$

si

$$Q^t k(c) \geq cons(k) et \mid Q^t k(c) - cons(k) \mid \tag{3.8}$$

Cons(k) : représente la K^{eme} contrainte globale de l'utilisateur. Nous notons aussi que la meilleure composition associée à la base décrite en section II, a une performance (fitness) =0.78, et de ce fait le taux d'optimalité d'une solution candidate c est f(c)/0.78.

Interface Homme/Machine (IHM) :

L'IHM représente l'élément clé dans l'utilisation de tout système informatique. Les interfaces de notre système de recherche sont conçues de manière à être simples, naturelles, compréhensible et faciles à manipuler.

Fenêtre principale :

La fenêtre principale se compose de trois onglets :

Le premier onglet : permet l'introduction des besoins de l'utilisateur (contrainte globales).

Le deuxième onglet : permet le chargement, génération, enregistrement et l'affichage de la base des services.

Le troisième onglet : permet l'invocation de l'optimisation par recherche harmonique.

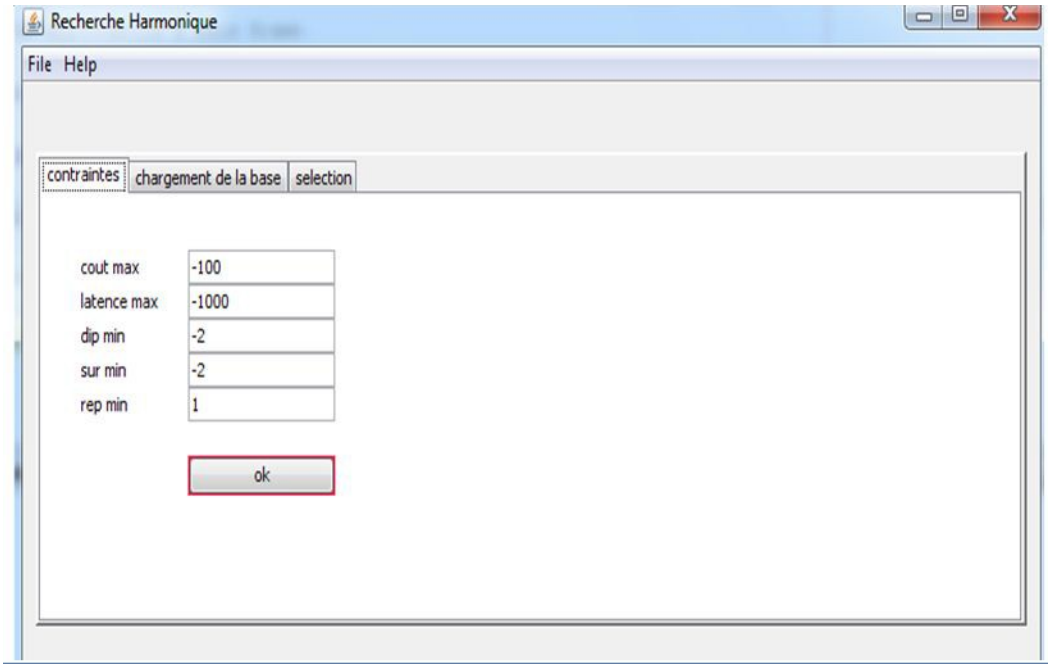

Figure 3.1 – Introduction des contraintes globales.

