Université Abou Bekr Belkaid
Tlemcen Algérie
جامعة أبي بكر بلقايد
تلمسان الجزائر

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 : *Modèle Intelligent et Décision (MID)*

Thème

SELECTION DES SERVICES WEB : Une approche à base de Skylines et Clustering Hiérarchique Ascendant

Réalisé par :

- **GHAFFOUR Ayoub**
- **TAHIR Fouzi**

Présenté le 08 Juin 2013 devant le jury composé de MM.

- *Mr BENAMAR Abdelkrim* *(Président)*
- *Mr HADJILA Fethallah* *(Encadreur)*
- *Mr BELABED Amine* *(Examinateur)*
- *Mr MERZOUG Mohammed* *(Examinateur)*

Année Universitaire : 2012-2013

Université Abou Bekr Belkaid
Tlemcen Algérie
جامعة أبي بكر بلقايد
تلمسان الجزائر

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 : *Modèle Intelligent et Décision (MID)*

Thème

SELECTION DES SERVICES WEB : Une approche à base de Skylines et Clustering Hiérarchique Ascendant

Réalisé par :

- **GHAFFOUR Ayoub**
- **TAHIR Fouzi**

Présenté le 08 Juin 2013 devant le jury composé de MM.

- *Mr BENAMAR Abdelkrim* *(Président)*
- *Mr HADJILA Fethallah* *(Encadreur)*
- *Mr BELABED Amine* *(Examinateur)*
- *Mr MERZOUG Mohammed* *(Examinateur)*

Année Universitaire : 2012-2013

Remerciements

Avant tout, nous remercions DIEU le tout puissant qui nous a donné santé, courage pour réaliser ce travail.

*Tout d'abord, nous remercions sincèrement et très chaleureusement notre encadreur M^r^* ***. Hadjila Fethallah*** *que nous ne remercions jamais assez, pour avoir dirigé ce travail, pour de nombreuses discussions scientifiques, pour sa simplicité d'accès, pour l'énorme effort dans la correction de ce mémoire, pour son soutien permanant et sans relâche, pour sa façon à la fois sympathique et compétente et pour ses conseils et ses encouragements durant toute la période de son enseignement et son encadrement.*

Nos gratitudes s'adressent également à l'ensemble des enseignants du département de l'Informatique, université de Tlemcen qui ont contribué à notre formation dans cette filière.

Enfin, nos gratitudes s'adressent également à tous ceux qui nous ont aidés de près ou de loin à achever ce modeste travail.

Dédicaces

A mes très chers parents qui n'ont jamais cessé de m'aider et m'encourager durant tout mon parcours et qui m'ont permis d'être aujourd'hui ce que je suis,

A la mémoire de mes grands-parents qui ont voulu que je réussisse,

A mes chers frères Taha Yacine et Abdelghafour,

A mes tantes, oncles, cousins, cousines et tous les membres de ma grande et aimable famille qui m'ont toujours soutenu et poussé à donner le meilleur de moi-même,

A mon binôme ***Fouzi*** *avec qui j'ai appris tellement de choses et que je remercie d'avoir été patient et collaboratif,*

A tous mes amis, qu'ils m'excusent de ne pas pouvoir les citer au risque d'oublier quelqu'un, je vous dis MERCI pour tous les moments inoubliables qu'on a passé ensemble durant ces cinq années,

A tous ceux qui me sont chers,

Ayoub.

Dédicaces

A mes chers parents qui ont toujours été présents pour moi et qui m'ont toujours soutenu dans ma vie,

A mes deux sœurs Souad et Meriem, et trois frères Sidi Mohammed, Abdelkader et Fethi. Je leurs souhaite beaucoup de réussite dans la vie,

A mon ami et binôme ***Ayoub*** *sans qui tout ce projet n'aurai jamais existé, avec qui j'ai passé des moments inoubliables et que je remercie d'avoir été patient et collaboratif en étant aussi un grand professeur,*

A tous mes très nombreux amis, et anciens collègues que je ne pourrai pas tous vous citer sinon ça prendrait une page entière de noms, merci de m'avoir soutenu et aidé quand j'en avais besoin.

A tous ceux qui sont chers à moi,

Fouzi.

Résumé :

La sélection des services web composés est l'une des problématiques majeures de l'architecture SOA. En effet, les utilisateurs ont toujours besoin de sélectionner les meilleurs services qui répondent leurs besoins, tout en vérifiant des exigences globales.

Pour résoudre cette problématique, nous proposons une approche hybride (multi objective et mono objective). Notre proposition commence par l'extraction des éléments skylines des classes abstraites. Ensuite, nous effectuons un CHA (Clustering Hiérarchique Ascendant) sur toutes les classes. Enfin, nous parcourons les N hiérarchies (niveau par niveau) jusqu'à l'obtention de l'optimum global.

Les expérimentations menées confirment l'obtention de l'optimum global dans un temps réduit. En plus, elles montrent la supériorité de notre approche par rapport aux autres méthodes mono objectives.

Mots clés : La sélection de service web, SOA, skyline, CHA.

Abstract:

The selection of web services is one of the major problems of SOA architecture. The users still need to select the best services that satisfy their needs by checking the overall requirements.

To solve this problem, we propose a hybrid approach (multi-objective and mono-objective).

Our proposal begins with the extraction of the skylines of each abstract class. Then we make a Hierarchical Clustering on each class. Finally, we explore the N hierarchies (level by level) until the reach of the global optimum. The experiments carried out confirm the obtaining of the global optimum in a short time. In addition, the results show the superiority of our approach compared to other mono-objective methods.

Keywords: The selection of web services, SOA, skyline, Hierarchical Clustering Ascending.

ملخص

اختيار خدمات الواب هي واحدة من المشاكل الرئيسية لهندسة SOA. المستخدمون بحاجة دائمة لأفضل الخدمات التي تلبي احتياجاتهم، و ذلك بالتحقق من المتطلبات العامة.

لحل هذه المشكلة نقترح اتباع نهج هجين (متعدد الأهداف وأحادي الأهداف). يبدأ اقتراحنا باستخراج عناصر الأفق للفئات المجردة، ثم نقوم بتقسيم هرمي تصاعدي على جميع الفئات. وأخيرا، نجتاز الهرم مستوى بعد مستوى حتى الوصول إلى أفضل الحلول.

التجارب تؤكد الحصول على أفضل الحلول في وقت قصير. وبالإضافة إلى ذلك، فإنها تظهر تفوق نهجنا بالمقارنة مع الطرق الأخرى ذات الهدف الواحد.

الكلمات الرئيسية : خدمات الواب، الأفق، التقسيم الهرمي التصاعدي.

TABLE DE MATIERES

LISTE DE FIGURES

LISTE DE TABLEAUX

Introduction générale

Contexte :

L'interaction de différentes applications offertes par des entreprises et organisations en réseau a toujours été une affaire complexe, tant que ces applications peuvent être écrites dans des langages de programmation différents et exécutées sur des plates-formes hétérogènes, la standardisation de certains aspects de cette interaction est nécessaire. Plusieurs technologies permettent de résoudre ce problème de communication à travers un réseau. Parmi eux les « Services Web ».

Les services web, comme une technologie clé pour la réalisation des architectures orientées services, promettent de permettre l'interopérabilité et l'intégration entre systèmes et applications hétérogènes. La découverte et la sélection des services appropriés pour répondre à une demande donnée, constitue une tâche fondamentale dans de telle architecture.

L'une des questions de recherche des plus importantes dans ce domaine est relative à la prise en compte des différents critères de la qualité de service dans la découverte et la sélection du meilleur service web ayant une fonctionnalité donnée, afin de satisfaire au mieux le demandeur du service.

Problématique :

Ce problème a attiré l'attention des chercheurs au cours de la dernière décennie. La raison pour cela est que cette nouvelle technologie évolue et que plus de services deviennent disponibles, il devient important d'être en mesure de localiser le service qui répond à nos besoins. Beaucoup de propositions ont été mises en avant pour résoudre ce problème et plusieurs normes ont été définies, mais aucun d'entre eux n'a été efficace ou est maintenant accepté comme la façon d'effectuer une recherche de service.

Par conséquent, les utilisateurs ont besoin d'approches qui permettent la sélection et la personnalisation de leurs compositions en utilisant les critères de QoS (Quality of Services) offerts par les fournisseurs de services.

Cette sélection doit optimiser au mieux les différents attributs de QoS, en plus elle doit prendre en compte les contraintes globales exigées par les utilisateurs.

Contribution :

Notre projet a pour le but d'assurer la sélection des services web à base de QoS en prenant une composition abstraite qui comporte de N tâches et un ensemble de contraintes globales, le problème revient à instancié ces N taches avec des services concrets de tel sorte que les valeurs de QoS sont optimisées, et en même temps les contraintes globales de l'utilisateur sont satisfaites.

Pour cela nous proposons une approche constituée de trois étapes :

- La première extrait les éléments skylines de chaque tache.
- La deuxième réalise un clustering hiérarchique de chaque classe.
- La troisième effectue un parcours systématique de toutes les hiérarchies (et niveau par niveau) jusqu'à l'obtention de l'optimum.

Plan du mémoire :

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

Chapitre I :

Nous avons présenté les concepts généraux des services web. Tout d'abord, nous commençons par introduire l'origine des concepts, quelques définitions ainsi l'architecture des services web. Ensuite, nous détaillons les différents standards associés aux services web. De plus, nous discutons sur quelques thèmes de recherche actuels dans ce domaine. Enfin, nous citons quelques domaines d'applications, avantages et limites des services web afin de conclure ce que nous avons fait.

Chapitre II :

Ce chapitre est consacré à la conception, l'implémentation et l'expérimentation du prototype. Tout d'abord, nous présentons nos deux algorithmes : l'algorithme BBS (Branch and Bound Skyline) et l'algorithme agglomératif. 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.

Conclusion générale :

Cette partie résumé les résultats de notre travail, et présente les perspectives que nous souhaitons réaliser dans le futur.

Chapitre 1 : Les services Web

I Introduction :

L'accès aux systèmes d'information s'appuie aujourd'hui de plus en plus sur des technologies internet. Les efforts de standardisation dans ce contexte ont accentué l'engouement des personnes et des organisations (aussi bien académiques, qu'industrielles, commerciales, ou institutionnelles) pour l'utilisation de l'Internet et ont permis l'´émergence des services Web comme support de développement des applications accessibles par Internet.

Ainsi, les technologies associées aux services Web sont devenues incontournables pour le développement d'applications interagissant les unes avec les autres par le biais de l'Internet.

Nous proposons dans ce chapitre de faire un point sur le sujet des services Web.

II Historique et Origine des Concepts :

Les services Web sont nés de l'effort de plusieurs organisations qui ont partagé un intérêt commun en développant et en maintenant "un marché électronique". Celles-ci souhaitaient pouvoir communiquer plus simplement et sans avoir à se concerter sur chacune de leur transaction pour pouvoir interpréter leurs différentes données. Elles souhaitaient supprimer l'isolement de leur système informatique avec les autres.

C'est ainsi que naquit en 1975 l'EDI (Échange de Données Informatisées) [1]. Vers la fin des années 80, l'évidence fut que l'âge des systèmes informatiques isolés touchait à son terme tandis que différents ordinateurs, de tailles, de capacités et de formes variées, apparaissaient au sein d'une même organisation. Les départements informatiques veulent bien évidemment exploiter au mieux et au plus bénéfique la précieuse puissance d'analyse qu'ils avaient à disposition. Il fallut donc rendre les applications informatiques capables de déplacer leurs travaux, c'est-à-dire de procéder à un véritable traitement distributif.

Les années 90 furent témoin non seulement de la recrudescence des ordinateurs personnels, mais aussi du décollage phénoménale d'Internet avec une demande grandissante de standards susceptibles de travailler sur n'importe quelle plateforme.

A la fin des années 90, l'e-speak d'Hewlett Packard fit son apparition en même temps que XML (eXtensible Markup Language) [2]. HP (Hewlett Packard) voulait ainsi concurrencer l'e-business d'IBM (International Business Machines) qui par leur

campagne de publicité retentissante donnait la forte sensation d'être les inventeurs du système d'e-services utilisant les protocoles établis comme l'HTTP (Hyper-Text Transfer Protocol) [3] et XML pour permettre de passer outre les différences entre les divers systèmes coexistant dans un même réseau. Sans le savoir, les services Web étaient nés.

Aujourd'hui, les services Web provoquent un intérêt certain auprès des architectes et des décideurs.

Dès à présent, les Web Services sont sortis du champ des échanges interentreprises pour s'adapter à celui du référencement et de la mise à disposition des ressources de l'entreprise, empiétant en ce sens sur les technologies de type EAI (Intégration d'Applications d'Entreprise) [4]. Cette utilisation à elle seule prouve la qualité du modèle et sa pérennité, notamment au niveau des couches les plus basses. Par contre, la normalisation complète d'une architecture distribuée construite sur les Web Services n'est pour l'instant pas encore tout à fait établie. Par ailleurs, ce modèle n'échappe pas à des problèmes de performance. [5]

III Définitions :

III-1 Définition 1 :

Le consortium W3C (World Wide Web Consortium) définit un service Web comme étant: «*A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interacact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.*» [6]:

- Il est identifié par un URI (Uniform Resource Identifier) [7].
- Ses interfaces et ses liens peuvent être décrits en XML.
- Sa définition peut être découverte par d'autres services Web.
- Il peut interagir directement avec d'autres services Web à travers le langage XML en utilisant des protocoles Internet standards.

III-2 Définition 2 :

Les services web représentent un domaine de recherche jeune. IBM donne dans un tutorial la définition suivante des services web: « *Web services are a set of emerging standards that enable interoperable integration between heterogeneous IT processes and systems. You can think of them as a new breed of web application that is self-contained and self-describing, and that can provide functionality and interoperation ranging from the basic to the most complicated business and scientific processes. In short, web services hold the promise for providing a common standard mechanism for interoperable integration among disparate systems, and the key to their utility is their standardization. This common mechanism for delivering a "service" makes them ideal for implementing a service-oriented architecture (SOA).* » [8]

III-3 Définition 3 :

Selon notre point de vue, un service web peut être définit comme un composant logiciel représentant une fonction applicative (ou un service applicatif). Il peut être accessible depuis une autre application (un client, un serveur ou un autre service Web) à travers le réseau Internet en utilisant les protocoles de transport disponibles.

Les services Web font alors évoluer Internet vers une véritable plate-forme informatique distribuée et permettent à des systèmes hétérogènes de coopérer aisément et en toute sécurité.

IV Architecture des services web:

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

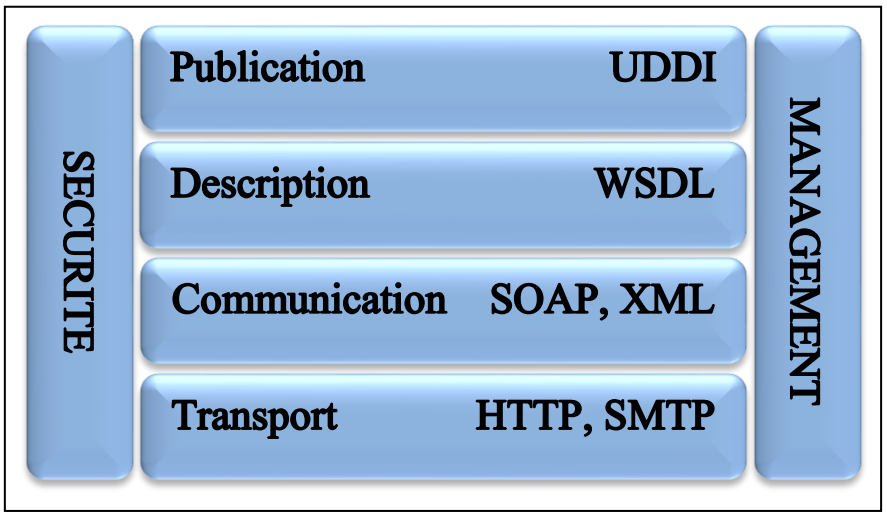

Figure I.1 : Architecture en Pile des services Web. [13]

Couche transport

Cette couche est responsable du transport des messages XML échangés entre les applications. Actuellement, cette couche inclut HTTP, SMTP (Simple Mail Transfer Protocol) [9], FTP (File Transfer Protocol) [10]. Le transport de messages XML-RPC (Remote Procedure Call) [11] ou SOAP (Simple Object Access Protocol) [12] est assuré par le standard HTTP.

Couche communication

Cette couche est responsable du formatage des données échangées de sorte que les messages peuvent être compris à chaque extrémité. Cette couche utilise des protocoles reposants sur le langage XML, car sa syntaxe unique résout les conflits syntaxiques lors de l'encodage des données. Actuellement, SOAP est le protocole le plus utilisé pour cette couche.

Couche description de service

Cette couche est responsable de la description de l'interface publique du service Web. Le langage utilisé pour décrire un service Web est WSDL (Web Services Description Language) [12] qui est la notation standard basée sur XML pour construire la description de l'interface d'un service.

Couche de découverte de service

Cette couche est chargée de centraliser les services dans un registre commun, et de simplifier les fonctionnalités de recherche et de publication des services Web. Actuellement, la découverte des services est assurée par un annuaire UDDI (Universal Description, Discrovery, and Integration) [12].

V Cycle de vie des services Web :

Il s'agit maintenant d'identifier chaque acteur de ses Web services et de comprendre comment ils interagissent les uns avec les autres. Les trois éléments les plus importants des services Web sont les fournisseurs de service, les annuaires de services et les consommateurs de service. Le fournisseur (ou serveur) crée le service Web et publie toutes ces caractéristiques dans l'annuaire de service.

L'annuaire rend disponible les interfaces d'accès aux services et donnant le contrat et l'architecture employée pour permettre les interactions. Le consommateur (ou client) quant à lui, accède à l'annuaire pour rechercher les services Web dont il a besoin. Il peut ainsi envoyer ses requêtes au service désiré et obtenir les réponses qu'il pourra analyser. Cette architecture fonctionne de la manière suivante :

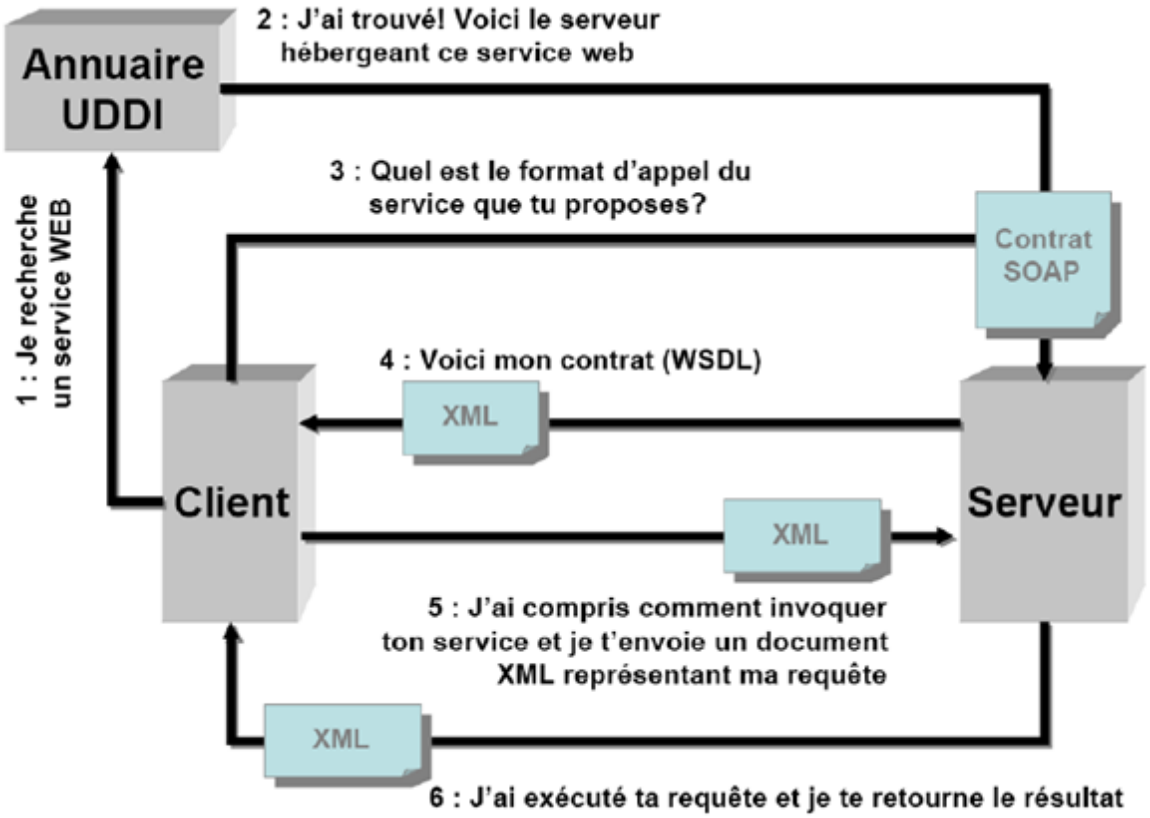

Figure I.2 : Cycle de vie des services Web. [5]

1. Le client envoie une requête à l'annuaire de Service pour trouver le service Web dont il a besoin.

2. L'annuaire cherche pour le client, trouve le service Web approprié et renvoie une réponse au client en lui indiquant quel serveur détient ce qu'il recherche.
3. Le client envoie une deuxième requête au serveur pour obtenir le contrat de normalisation de ses données.
4. Le serveur envoie sa réponse sous la forme établie par WSDL en langage XML.
5. Le client peut maintenant rédiger sa requête pour traiter les données dont il a besoin.
6. Le serveur fait les calculs nécessaires suite à la requête du client, et renvoie sa réponse sous la même forme normalisée. [5]

VI Les standards associés aux Services Web :

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

VI-1 XML – eXtensible Markup Language

XML « eXtensible Markup Language » est un format texte simple, très flexible tiré du SGML (Standard Generalized Markup Language) [14]. À l'origine conçu pour la publication électronique à grande échelle, XML joue aussi un rôle de plus en plus important dans l'échange d'une large variété de données sur le Web et ailleurs.

W3C recommande depuis 1998 XML en tant que standard de description de données. XML est un méta langage permettant d'identifier la structure d'un document. Un document est composé d'une définition de sa structure et d'un contenu. La structure d'un document XML est souvent représentée graphiquement comme un arbre. La racine du document constitue le sujet du document, et les feuilles sont les éléments de ce sujet. De ce fait, XML est alors flexible et extensible, et est devenu rapidement le standard d'échange de données sur le web.

Il a été conçu pour des documents arbitrairement complexes, tout en s'appuyant sur cinq grands principes simples et clairs :

- Lisibilité à la fois par les machines et par les utilisateurs.
- Définition sans ambiguïté du contenu d'un document.
- Définition sans ambiguïté de la structure d'un document.

- Séparation entre documents et relation entre documents.
- Séparation entre structure des documents et présentation des documents

XML possède une galaxie de technologies. Parmi celles-ci nous pouvons citer XQuery – XML Query (langage de requêtes) [15], XSL – eXtensible Stylesheet Language (définition de présentations de documents XML) [16], XSLT – XSL Transformations (langage permettant la transformation de documents XML) [17], XPath – XML Path (langage de requêtes permettant d'accéder à chaque élément d'un document XML) [18]. [19]

VI-2 SOAP : Simple Object Access Protocol

SOAP est un protocole d'invocation de méthodes sur des services distants. Basé sur XML, SOAP a pour principal objectif d'assurer la communication entre machines. Le protocole permet d'appeler une méthode RPC et d'envoyer des messages aux machines distantes via HTTP. Ce protocole est très bien adapté à l'utilisation des services Web, car il permet de fournir au client une grande quantité d'informations récupérées sur un réseau de serveurs tiers.

SOAP est bien plus populaire et utilisé que XML-RPC. C'est une recommandation du W3C. D'après cette recommandation, SOAP est destiné à être un protocole léger dont le but est d'échanger des informations structurées dans un environnement décentralisé et distribué. Une des volontés du W3C vis-à-vis de SOAP est de ne pas réinventer une nouvelle technologie. SOAP a été construit pour pouvoir être aisément porté sur toutes les plates-formes et les technologies existantes.

a- Structure d'un message SOAP

La grammaire de SOAP est assez simple à comprendre. Elle procure un moyen d'accès aux objets par appel de méthodes à distance. Les deux plus fortes fonctionnalités de SOAP sont sa simplicité et le fait que tout le monde a accepté de l'utiliser. Un message SOAP est composé de deux parties obligatoires : l'enveloppe SOAP et le corps SOAP ; et une partie optionnelle : l'entête SOAP.