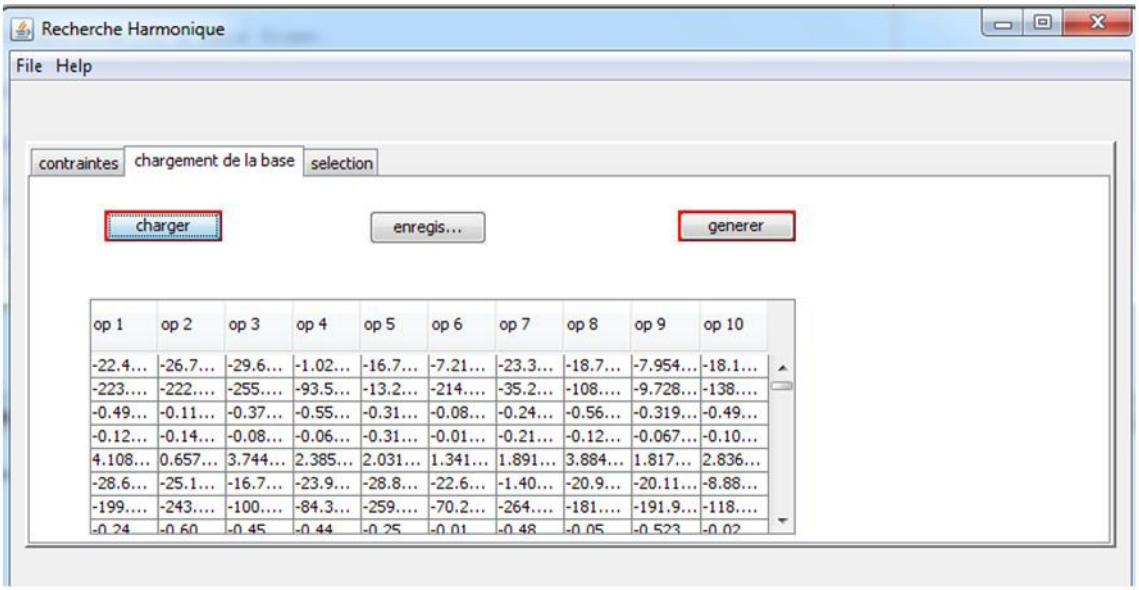

Figure 3.2 – Igénération et mise à jours de la base des services.

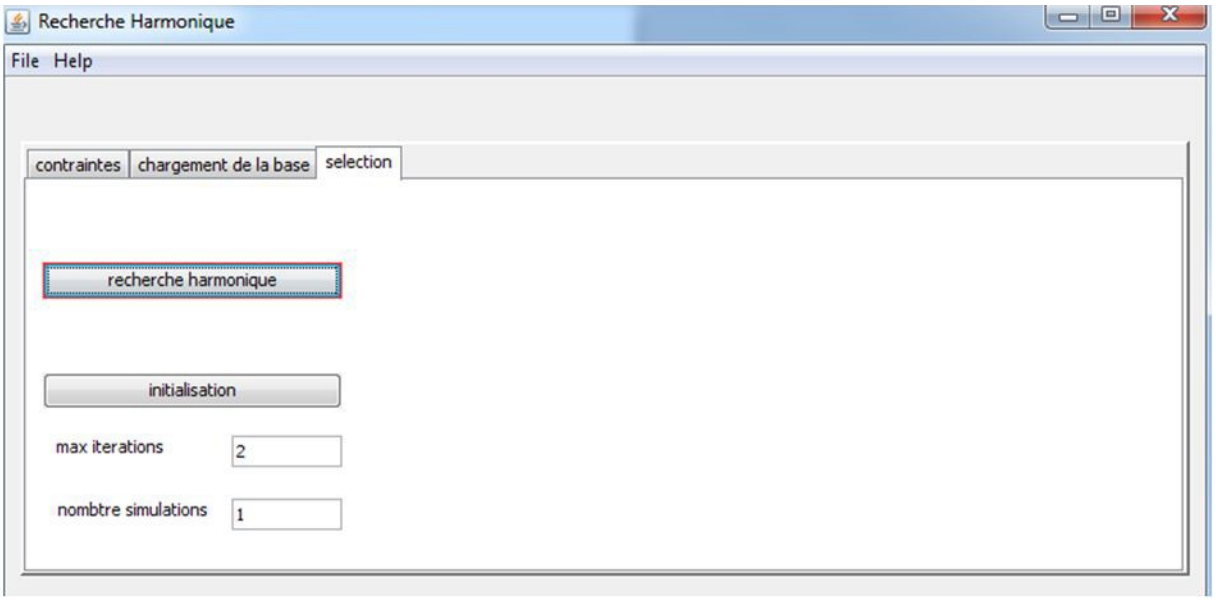

Figure 3.3 – Optimisation par recherche harmonique.