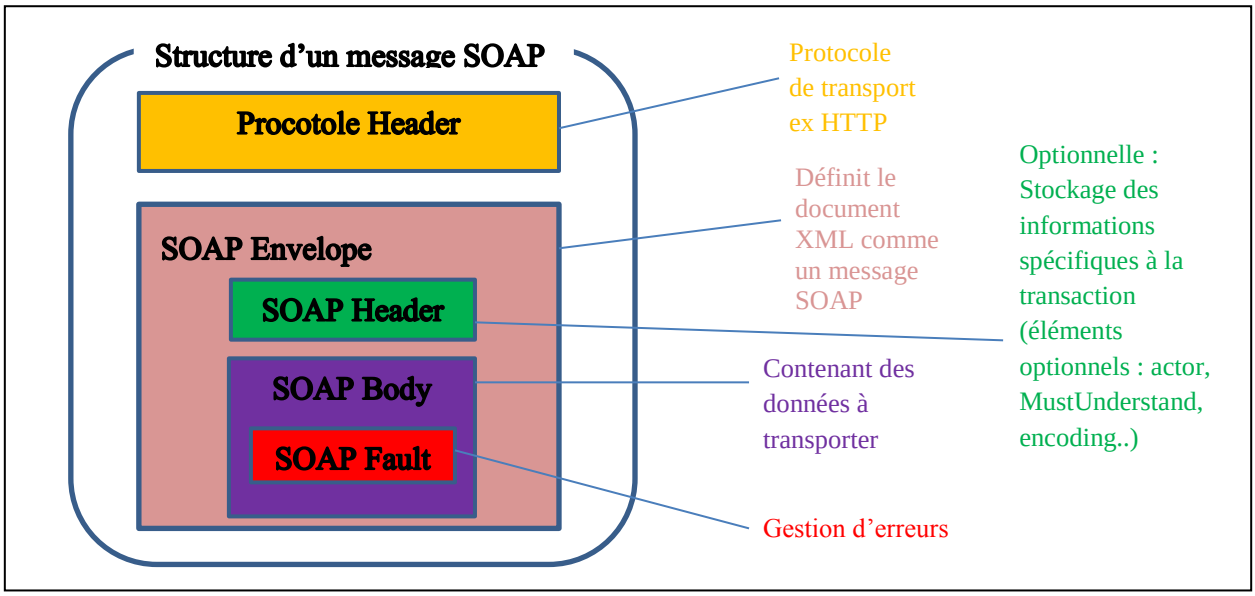

Figure I.3 : Le contenu d'une enveloppe SOAP [12]

- **Protocole header** (entête du protocole de transport)**:** qui dépend de protocole de transport utilisé, par exemple si le protocole HTTP est utilisé, l'en-tête contient :
  - La version de protocole HTTP utilisée.
  - La date de génération de message SOAP.
  - Le type d'encodage du contenu (généralement de type XML).
- **SOAP envelope** (enveloppe) est l'élément de base du message SOAP. L'enveloppe contient la spécification des espaces de désignation (namespace) et du codage de données.
- **SOAP header** (entête) est une partie facultative qui permet d'ajouter des fonctionnalités à un message SOAP de manière décentralisée sans agrément entre les parties qui communiquent. C'est ici qu'il est indiqué si le message est mandataire ou optionnel. L'entête est utile surtout, quand le message doit être traité par plusieurs intermédiaires.
- **SOAP body** (corps) est un *container* pour les informations mandataires à l'intention du récepteur du message, il contient les méthodes et les paramètres qui seront exécutés par le destinataire final.
- **SOAP fault** (erreur) est un élément facultatif défini dans le corps SOAP et qui est utilisé pour reporter les erreurs. [12]

VI-3 WSDL : Web Services Description Language

Un document WSDL se compose d'un ensemble d'éléments décrivant les types de données utilisés par le service, les messages que le service peut recevoir, ainsi que les liaisons SOAP associées à chaque message. Le schéma suivant illustre la structure du langage WSDL qui est un document XML, en décrivant les relations entre les sections constituant un document WSDL.

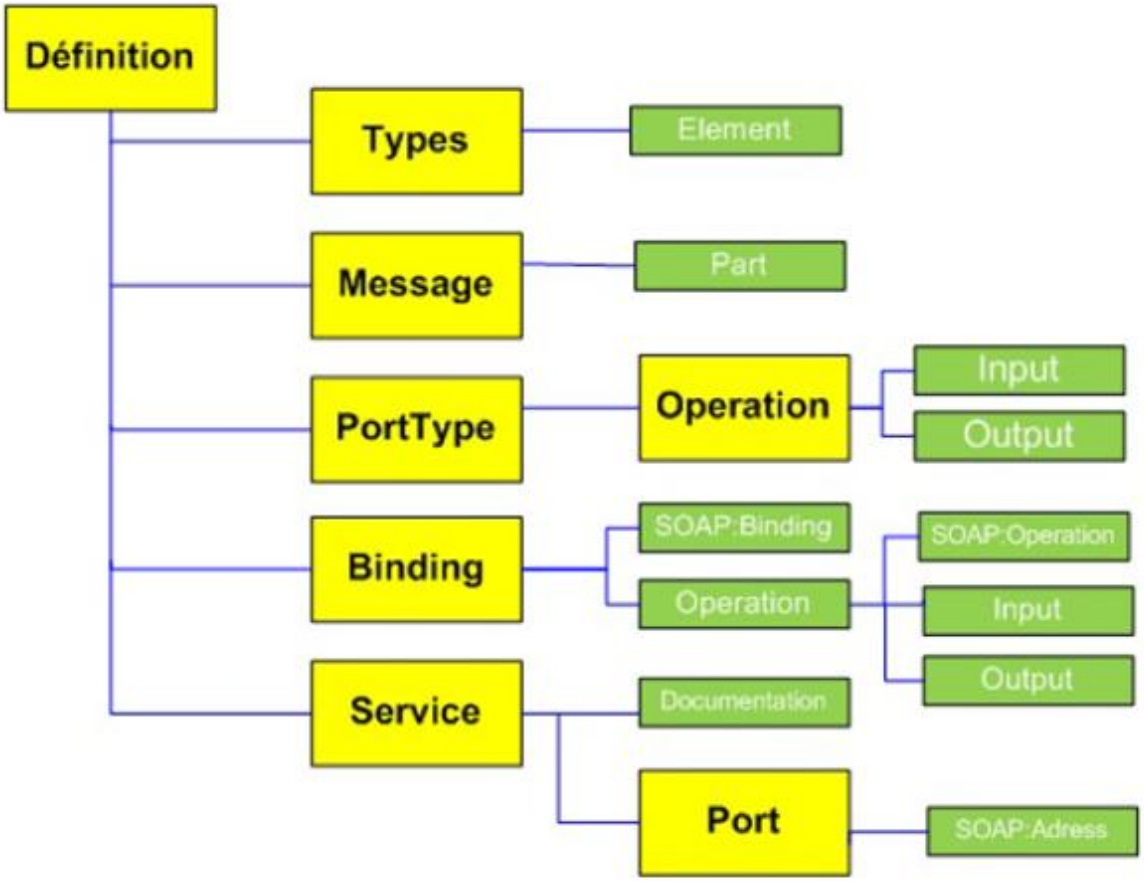

Figure I.4 : Structure d'un document WSDL [12]

Un fichier WSDL contient donc sept éléments.

- **Types** : fournit la définition de types de données utilisés pour décrire les messages échangés.
- **Messages** : représente une définition abstraire (noms et types) des données en cours de transmission.
- **PortTypes** : décrit un ensemble d'opérations. Chaque opération a zéro ou un message en entrée, zéro ou plusieurs messages de sortie ou d'erreurs.

- **Binding** : spécifie une liaison entre un <portType> et un protocole concret (SOAP, HTTP...).
- **Service** : indique les adresses de port de chaque liaison.
- **Port** : représente un point d'accès de services défini par une adresse réseau et une liaison.
- **Opération** : c'est la description d'une action exposée dans le port.

Le document WSDL peut être divisé en deux parties. Une partie pour les **définitions abstraites**, tandis que la deuxième contient les **descriptions concrètes**.

La description concrète est composée des éléments qui sont orientés vers le client pour le service physique. Les trois éléments concrets XML présents dans un WSDL sont :
<wsdl:service> ;
<wsdl:port> ;
<wsdl:binding>.
La description abstraite est composée des éléments qui sont orientés vers la description des capacités du service Web. Ses éléments abstraits définissent les messages SOAP de façon totalement indépendante de la plate-forme et de la langue. Cela facilite la définition d'un ensemble de services pouvant être implémentés par différents sites Web.
Les quatre éléments abstraits XML qui peuvent être définis dans un WSDL sont :
<wsdl:types> ;
<wsdl:message> ;
<wsdl:operation> ;
<wsdl:portType>. [12]

VI-4 UDDI : Universal Description Discovery and Integration

L'annuaire UDDI se concentre sur le processus de découverte de SOA (Architecture Orienté Service) [20], et utilise des technologies standards telles que XML, SOAP et WSDL qui permettent de simplifier la collaboration entre partenaires dans le cadre des échanges commerciaux. L'accès au référentiel s'effectue de différentes manières.
Les pages blanches comprennent la liste des entreprises ainsi que des informations associées à ces dernières (coordonnées, description de l'entreprise, identifiants...).
Les pages jaunes recensent les services Web de chacune des entreprises sous le standard WSDL.

Les pages vertes fournissent des informations techniques précises sur les services fournis.

Les entreprises publient les descriptions de leurs services Web en UDDI, sous la forme de fichiers WSDL. Ainsi, les clients peuvent plus facilement rechercher les services Web dont ils ont besoin en interrogeant le registre UDDI.

Lorsqu'un client trouve une description de service Web qui lui convient, il télécharge son fichier WSDL depuis le registre UDDI.

Ensuite, à partir des informations inscrites dans le fichier WSDL, notamment la référence vers le service Web, le client peut invoquer le service Web et lui demande d'exécuter certaines de ses fonctionnalités.

Le scénario classique d'utilisation d'UDDI est illustré ci-dessous. L'entreprise B a publié le service Web S, et l'entreprise A est client de ce service :

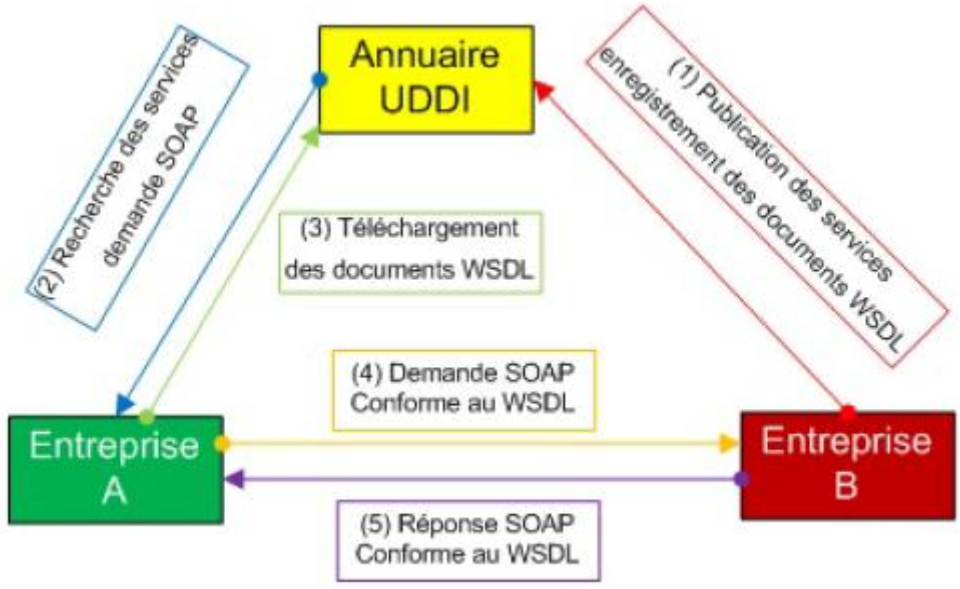

Figure I.5 : Schéma Général de l'annuaire UDDI [12]

a- Structures de données UDDI

Un registre UDDI se compose de quatre types de structures de données, le **businessEntity**, le **businessService**, le **bindingTemplate** et la **tModel**. Cette répartition par type fournit des partitions simples pour faciliter la localisation rapide et la compréhension des différentes informations qui constituent un enregistrement.

b- BusinessEntity (entité d'affaires)

Les « businessEntities » sont en quelque sorte les pages blanches d'un annuaire UDDI. Elles décrivent les organisations ayant publié des services dans le répertoire. On y

trouve notamment le nom de l'organisation, ses adresses (physiques et Web), des éléments de classification, une liste de contacts ainsi que d'autres informations.

c- BusinessService (service d'affaires)

Les « businessServices » sont en quelque sorte les pages jaunes d'un annuaire UDDI. Elles décrivent de manière non technique les services proposés par les différentes organisations. On y trouve essentiellement le nom et la description textuelle des services ainsi qu'une référence à l'organisation proposant le service et un ou plusieurs « bindingTemplate ».

d- BindingTemplate (modèle de rattachement)

UDDI permet de décrire des services Web utilisant HTTP, mais également des services invoqués par d'autres moyens (SMTP, FTP...). Les « bindingTemplates » donnent les coordonnées des services. Ce sont les pages vertes de l'annuaire UDDI. Ils contiennent notamment une description, la définition du point d'accès (une URL (Uniform Resource Locator) [21]) et les éventuels « tModels » associés.

e- tModel (index)

Les « tModels » sont les descriptions techniques des services. UDDI n'impose aucun format pour ces descriptions qui peuvent être publiées sous n'importe quelle forme et notamment sous forme de documents textuels (XHTML (Extensible HyperText Markup Language) [22], par exemple). C'est à ce niveau que WSDL intervient comme le vocabulaire de choix pour publier des descriptions techniques de services. [12]

VII Les principales problématiques associées aux services web

Les actuels sujets de recherche dans le domaine des services web sont nombreux. Un nombre considérable d'études tournent autour de la découverte des services et ses sujets rattachés comme sont la sélection et la substitution. Dans cette section nous allons analyser et décrire les travaux les plus pertinents.

VII-1 La découverte des services Web

Dans le contexte d'une application qui a besoin d'exécuter une fonctionnalité implémentée comme un service web par plusieurs fournisseurs, la découverte fait référence au processus de recherche des services web implémentant la fonctionnalité souhaitée. Les registres UDDI sont entités qui servent d'appui à la découverte de

services web pour les applications client. De cette façon une application interroge un registre UDDI pour les fournisseurs d'un service web. [19]

VII-2 La composition des services web :

La composition de services web est considérée comme une révolution qui permet la distribution, sur le web, non seulement des données et des documents mais aussi des applications. Un service web est une interface qui décrit une collection d'opérations qui sont accessibles à travers des messages XML. La composition de service web ne consiste pas à composer les services appartenant à un seul partenaire mais plutôt d'aller plus loin et chercher les services des autres partenaires. Ce nouveau paradigme se focalise sur la composition des services complexes à partir des services élémentaires se trouvant sur Internet.

Ainsi, les services Web composés constituent différents composants mis en application à différents endroits. Ces services s'exécutent dans différents contextes, cependant ils ont besoin de communiquer pour avoir le comportement désiré. En effet, La composition peut être statique ; cela veut dire que les services interagissent entre eux d'une manière prédéfinie.

Elle peut être aussi dynamique ; c'est à dire que les services sont découverts les uns des autres. [23]

VII-3 La sélection des services Web

Avec la sélection des services web, on cherche à choisir le meilleur fournisseur d'un service web, étant donné un ensemble de fournisseurs de ce service.

Il y a une proposition de base sur la Qualité de Service (i.e. QoS en anglais Quality of Service – QoS). La plate-forme de cette proposition est un modèle réglé qui peut coexister avec les registres UDDI dérégulés. Les registres dérégulés actuels peuvent fournir des services aux gens pour qui la qualité de service n'est pas importante. Les registres réglés basés sur ce modèle peuvent servir aux applications qui ont besoin de qualité de service.

Il y a quatre rôles dans ce modèle (Figure I.6):

- Fournisseur de services web.

- Consommateur de services web.
- *Certificateur* de la Qualité de Service.
- Et le nouvel registre.

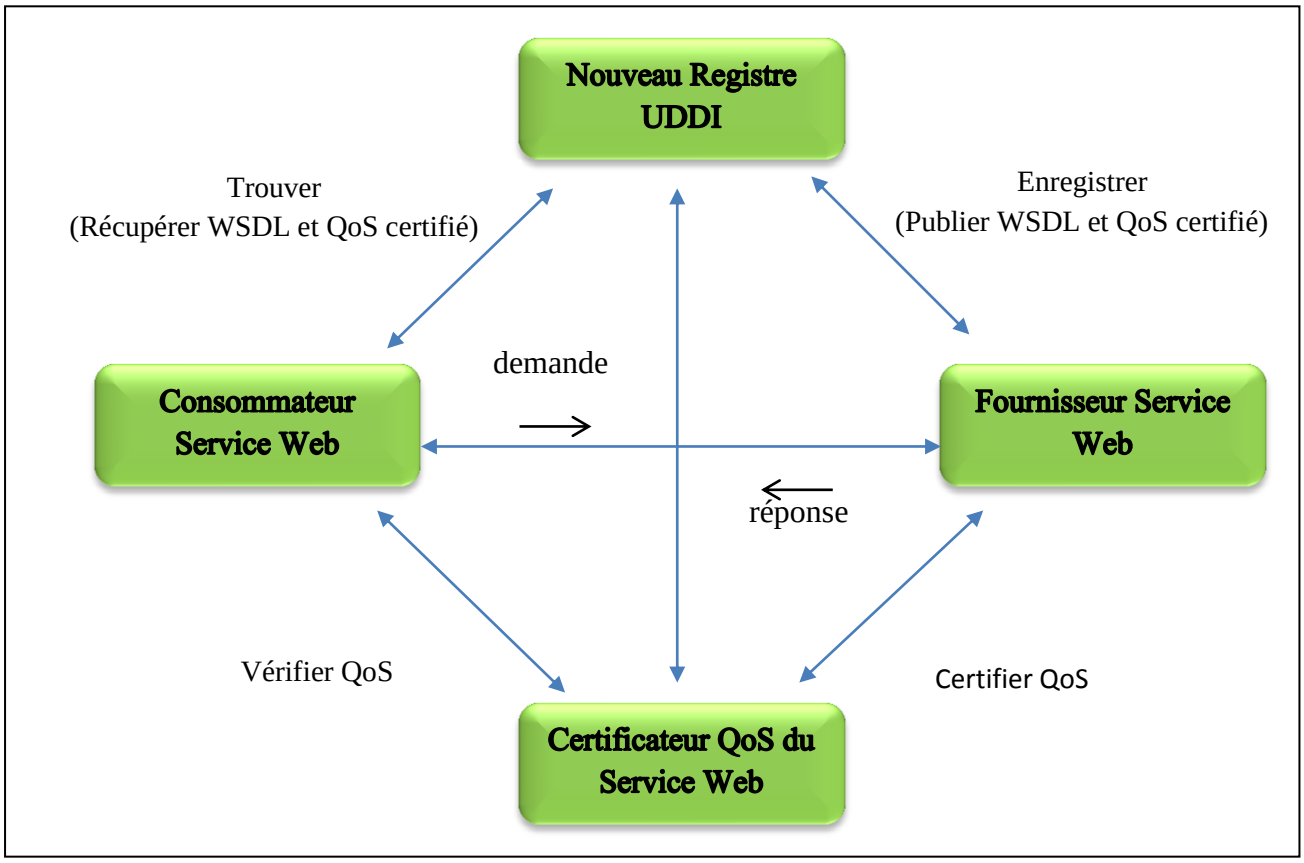

Figure I.6 : Un nouveau modèle de registre et découverte de services web [19]

Le fournisseur du service offre le service web en publiant ce dernier dans le nouveau registre; le consommateur a besoin du service web offert par le fournisseur ; le nouvel registre UDDI est un lieu de stockage de services web enregistrés avec facilités de recherche. Le nouveau rôle certificateur sert à vérifier la qualité de service du fournisseur du service. Le nouveau registre est différent de l'actuel modèle UDDI en ayant information sur la description fonctionnelle du service web, et sur la qualité de service enregistré dans le stockage.

VII-4 La substitution des services Web

Il y a besoin de la substitution d'un service web par un autre dans les cas suivants :

- Pendant l'exécution, le service web échoue ou ne peut pas être atteint.

- Il y a un nouveau service web qui fournit le même fonction mais avec une meilleure QoS.
- Une nouvelle version d'un service web choisi, offert par le même fournisseur est disponible.

La substitution peut poser parfois des problèmes orienté contexte. Par exemple, il est possible de trouver deux services web similaires d'un point de vue fonctionnel (les deux acceptent des modèles de voitures et fournissent les prix correspondant), mais qui ont des contextes différents ; par exemple le premier donne des prix en euro et le deuxième donne des prix en dollars. Dans ce cas-là, il faut adapter le contexte durant la substitution. [19]

VIII Quelques domaines d'application de services Web

- Les services Web peuvent être utiles dans la plupart des scénarios applicatifs lorsque la communication peut être établit sur un modèle bidirectionnel (requête/réponse).
- L'application des services Web est multiple, autant dans les domaines du B2B (Business to Business) [24], B2C (Business to Consumer) [25] que pour les domaines de gestion de stock, etc.
- Jusqu'à aujourd'hui, la plupart des B2B transactions est basée sur EDI qui récemment assouplies via des standards basés sur XML. [13]

IX Avantages et limites

IX-1 Avantage :

- Les Web Services possèdent une simplicité de mise en œuvre : Ils rendent en effet accessibles depuis Internet des fonctionnalités d'une application existante tout en ne modifiant pas en profondeur le système d'information de l'entreprise.
- Les services Web avec ses protocoles et ses standards avance toujours vers plus de normalisation.
- Les Web Services reposent sur des bases solides (SOAP et WSDL) qui ont prouvé leur efficacité et leur maturité même si une normalisation complète n'existe pas encore.

- Cette standardisation permet une grande interopérabilité entre des applications de technologie différente.
- Ces standards assurent une grande indépendance de l'implémentation par rapport au système d'exploitation, à l'architecture de la machine ou au standard utilisé.
- Un des avantages principaux des Web Services est qu'ils sont basé sur Internet qui est on le sait fiable et mature. [5]

IX-2 Limites

Les Web Services ont cependant des limites qui sont :

- Manque de sécurité au niveau de l'interaction avec les Web Services. En effet les standards qui existent peuvent très bien ne pas être utilisés ou peuvent n'assurer qu'une partie des exigences de sécurité.
- En ce qui concerne les performances des Web Services, l'échange de messages est particulièrement lent du fait que la quantité d'informations véhiculée est importante.
- Des contournements de sécurité sont possibles i.e., les Web Services utilisent HTTP et héritent des problèmes de celui-ci : Les firewalls peuvent être contournés. [5]

X Conclusion

L'introduction des Web Services a donc changé l'approche des applications en effectuant le passage d'une architecture orienté objet à une architecture orienté service.

Aujourd'hui, on trouve un nombre conséquent de serveurs différents, de systèmes différents et le but serait donc de pouvoir faire communiquer tous ces machines entre elles. Grâce à des protocoles standardisés, les Web Services permettent une interopérabilité entre ces différents systèmes hétérogènes.

Mais les Web Services ont donc aussi apportés de nouvelles contraintes en matière de sécurité. Pour essayer de palier à ces contraintes, des protocoles de sécurité ont été mis en place dont WS-Security permettant la signature et le chiffrement des messages.

La fédération entre entreprises et les scénarios B2B sont amenés à s'agrandir. Les sociétés auront de plus en plus à collaborer ensemble. Le partage d'application Web pour les collaborateurs ou les clients par exemple va être de plus en plus démocratisé.

Même s'il reste encore des améliorations à effectuer au niveau de la sécurité notamment, les Web Services semblent être la solution appropriée pour résoudre les problèmes d'échange de données entre les systèmes hétérogènes et l'intégration d'applications.

Chapitre 2 : Conception et Implémentation du prototype

I Introduction :

Notre travail consiste à développer un système de sélection des services web à base de qualité de service dénommé FA_APP (Fouzi et Ayoub APPlication).

L'objectif de ce dernier chapitre est de présenter notre approche pour résoudre le problème de sélection.

Donc, nous allons voir les différentes étapes suivies durant la réalisation de notre application en commençant, par une présentation de la notion de la sélection où nous allons définir les qualités de services web, leurs critères ainsi que la fonction objective. Ensuite, nous citons les différents outils et environnements de développement utilisés. Puis, nous décrivons la base de sélection. Enfin, nous finissons par présenter les différents diagrammes de conception et algorithmes utilisés ainsi que l'application et les résultats obtenus.

II Sélection des services web

La migration des systèmes d'information des entreprises vers un schéma orienté Business to Business (i.e., des entreprises mettant leurs services à disposition d'autres entreprises sous forme de composants logiciels) a favorisé l'émergence des services Web depuis le début des années 2000. Par exemple, une compagnie aérienne peut mettre en ligne à disposition des agences de voyage un service logiciel de réservation de vols. Pour fournir une valeur ajoutée à l'utilisateur final, une agence de voyage peut décider de composer plusieurs services Web afin de faciliter l'organisation d'un séjour. Par exemple, une agence peut proposer un pack „‟déplacement„‟ comportant d'une part la réservation d'un billet pour un événement sportif, d'autre part une réservation de billet de train ou d'avion. On parle alors de composition de services Web.

Il arrive très fréquemment que plusieurs services répondent à un même ensemble de besoins fonctionnels. Par exemple, pour réaliser la réservation d'une place au stade, un utilisateur peut mettre en concurrence plusieurs services de billetterie. Une sélection de services Web consiste alors à assigner un service Web à chaque classe. Cette affectation est appelée plan d'exécution. L'ensemble des plans d'exécution possibles est combinatoire : pour N classes avec un choix parmi M services pour chacune, il y a M^N

compositions possibles. Afin de discriminer parmi ces plans d'exécution, on utilise la notion de qualité de service. [28]

II-1 Qualité des services 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).

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. [29]

II-2 Critères de qualité des services web

Dans ce qui suit, nous présentons l'ensemble des attributs de QoS génériques. Les attributs les plus communs sont décrits par les paramètres liés à la performance.

– Le débit : Le nombre de requêtes servies par unité de temps.

– Le temps de réponse : Le temps séparant l'émission de la requête et la réception des résultats.

– La fiabilité : La capacité d'un service d'exécuter correctement ses fonctions.

– La scalabilité : La capacité du service de traiter le plus grand nombre d'opérations ou de transactions pendant une période donnée tout en gardant les mêmes performances.

– La robustesse : La probabilité qu'un service peut réagir proprement à des messages d'entrées invalides, incomplets ou conflictuels.

– La disponibilité : La probabilité d'accessibilité d'un service.

– Le prix d'exécution : C'est le prix qu'un client du service doit payer pour bénéficier du service.

– La réputation : C'est une mesure de la crédibilité du service. Elle dépend principalement des expériences d'utilisateurs finaux.

– La sécurité : C'est un regroupement d'un ensemble de qualités à savoir : la confidentialité, le cryptage des messages et le contrôle d'accès.

II-3 Fonction objective

Dans l'ordre d'évaluer la qualité multidimensionnelle du web service composite, une fonction utilitaire est utilisée. Celle-ci peut faciliter la comparaison entre les qualités de chaque composition de web services.

Le calcul de la fonction objective nécessite le calcul de valeurs maximum Q_i^{max} et minimum Q_i^{min} des attributs de QoS

Le présent processus est ensuite suivi par le processus de weighting pour représenter les priorités et les préférences des clients, puisque les utilisateurs peuvent avoir des préférences sur le résultat de leurs requêtes, ils peuvent spécifier l'importance relative à chaque paramètre de QoS. Nous attribuons des poids rangé entre 0 et 1 pour chaque paramètre de QoS pour refléter le niveau de leur importance.

Dans notre étude nous supposons que les poids sont égaux. Nous utilisons le score de la fonction objective F pour évaluer la qualité de la composition des web services C. Par l'utilisation de la fonction objective, l'optimisation du QoS est de trouver la composition des web service C avec un maximum score.

$$F(C) = \sum_{Q_i \in Neg} W_i \frac{Q_i^{max} - Q_i(C)}{Q_i^{max} - Q_i^{min}} + \sum_{Q_i \in Pos} W_i \frac{Q_i(C) - Q_i^{min}}{Q_i^{max} - Q_i^{min}}$$

Tel que $W_i \in R_0^+$ et $\sum_{i=1}^{n} W_i = 1$ c'est les poids de chaque paramètres pour représenter les priorités de l'utilisateur.