Expérimentation :

Nous avons mené une expérience pour évaluer la performance de l'approche proposée. Nous utilisons la base décrite en section II, comme moyen de test. Le but est de démontrer comment notre approche peut aider le client à sélectionner la meilleure offre. Nous avons développé notre prototype sous NetBeans IDE 6.8 de Sun Micro systems sous le système d'exploitation Windows, Processeur intelCorei3, 4 Go de RAM. Les sections suivantes montrent quelques simulations de l'algorithme proposé.

Résultats :

Les résultats des d'optimalité (fonction objective) ainsi que le temps d'exécution des différentes simulations sont présentés dans figures suivantes :

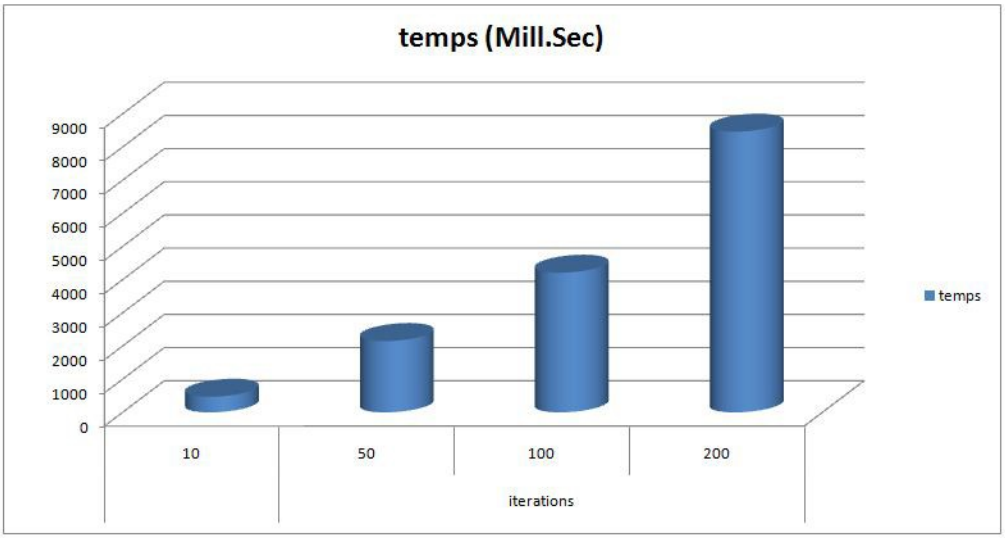

Figure 3.4 – Itérations vs temps (pour N =10).