Neg : représente l'ensemble des critères négatifs tels que le coût et le temps d'exécution.

Pos : représente l'ensemble des critères positifs tels que la disponibilité, la fiabilité et la réputation.

III Outils et environnement de développement

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

III-1 Java

L'application est développée en utilisant le langage de programmation Java. Java est un langage de programmation orienté objet et familier. Sa simplicité, sa robustesse, sa portabilité ainsi que sa performance lui ont permis d'être le choix préféré pour le développement de notre application.

III-2 NetBeans

Nous avons écrit notre application en Netbeans version 7.3. NetBeans est l'environnement de développement intégré (IDE) pour Java proposé par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License), et le concurrent direct d'Eclipse sur le marché des environnements Open Source. En tirant avantage de cette trousse à outils gratuite, basée sur des standards, les développeurs peuvent concevoir des applications complexes plus rapidement, avec une plus grande assurance de robustesse et de concevoir des applications qui résisteront à l'épreuve du temps. Pour cela le choix de Netbeans était fondamental.

III-3 L'API JDOM

JDOM (Java-based Document Object Model) est une API open source Java, vue comme un modèle de documents objets dont le but est de représenter et manipuler un document XML de manière intuitive pour un développeur Java sans requérir une connaissance pointue de XML. JDOM permet aussi de vérifier que les données contenues dans les éléments respectent la norme XML. La simplicité d'utilisation de JDOM lui permet d'être une API dont l'utilisation est assez répandue.

III-4 JFreeChart

JFreeChart est une bibliothèque open source qui permet d'afficher des données statistiques sous la forme de graphiques. Elle possède plusieurs formats dont le camembert, les barres ou les lignes et propose de nombreuses options de configuration pour personnaliser le rendu des graphiques. Elle permet également d'exporter le graphique sous la forme d'une image.

IV Description de l'application

Pour cela, nous proposant une méthode de résolution composée des étapes suivantes :

- Rechercher des éléments Skylines de chaque classe.
- Création d'une arborescence de clusters des éléments Skylines de chaque classe abstraite.
- Parcours de N arborescences (niveau par niveau) et identification des compositions concrètes qui vérifient les contraintes globales.
- Affichage des Top K meilleures solutions.

IV-1 Description de la base

Nous avons créé un schéma de service qui contient dix (10) classes de services web. Le nombre d'instances dans chaque classe et quarante (40) fournisseurs de services web. Chaque fournisseur se caractérise par des qualités de service. Nous utilisons cinq (5) paramètres pour évaluer les opérations des services : coût, latence, disponibilité, fiabilité et réputation. Les valeurs de ces paramètres sont générées en se basant sur une distribution uniforme. Le tableau II.1 représente les intervalles des valeurs prises par les 05 paramètres précédentes :

Critère	Intervalle
Coût	0 – 30 $
Temps d'exécution	0 – 300 ms
Disponibilité	0.7 – 1
Fiabilité	0.5 – 1
Réputation	0 – 5

Tableau II.1 : L'intervalle des valeurs prises par les critères de QoS

Un échantillon de notre base est illustré dans le tableau II.2 qui représente le fournisseur i des cinq premières classes.

Attribut	Op 1	Op 2	Op 3	Op 4	Op 5	Op 6	Op 7	Op 8	Op 9	Op 10
Coût	-22.3	-26.7	-29.6	-1.0	-16.7	-7.2	-23.3	-18.7	-7.9	-18.1
Latence	-223.1	-222.7	-255.7	-93.5	-13.2	-214.2	-35.2	-108.8	-9.7	-135.1
Disponibilité	-0.4	-0.1	-0.3	-0.5	-0.3	-0.08	-0.2	-0.5	-0.3	-0.4
Fiabilité	-0.1	-0.1	-0.08	-0.06	-0.3	-0.01	-0.2	-0.1	-0.06	-0.1
Réputation	4.1	0.6	3.7	2.3	2.03	1.3	1.8	3.8	1.8	2.8

Tableau II.2 : Un exemple de base de données [32]

Les valeurs des différents critères sont négatives parce qu'on stocke leurs logarithmes népériens.

IV-2 Description de la requête

La sélection d'une composition de service est formalisée comme suit :

- L'utilisateur a besoin de chercher une composition C constituée de N étapes ou tâches.
- Chaque tâche peut être modélisée par un ou plusieurs instances de service.
- Chaque service est caractérisé par cinq valeurs de QoS (le coût, la latence, la disponibilité, la fiabilité et la réputation).
- L'utilisateur doit maximiser la disponibilité, la fiabilité et la réputation et minimiser le coût et la latence.

- La solution recherchée doit satisfaire les cinq contraintes globales. Elles sont spécifiées comme suit :

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

V Conception :

V-1 Les algorithmes proposés :

a- Algorithme BBS :

Les traitements des requêtes Skylines sont fondamentaux pour la plupart des applications qui se basent sur les décisions multicritères.

Il existe plusieurs algorithmes pour cela. Parmi eux, l'algorithme BBS (Branch and Bound Skyline) [27] qui est un algorithme développé par Papadias et al. Il a été développé pour améliorer les inconvénients de l'algorithme NN (Nearest Neighbours) qui est un autre algorithme des traitements des requêtes Skylines.

Cet algorithme se base sur la technique du premier meilleur résultat en gardant tous les entrées nécessaires dans une pile jusqu'à ce qu'ils ne soient plus utiles.

Parmi les points forts de cet algorithme est qu'il assure l'optimalité des E/S (Entrées / Sorties) parce qu'il ne visite que les nœuds qui peuvent contenir les points skylines, et qu'il n'a pas accès au même nœud deux fois. [30]

Ci-dessous le pseudo-code de cet algorithme (Figure II.1) :

1. $S = \emptyset$ // la liste des points skylines.
2. **Insérer** toutes les entrées de la racine de la base R dans la pile P.
3. **Tant que** P n'est pas vide
   3.1. **Supprimer** la tête e de P
   3.2. **Si** e n'est pas dominé par un autre point de S **Alors**
      3.2.1. **Supprimer** l'entrée e.
   3.3. **Sinon** // e n'est pas dominé
      3.3.1. **Si** e est une entrée intermédiaire **Alors**
         3.3.1.1. **Pour** chaque e_i fils de e **Faire**
            3.3.1.1.1. **Si** e_i n'est pas dominé par un point de S **Alors**
               3.3.1.1.1.1. **Insérer** e_i dans la pile P.
            3.3.1.1.2. **Sinon** // e est un point skyline
               3.3.1.1.2.1. **Insérer** e_i dans S
            3.3.1.1.3. **Fin Si**
         3.3.1.2. **Fin Pour**
      3.3.2. **Fin Si**
   3.4. **Fin Si**
4. **Fin Tant que**
5. **Fin**

Figure II.1 : Le pseudo code de l'algorithme BBS

b- Algorithme agglomératif:

Les algorithmes de clustering hiérarchiques fonctionnent itérativement en regroupant à chaque étape les clusters les plus similaires (bottom-up, ou agglomératif), ou en scindant les groupes les moins homogènes (top-down, divisive). Les algorithmes agglomératifs sont tous basés sur la combinaison des éléments suivants (Kaufmann, 1990) :

- un critère pour estimer la similarité entre éléments

- une méthode pour estimer la similarité entre clusters

- un algorithme de fusion de clusters

- une méthode pour choisir un représentant d'un cluster. [31]

Ci-dessous le pseudo-code de cet algorithme (Figure II.2) :

1. **Calculer** la matrice de similarité
2. **Affecter** chaque donnée à un cluster
3. **Répéter**
   3.1. **Fusionner** les deux clusters les plus proches
   3.2. **Mise à jour** de la matrice de similarité
4. **Jusqu'à ce qu'**il reste un seul cluster
5. **Fin**

Figure II.2 : Le pseudo code de l'algorithme agglomératif

V-2 Diagramme de cas d'utilisation

La Figure II.3 représente le diagramme de cas d'utilisation de notre application :

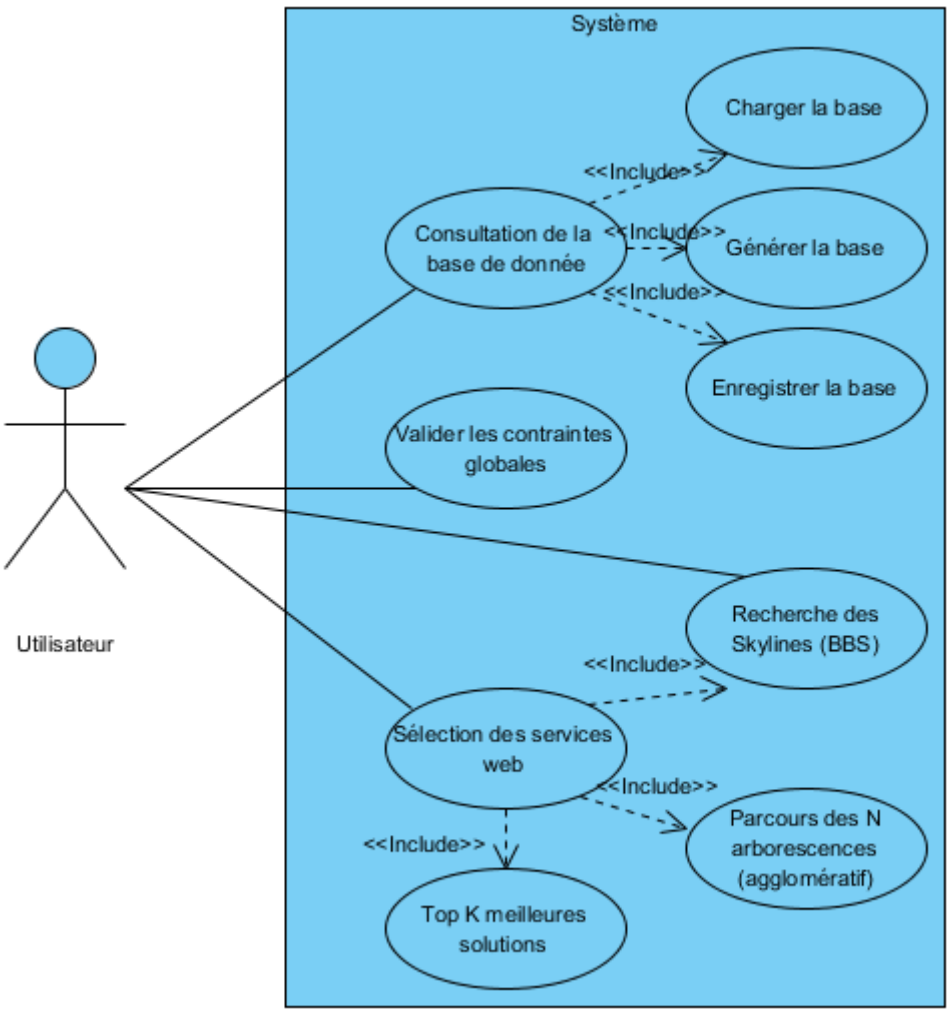

Figure II.3 : Diagramme de cas d'utilisation.

V-3 Diagramme de classe

La Figure II.4 représente le diagramme de classe de notre application :

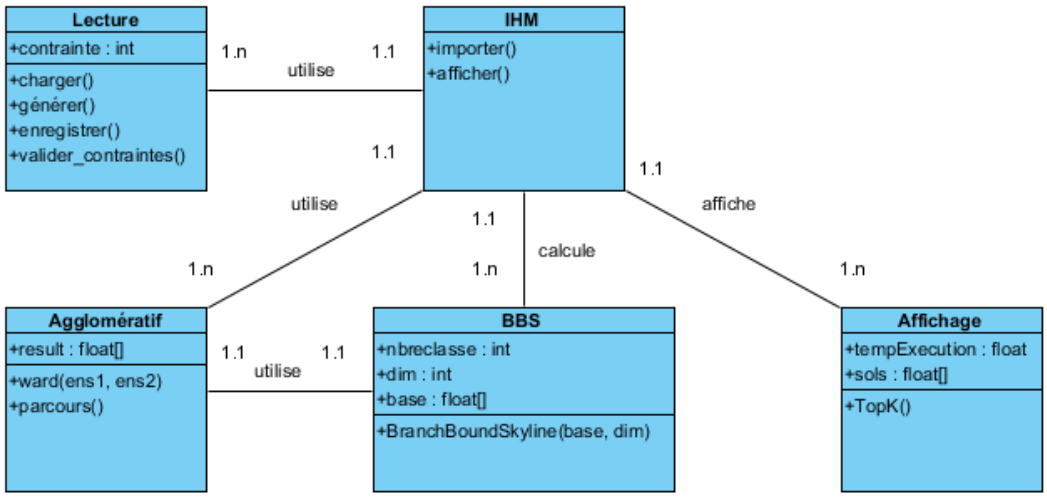

Figure II.4 : Diagramme de classe.

V-4 Diagramme de séquence

La Figure II.5 représente un exemple de diagramme de séquence de notre application :

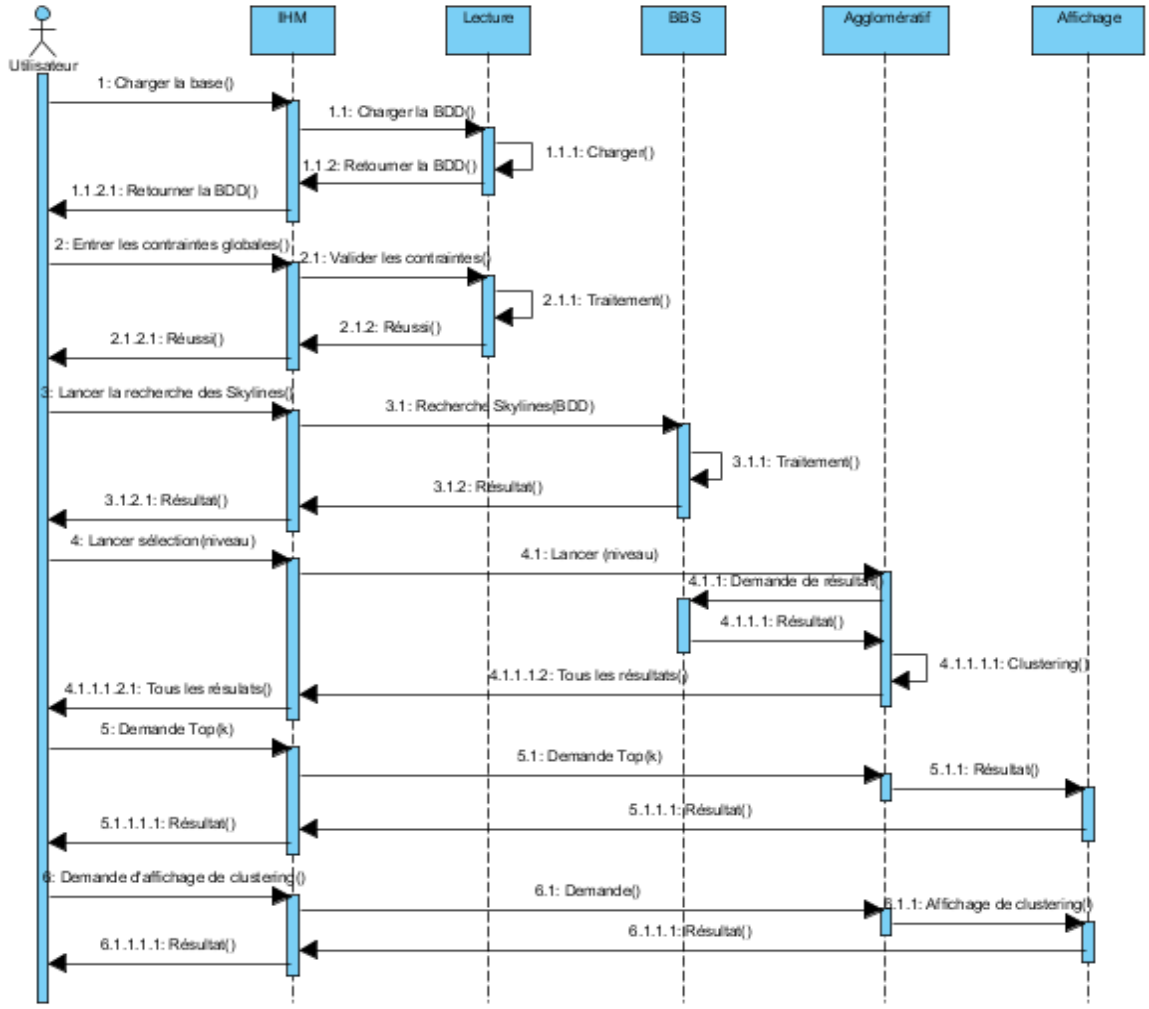

Figure II.5 : Diagramme de séquence.

VI Présentation de l'IHM :

Dans ce qui suit nous présentons les interfaces graphiques de notre prototype.

VI-1 Chargement de la base des services web

La première étape consiste à charger la base des services web (Figure II.6).

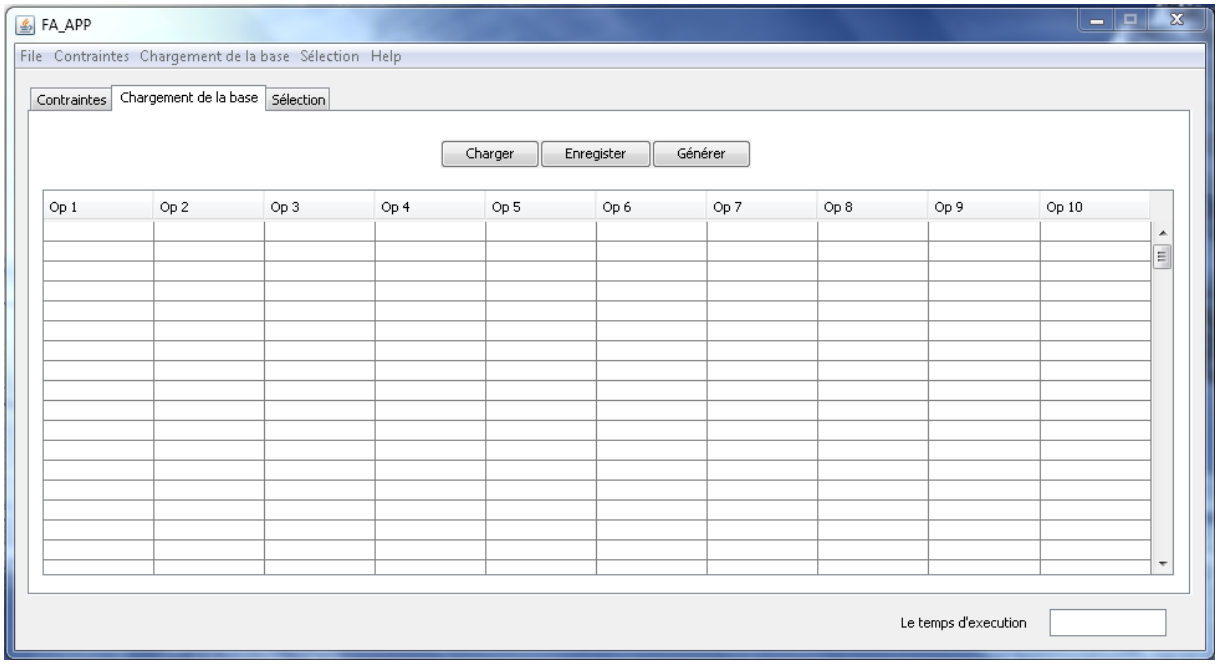

Figure II.6 : Chargement de la base

Cet onglet contient :

- 3 boutons : un bouton Charger pour charger la base en indiquant le fichier .xml qui contient les données, un autre bouton Enregistrer afin de sauvegarder la base dans un fichier .xml et un bouton Générer pour générer une nouvelle base au hasard.
- Un tableau pour afficher la base.

VI-2 Valider les contraintes globales

Une fois la base des services est chargée, l'utilisateur pourra formuler ces contraintes globales (Figure II.7).

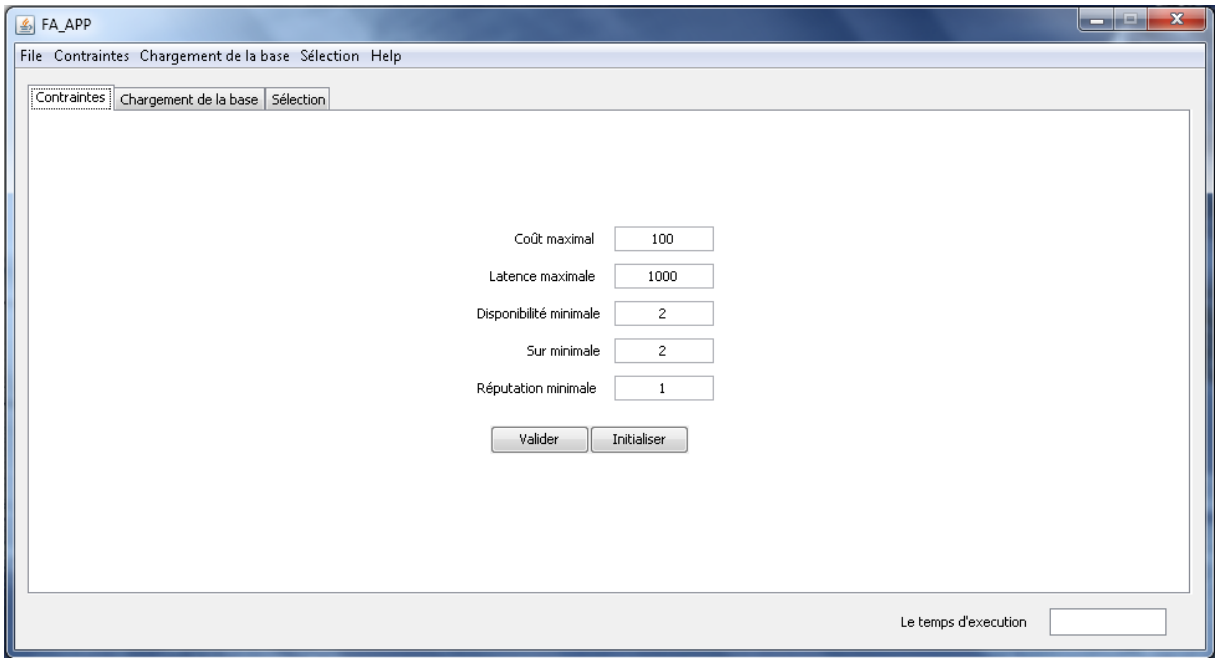

Figure II.7 : Saisie et validation des contraintes globales

VI-3 L'exécution

L'exécution de l'algorithme BBS (qui donnera les points Skylines) et agglomératif avec validation de nombre de niveau, et en fin les k meilleures solutions retourné (Figure II.8).

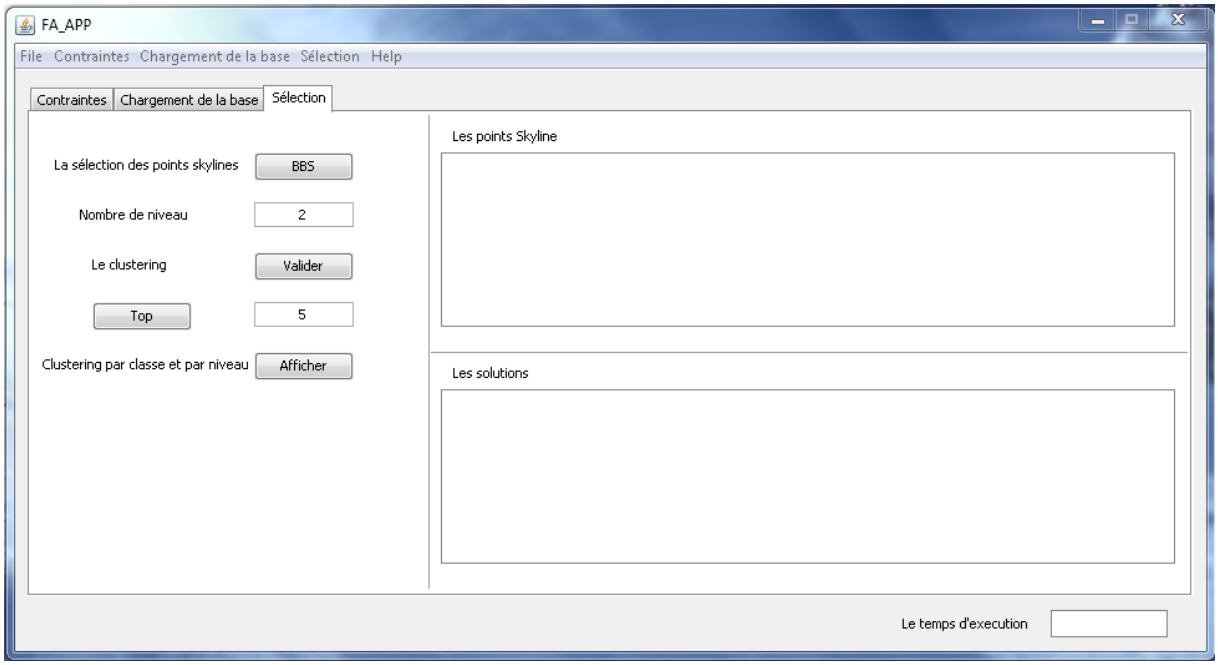

Figure II.8 : Sélection des meilleurs services web

La dernière étape consiste à :

- D'abord, lancer la sélection des points skylines à l'aide du bouton BBS.
- Ensuite, entrer le nombre de niveau maximal dans le champ de texte et lancer le clustering via le bouton Valider.
- Pour afficher le top k solutions dans chaque niveau possible, il faudra entrer le paramètre k dans le champ de texte et valider en cliquant sur le bouton Top.
- De plus, un bouton Afficher sert à afficher le clustering par niveau et par classe.
- Enfin, le temps d'exécution nécessaire, pour calculer et afficher les meilleures solutions à l'utilisateur, est affiché dans le champ de texte qui est en bas.

VI-4 Exceptions:

La Figure II.9 montre une des exceptions implémentées dans notre application afin d'éviter les messages d'erreurs de Netbeans.

Ce message est affiché lorsqu'on clique sur le bouton de sélection des points skylines et que la base est toujours vide.

Figure II.9 : Message d'erreur retourné lorsqu'on la base est vide.