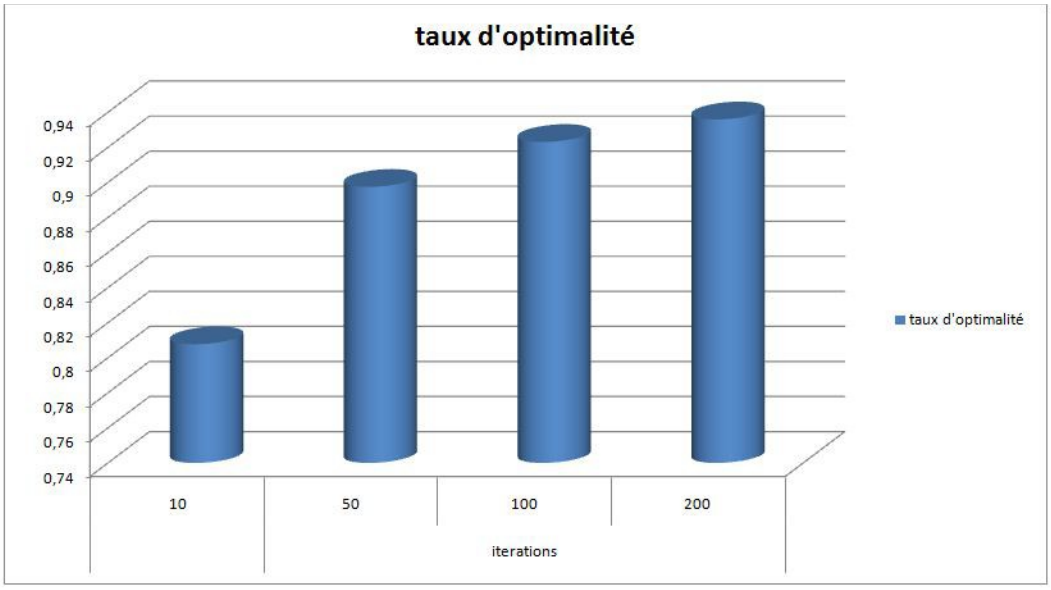

Figure 3.5 – Itérations vs optimalité (pour N =10).

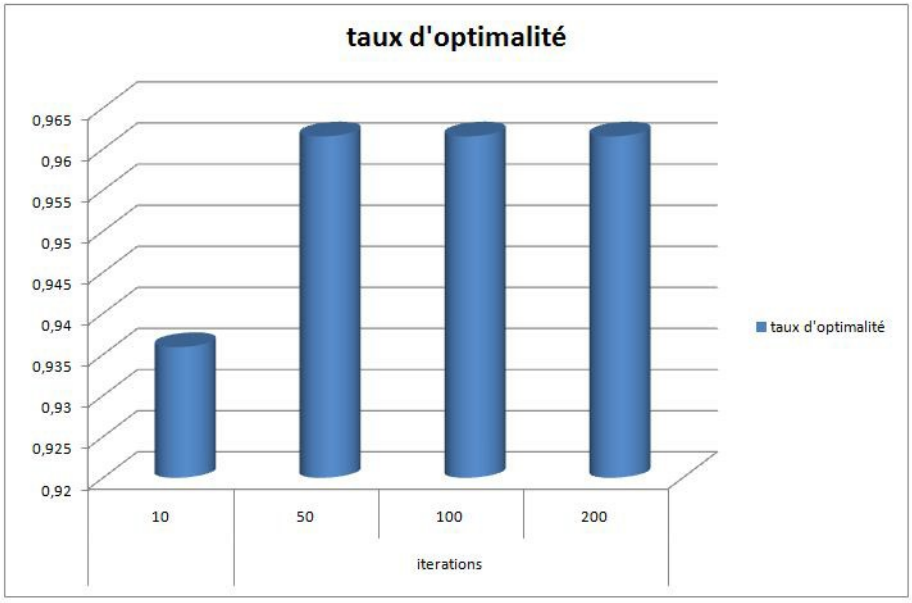

Figure 3.6 – Itérations vs optimalité (pour N =5).