VII Expérimentation

Nous avons mené une expérience pour évaluer la performance de l'approche proposée. Nous utilisons le scénario introduit dans le chapitre II pour expérimenter l'approche. 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 7.3 de Sun Microsystems, la machine d'expérimentation possède les caractéristiques suivantes :

- le système d'exploitation est Windows 7 (64 bit),
- Processeur Intel Core i3,

- 4 Giga de RAM.

Nous considérons les contraintes globales suivantes :

Coût = 100 ; Latence = 1000 ; Disponibilité = 2 ; Fiabilité = 2 ; Réputation = 1.

Nous avons obtenus les résultats suivants pour les 3 premiers niveaux (Figure II.10 et II.11) :

Pour le niveau 0, aucune solution n'a été trouvée.

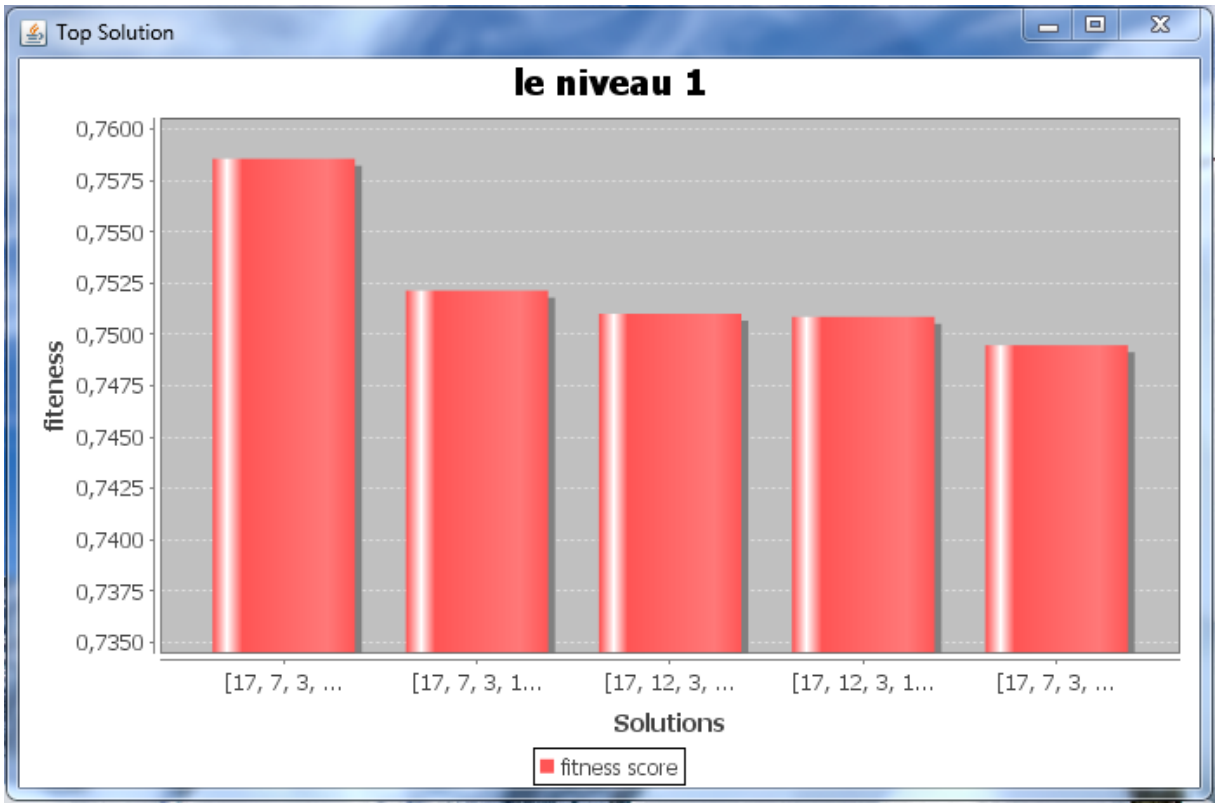

Figure II.10 : Les meilleures solutions pour le niveau 1

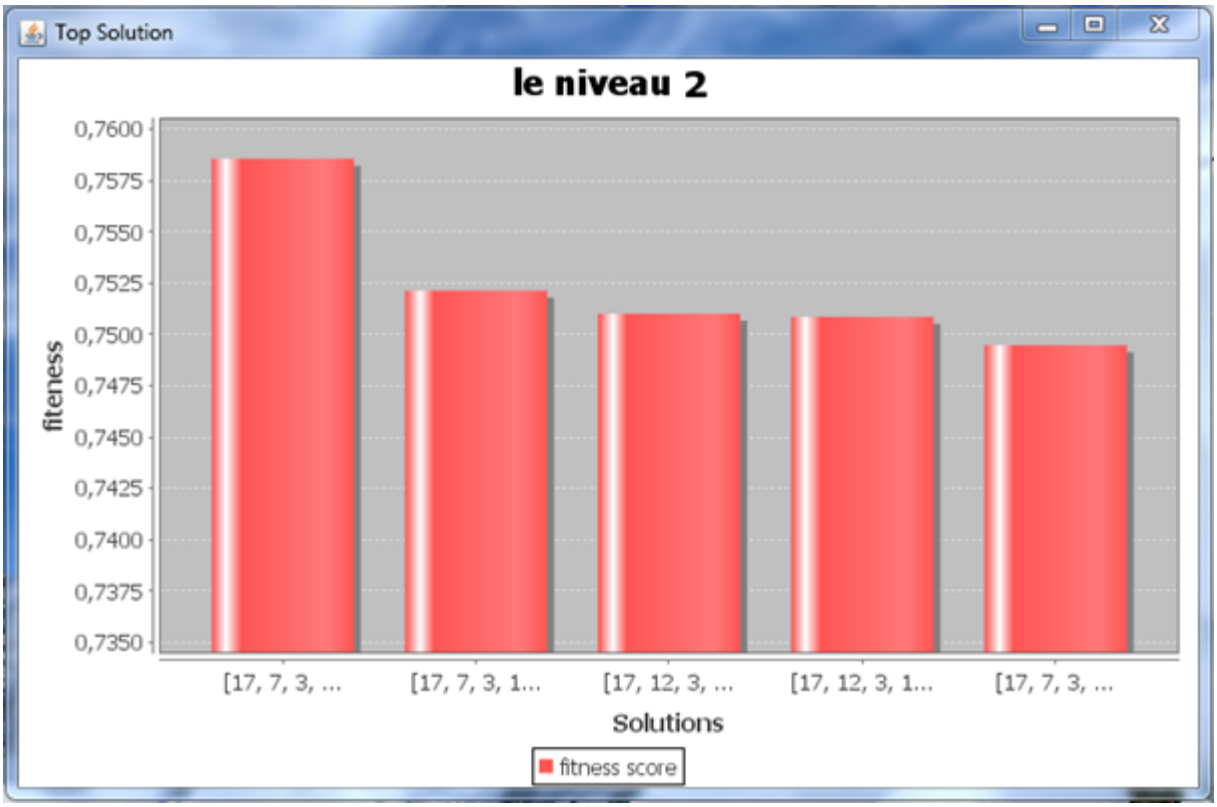

Figure II.11 : Les meilleures solutions pour le niveau 2

Nous remarquons que les meilleures résultats du niveau 2 sont les mêmes du niveau 1. Donc, nous trouverons l'optimum global dans les premiers niveaux.

Ci-dessous, la Figure II.12 qui montre le temps d'exécution nécessaire pour calculer les solutions possibles trouvées dans les 4 premiers niveaux.

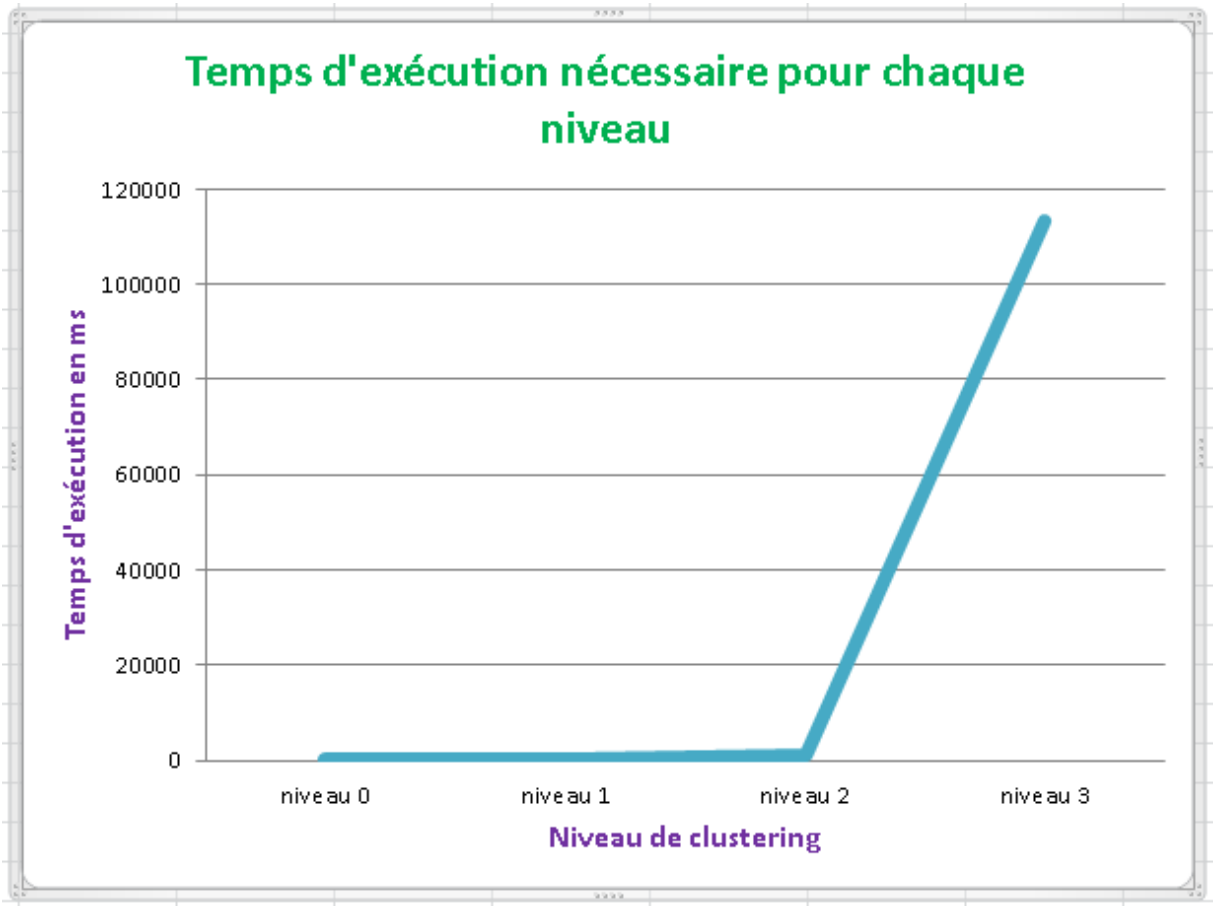

Figure II.12 : Temps d'exécution nécessaire pour les 4 premiers niveaux

VIII Discussion

D'après nos expériences on conclut les points suivants :

- Plus le nombre de niveau est petit mieux seront les résultats (en termes de temps), c'est-à-dire on trouvera les meilleures solutions dans les premiers niveaux et on n'aura pas besoin de parcourir tous les niveaux.
- Plus les contraintes globales sont serrées plus on parcourt de nouveaux niveaux
- Le temps d'exécution est toujours fixé pour une même requête et il est nettement inférieur par rapport aux techniques d'optimisation globales et mono objectives (telles que SPO et sélection clonale).
  La Figure II.13 montre le temps d'exécution nécessaire pour chaque technique afin d'atteindre l'optimum global.

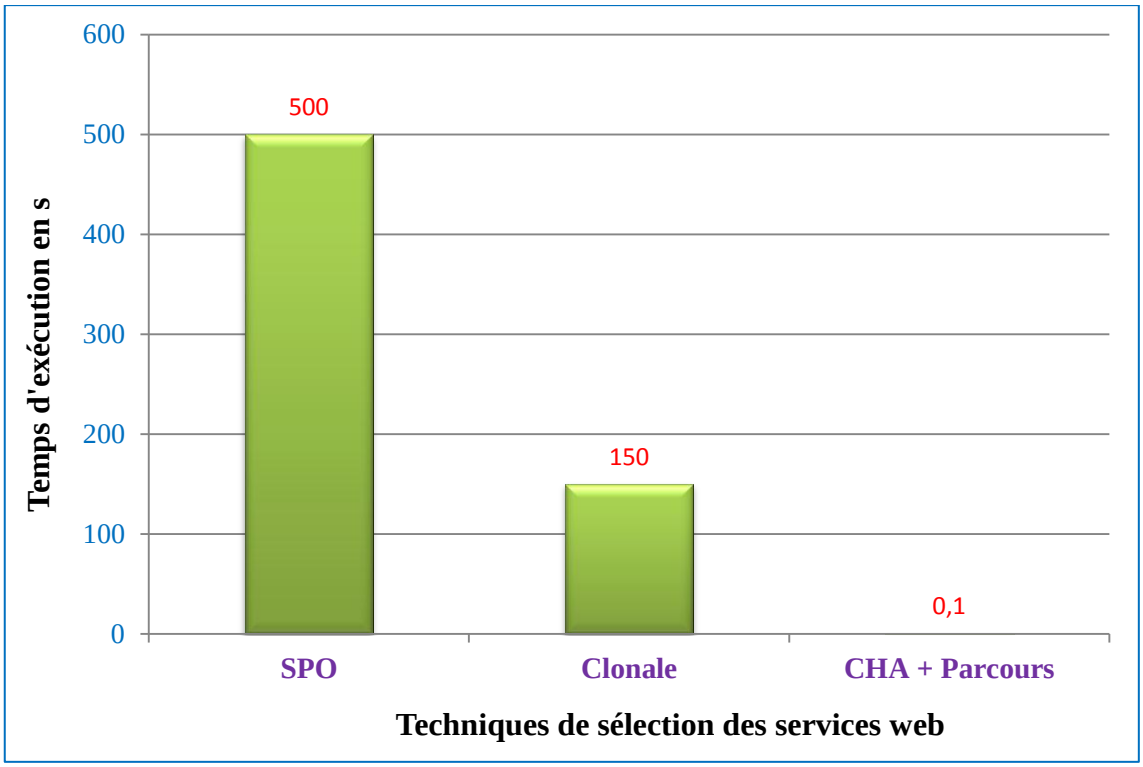

Figure II.13 : Temps d'exécution nécessaire pour les différentes techniques de sélection des services web

- Par opposition aux autres approches, l'optimum global de notre problème est toujours atteint.
- Puisqu'on trouvera les meilleures solutions dans les premiers niveaux, le temps d'exécution sera réduit et on diminuera le temps d'attente de l'utilisateur.
- L'adoption du clustering hiérarchique ascendant donne un résultat stable et facilite la recherche de l'optimum, par opposition les autres techniques telles que Kmeans donnent des résultats instables (plusieurs hiérarchies) et par conséquent la recherche de l'optimum peut être ralentie.

D'après ces résultats on confirme l'efficacité des deux algorithmes choisis (BBS et agglomératif).

IX Conclusion

Dans ce chapitre nous avons proposé une approche pour résoudre le problème de sélection de services.

Notre proposition se base sur plusieurs étapes, en premier lieu nous recherchons les éléments skylines de chaque classe, ensuite nous réalisons le clustering de ces résultats à l'aide des algorithmes hiérarchiques, en fin nous sélectionnons l'optimum en parcourant les hiérarchies niveau par niveau.

Les résultats obtenus sont très acceptables et montrent leur supériorité par rapport à d'autres approches.

Conclusion générale :

Nous avons présenté dans ce travail une approche hybride pour la sélection des services web. Notre approche combine l'optimisation multi objective et mono objective. En premier lieu, nous faisons l'extraction des skylines en utilisant l'algorithme BBS (l'étape multi objective). Ensuite, nous réalisons un groupement hiérarchique des skylines de chaque classe à l'aide de l'algorithme CHA (Clustering Hiérarchique Ascendant). Après, nous parcourons les N hiérarchiques et niveau par niveau afin de retenir les compositions satisfaisant les contraintes globales.

Les solutions du même niveau sont classées à l'aide d'une fonction mono objective.

Les expérimentations menées garantissent l'obtention de l'optimum global tout en restant dans les délais imposés par l'utilisateur.

Comme perspectives à ce travail, nous proposons les améliorations suivantes :

- L'adoption de la programmation par contrainte pendant le parcours des différentes hiérarchiques.
- L'adoption de la programmation dynamique pendant le parcours des différentes hiérarchiques.
- L'adoption d'un autre algorithme de clustering tel que les Carte Topologiques, l'algorithme EM, …
- L'adoption d'un autre algorithme d'extraction des skylines tel que : l'algorithme NN (Nearest Neighbor), l'algorithme DCSkylines (Divide-and-Conquer), …

Références Bibliographiques

[1] A. Martin, Echange de données informatisées (EDI), http://www.commentcamarche.net/contents/315-echange-de-donnees-informatisees-edi, (dernière visite 09/05/2013).

[2] T. Bray, J. Paoli, C.M.Sperberg-McQueen, E. Maler, F. Yergeau et J. Cowan, Extensible Markup Language (XML) 1.1 (Second Edition), W3C Recommandation 16 Août 2006, édité le 29 Septembre 2006, www.w3.org/TR/2006/REC-xml11-20060816/, (dernière visite 09/05/2013).

[3] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach et T. Berners-Lee, Hypertext Transfer Protocol -- HTTP/1.1, Juin 1999, http://www.w3.org/Protocols/rfc2616/rfc2616.html, (dernière visite 09/05/2013).

[4] Intégration d'applications d'entreprise, édité le 06 Mai 2013, http://fr.wikipedia.org/wiki/Int%C3%A9gration_d%27applications_d%27entreprise, (dernière visite 09/05/2013).

[5] C.Lablanche, F. Seine et S.Gastaud, Les Web Services, 2004/2005, http://deptinfo.unice.fr/twiki/pub/Linfo/Organisation%20Rapports/rapport-WebServices.pdf, (dernière visite 09/05/2013).

[6] D.Booth, H. Haas, F. McCabe, E. Newcomer, M.Champion, C.Ferris, et D. Orchard, Web Service Architecture, W3C Working Group Note, Février 2004, http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/, (dernière visite 09/05/2013).

[7] Uniform Resource Identifier, édité le 03/05/2013, http://fr.wikipedia.org/wiki/Uniform_Resource_Identifier, (dernière visite 09/05/2013).

[8] Standards and web services, http://www.ibm.com/developerworks/webservices/standards/#ibm-pcon, (dernière visite (09/05/2013).

[9] Simple Mail Transfer Protocol, édité le 29/04/2013, http://fr.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol, (dernière visite 09/05/2013).

[10] Le protocole FTP (File Transfer Protocol), http://www.commentcamarche.net/contents/519-le-protocole-ftp-file-transfer-protocol, (dernière visite 09/05/2013).

[11] XML-RPC, édité le 03/05/2013, http://php.net/manual/fr/book.xmlrpc.php, (dernière visite 09/05/2013).

[12] Les services Web, édité le 26/10/2011, http://uploads.siteduzero.com/pdf/203276-les-services-web.pdf, (dernière visite 09/05/2013).

[13] A. Dehane et Z. Zohra, Evaluation des techniques de codage d'ontologies sur les performances de la composition de services Web, 2011/2012, http://bibfac.univ-tlemcen.dz/bibfs/opac_css/doc_num.php?explnum_id=268, (dernière visite 09/05/2013).

[14] D. Raggett, A. Le Hors et I. Jacobs, HTML 4.01 Specification, W3C Recommandation 24 Décembre 1999, http://www.w3.org/TR/1999/REC-html401-19991224/, (dernière visite 09/05/2013).

[15] S. Boag, D.Chamberlin, M. F. Fernandez, D. Florescu, J. Robie et J. Siméon, XQuery 1.0: An XML Query Language (Second Edition), W3C Recommendation 14 Décembre 2010, http://www.w3.org/TR/xquery/, (dernière visite 09/05/2013).

[16] A. Berglund, Extensible Stylesheet Language (XSL) Version 1.1, W3C recommandation 05 Décembre 2006, http://www.w3.org/TR/xsl/, (dernière visite 09/05/2013).

[17] J. Clark, XSL Transformations (XSLT) Version 1.0, W3C Recommandation 16 Novembre 1999, http://www.w3.org/TR/xslt, (dernière visite 09/05/2013).

[18] J. Clark et S. DeRose, XML Path Language (XPath) Version 1.0, W3C recommandation 16 Novembre 1999, http://www.w3.org/TR/xpath/, (dernière visite 09/05/2013).

[19] R. De La Rosa-Rosero, Découverte et Sélection de Services Web pour une application Mélusine, soutenu le 15 septembre 2004, http://www-adele.imag.fr/Les.Publications/reports/DEA2004Del.pdf, (dernière visite 09/05/2013).

[20] SOA - Architecture Orientée Service, http://www.commentcamarche.net/contents/1241-soa-architecture-orientee-service, (dernière visite 09/05/2013).

[21] URL, http://www.commentcamarche.net/contents/542-url, (dernière visite 09/05/2013).

[22] S. Pemberton et al. , XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition) A Reformulation of HTML 4 in XML 1.0, W3C Recommandation 26 Janvier 2000, http://www.w3.org/TR/xhtml1/, (dernière visite 09/05/2013).

[23] Schahram Dustdar* et Wolfgang Schreiner, A survey on web services composition, Int. J. Web and Grid Services, Vol. 1, No. 1, 2005, http://www.infosys.tuwien.ac.at/Staff/sd/papers/A%20survey%20on%20web%20services%20composition_Dustdar_Schreiner_inPress.pdf, (dernière visite 10/05/2013).

[24] Business to business, édité le 25/04/2013, http://fr.wikipedia.org/wiki/Business_to_business, (dernière visite 10/05/2013).

[25] Business to consumer, édité le 13/03/2013, http://fr.wikipedia.org/wiki/Business_to_consumer, (dernière visite 10/05/2013).

[26] E. Dellis et B. Seeger, Efficient Computation of Reverse Skyline Queries, publié le 23/09/2007, http://www.vldb.org/conf/2007/papers/research/p291-dellis.pdf, (dernière visite 10/05/2013).

[27] Y. Tao, V. Hristidis, D. Papadias et Y. Papakonstantinou, Branch-and-Bound Processing of Ranked Queries, http://db.ucsd.edu/pubsFileFolder/250.pdf, (dernière visite 10/05/2013).

[28] Z. Kameche, Sélection des Web Services à Base Des Essaimes Prticulaires, présenté le 27/09/2011, http://dspace.univ-tlemcen.dz/bitstream/112/1012/1/Zeyneb-KAMECHE.pdf, (dernière visite 13/05/2013).

[29] T. Kossentini, L.Baccouche et H. H Ben Ghezala, Etude de Problème de Sélection de Services Web en tenant Compte des contraintes temps réel, Vol. 18 - N° 1 Année 2010,

http://www.researchgate.net/publication/200497344_ETUDE_DU_PROBLEME_DE_SELECTION_DE_SERVICES_WEB_EN_TENANT_COMPTE_DES_CONTRAINTES_TEMPS_REEL/file/e57f3a4185549235ac41ecaef67e13a1.pdf?ev=pub_ext_doc_dl&docViewer=true, (dernière visite 13/05/2013).

[30] X. Lu, T. Luo et X. Lin, An Optimal Divide-Conquer Algorithm for 2D Skyline Queries, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.7.1241&rep=rep1&type=pdf, (dernière visite 14/05/2013).

[31] D. François, C. Krier, F. Rossi et M. Verleysen, Estimation de redondance pour le clustering de variables spectrales « Estimating redundancy for spectral variable clustering », http://perso.uclouvain.be/michel.verleysen/papers/agrostat08df.pdf, (dernière visite 14/05/2013).

[32] Q. Yu et A. Bouguettaya, Foundation for Efficient Web Service Selection Springer Science+Business Media, 2010.

Résumé :

La sélection des services web composés est l'une des problématiques majeures de l'architecture SOA. En effet, les utilisateurs ont toujours besoin de sélectionner les meilleurs services qui répondent leurs besoins, tout en vérifiant des exigences globales.

Pour résoudre cette problématique, nous proposons une approche hybride (multi objective et mono objective). Notre proposition commence par l'extraction des éléments skylines des classes abstraites. Ensuite, nous effectuons un CHA (Clustering Hiérarchique Ascendant) sur toutes les classes. Enfin, nous parcourons les N hiérarchies (niveau par niveau) jusqu'à l'obtention de l'optimum global.

Les expérimentations menées confirment l'obtention de l'optimum global dans un temps réduit. En plus, elles montrent la supériorité de notre approche par rapport aux autres méthodes mono objectives.

Mots clés : La sélection de service web, SOA, skyline, CHA.

Abstract:

The selection of web services is one of the major problems of SOA architecture. The users still need to select the best services that satisfy their needs by checking the overall requirements.

To solve this problem, we propose a hybrid approach (multi-objective and mono-objective).

Our proposal begins with the extraction of the skylines of each abstract class. Then we make a Hierarchical Clustering on each class. Finally, we explore the N hierarchies (level by level) until the reach of the global optimum. The experiments carried out confirm the obtaining of the global optimum in a short time. In addition, the results show the superiority of our approach compared to other mono-objective methods.

Keywords: The selection of web services, SOA, skyline, Hierarchical Clustering Ascending.

ملخص

اختيار خدمات الواب هي واحدة من المشاكل الرئيسية لهندسة SOA. المستخدمون بحاجة دائمة لأفضل الخدمات التي تلبي احتياجاتهم، و ذلك بالتحقق من المتطلبات العامة.

لحل هذه المشكلة نقترح اتباع نهج هجين (متعدد الأهداف وأحادي الأهداف). يبدأ اقتراحنا باستخراج عناصر الأفق للفئات المجردة، ثم نقوم بتقسيم هرمي تصاعدي على جميع الفئات. وأخيرا، نجتاز الهرم مستوى بعد مستوى حتى الوصول إلى أفضل الحلول.

التجارب تؤكد الحصول على أفضل الحلول في وقت قصير. وبالإضافة إلى ذلك، فإنها تظهر تفوق نهجنا بالمقارنة مع الطرق الأخرى ذات الهدف الواحد.

الكلمات الرئيسية : خدمات الواب، الأفق، التقسيم الهرمي التصاعدي.