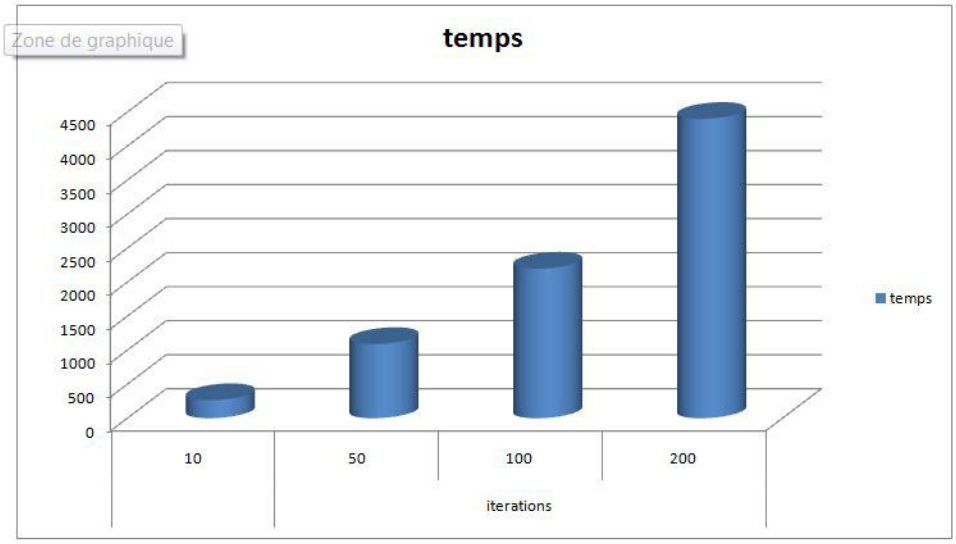

Figure 3.7 – Itérations vs optimalité (pour N =5).

Discussion :

D'après nos expériences on conclut les points suivants :

- Plus le nombre de classes N est grand, plus le taux d'optimalité est mauvais même pour un grand nombre d'itérations.
- Pour un nombre assez petit de classes les performances atteigne plus de %95 d'optimalité pour un temps raisonnable (moins de 09 SEC), mais si les le nombre de classe est grand (au de la de 100) les performances chutent de manière exponentielle.
- la configuration des taux de probabilités qui a donné les meilleures performances est la suivante : HMCR = 0,80

  PAR=0.3

3.7 Conclusion :

Dans ce chapitre on a proposé un algorithme à base d'optimisation basé sur la recherche harmonique pour résoudre le problème de sélection de service web. Nous avons présenté les résultats expérimentaux de notre approche, afin d'évaluer l'efficacité de cette dernière. Les résultats obtenus sont très encourageants et montrent la consistance de l'approche dans la pratique.

Conclusion Générale

Nous avons présenté dans ce mémoire les l'architecture orientée service ainsi que sa concréti-sation en forme de technologie de services web. La problématique étudiée consiste à instancier une composition abstraite, avec des services concrets tout en optimisant les R critères de QOS et en préservant un ensemble de contraintes globales.

Ce problème est reconnu comme étant NP Hard, et il est peu probable d'avoir une solution dans un temps raisonnable. Nous avons considéré (05) critères de QOS : Latence, coût, fiabilité, disponibilité, réputation. La composition pour résoudre ce problème, nous avons proposé un algorithme de sélection qui se base sur la recherche harmonique.

Ce dernier se base sur mémoire de solutions acceptables et un jeu d'opérateurs de déplacement inspirés de la recherche locale et de l'aléatoire.

Comme perspectives à ce travail, nous proposons d'implémenter d'autres algorithmes de sélection, pour cela, nous pouvons citer : - La programmation par contrainte, (Ou des versions du branch and Bound - La programmation dynamique, les algorithmes d'optimisation multi objectifs (NSGA II, SPEA II). . . .

Bibliographie

[Akbar et al., 2001] Akbar, M., Manning, E., Shoja, G., and Khan, S. (2001). Heuristic Solutions for the Multiple-Choice Multi-dimension Knapsack Problem. *Lecture Notes in Computer Science*, pages 659–668.

[Alrifai et al.,] Alrifai, M., Risse, T., Dolog, P., and Nejdl, W. A scalable approach for qos-based web service selection. In *Service-Oriented Computing–ICSOC 2008 Workshops*.

[Alrifai et al., 2012] Alrifai, M., Risse, T., and Nejdl, W. (2012). A hybrid approach for efficient web service composition with end-to-end qos constraints. *ACM Transactions on the Web (TWEB)*, 6(2) :7.

[Alrifai et al., 2010] Alrifai, M., Skoutas, D., and Risse, T. (2010). Selecting skyline services for qos-based web service composition. In *Proceedings of the 19th international conference on World wide web*, pages 11–20. ACM.

[Ardagna and Pernici, 2005] Ardagna, D. and Pernici, B. (2005). Global and local qos constraints guarantee in web service selection. In *Web Services, 2005. ICWS 2005. Proceedings. 2005 IEEE International Conference on*, pages –806.

[Ardagna and Pernici, 2007] Ardagna, D. and Pernici, B. (2007). Adaptive service composition in flexible processes. *Software Engineering, IEEE Transactions on*, 33(6) :369–384.

[B. Benatallah and Dumas, 2002] B. Benatallah, Q. Z. Sheng, A. H. H. N. and Dumas, M. (2002). Composition and peer-to-peer provisioning of dynamic web services. In *In 18th International Conference on Data Engineering,IEEE Computer Society*, page 297–308.

[Canfora et al., 2005] Canfora, G., Di Penta, M., Esposito, R., and Villani, M. (2005). Qos-aware replanning of composite web services. In *Web Services, 2005. ICWS 2005. Proceedings. 2005 IEEE International Conference on*, pages 121–129 vol.1.

[Dréo et al., 2006] Dréo, J., Petrowski, A., Siarry, P., and Taillard, E. (2006). *Metaheuristics for hard optimization : methods and case studies*. Springer Science & Business Media.

[E. Zitizler, 2005] E. Zitizler, L. T. (2005). Multiobjective optimisation using evolutionary algorithms a comparative case study. In *In Parallel Problem Solving from Nature V,Germany*, page 1998.

[F.Hadjila, 2013] F.Hadjila, M. C. (2013). Qos-aware service selection based on mimetic algorithm. In *In Proceedings of CNEDI '12 Skikda*.

[F.Li and S.Su., 2007] F.Li, F.Yang, K. and S.Su. (2007). Q-peer : A decentralized qos registry architecture for web services. In *In Proceedings of the International Conference on Services Computing*, page 145–156.

[G. Canfora, 2005] G. Canfora, M. Di Renta, R. E. M. L. V. (2005). An approach for qos aware service composition based on genetic algorithms. In *GECCO'05',Washington, USA,*.

[GARDIEN, 2002] GARDIEN, G. (2002). Xml des bases de données aux services web. *édition Dunod*.

[Geem, 2008] Geem, Z. W. (2008). Novel derivative of harmony search algorithm for discrete design variables. *Applied mathematics and computation*, 199(1) :223–230.

[Geem ZW and GV, 2001] Geem ZW, K. J. and GV, L. (2001). Original harmony search : A new heuristic optimization algorithm : Harmony search. *Simulation*.

[H.Kreger, 2001] H.Kreger (2001). Web service conceptual architecture.

[J. Wang, 2005] J. Wang, Y. H. (2005). Optimal web service selection based on multi-objective genetic algorithm. In *Proceedings of the ISCID 2008,Wuhan,2008*, pages 553–556.

[Jaeger and Mühl, 2006] Jaeger, M. C. and Mühl, G. (2006). Soft real-time aspects for service-oriented architectures. In *E-Commerce Technology, 2006. The 8th IEEE International Conference on and Enterprise Computing, E-Commerce, and E-Services, The 3rd IEEE International Conference on*, pages 5–5. IEEE.

[J.Garcia,] J.Garcia, E. Les services web.

[K. Benouaret, 2011] K. Benouaret, D. Benslimane, A. H. M. B. (2011). Fudocs : A web service composition system based on fuzzy dominance for preference query answering. *Proceedings of the VLDB Endowment*, Vol. 4, No. 12.

[K. Benouaret and Hadjali, 2011] K. Benouaret, D. B. and Hadjali, A. (2011). Top-k service compositions : A fuzzy set-based approach. *In ACM SAC,TaiChung,Taiwan*.

[K. Deb and Meyarivan, 2002] K. Deb, S. Agrawal, A. P. and Meyarivan, T. (2002). A fast and elitist multi-objective genetic algorithm : Nsga-ii. *Journal IEEE Transaction on Evolutionary Computation*.

[Kritikos and Plexousakis, 2009] Kritikos, K. and Plexousakis, D. (2009). Mixed-integer programming for qos-based web service matchmaking. *Services Computing, IEEE Transactions on*, 2(2):122–139.

[L. Xu,] L. Xu, L. Shi, R. B. J. A multiple criteria service composition selection algorithm supporting time-sensitive rules. In *Proceedings of the 12th IFIP,IEEE International Symposium on Integrated Network management Dublin Irland 2011*.

[Lee KS, 2005] Lee KS, G. Z. (2005). Continuous harmony search :, a new meta-heuristic algorithm for continuous engineering optimization : Harmony search theory and practice. In *Computer Methods in Applied Mechanics and Engineering*.

[Liu et al., 2011] Liu, X., Bouguettaya, A., Yu, Q., and Malik, Z. (2011). Efficient change management in long-term composed services. *Service Oriented Computing and Applications*, 5(2) :87–103.

[L.Zeng, 2004] L.Zeng, B.Benatallah, A.-H.-H. M. J. Q.-Z. (2004). Qos-aware middleware for web services composition. In *IEEE Trans. Softw. Eng*, page 11–327.

[M.Alrifai, 2009] M.Alrifai, T. (2009). Combining global optimization with local selection for efficient qos-aware service composition. In *In Proceedings of the 18th International Conference on World Wide Web (WWW'09),ACM, New York*, page 881–890.

[M.Gudgin and Henrik.F, 2002] M.Gudgin, M.Hadley, N.-J. and Henrik.F (2002). Soap version 1.2 part 2. In *Adjuncts*.

[Mostofa Akbar et al., 2006] Mostofa Akbar, M., Sohel Rahman, M., Kaykobad, M., Manning, E., and Shoja, G. (2006). Solving the Multidimensional Multiple-choice Knapsack

Problem by constructing convex hulls. *Computers and Operations Research*, 33(5) :1259–1273.

[Nilo.Mitra, 2002] Nilo.Mitra (2002). *SOAP Version 1.2 Part 0*.

[Papazoglou, 2003] Papazoglou, M. (2003). Service-oriented computing : concepts, characteristics and directions. In *Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on*, pages 3–12.

[Pisinger, 1995] Pisinger, D. (1995). Algorithms for knapsack problems.

[Q. Yu, 2010] Q. Yu, A. B. (2010). Foundations for efficient web service selection. In *Springer Science+Business Media*.

[Rampacek, 2006] Rampacek, S. (2006). *Sémantique, interactions et langages de description des services web complexes*. PhD thesis, Université de Bourgogne.

[R.Chinnici2003,] R.Chinnici2003, M. Gudgin, J.-J. M. J. S.-S. W. Web services description language (wsdl) version 2.0 part 1 : Core language. In *W3C Working Draft, See http ://www.w3.org/TR/2003/WD-wsdl20-20031110/*.

[Ricardo.R, 2004] Ricardo.R (2004). *Découverte et Sélection de Service Web pour une application Mélusine*. PhD thesis, l'Institut d'Informatique et de Mathématiques Appliquées de Grenoble.

[T. Gonsalves, 2009] T. Gonsalves, K. Y. K. (2009). Itoh service cost and waiting time – a multi-objective optimization scenario. In *In Proceeding of IEEE*.

[Zeng et al., 2003] Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J., and Sheng, Q. Z. (2003). Quality driven web services composition. In *Proceedings of the 12th international conference on World Wide Web*, pages 411–421. ACM.

[Zhang and Ren, 2011] Zhang, Y. and Ren, M. (2011). Web service selection based on utility of weighted qos attributes. In *Emerging Research in Web Information Systems and Mining*, pages 417–425. Springer.