الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة سعيدة د. مولاي الطاهر

كلية التكنولوجيا

قسم: الإعلام الآلي

Mémoire de Master

Spécialité : Réseau Informatique et Système Répartie

Thème

Mise en œuvre d'un environnement de simulation pour l'évaluation des performances d'applications logicielles

Présenté par :

CHAIB Hichem Slimane

SOUAR Aissa

Dirigé par :

Dr. YAHLALI Mbarka

Promotion 2021 - 2022

Remerciements

Avant tout, nous remercions Dieu, le tout puissant, que nous bénissons pour avoir terminé ce travail dans les meilleures conditions possibles.

Nous tenons à remercier sincèrement notre Professeur YAHLALI MEBARKA, en tant qu'Encadreur. Elle s'est toujours montré à notre écoute et très disponible tout au long de la réalisation de ce mémoire, nous voulons que vous sachiez que vos élèves garderont toute leur vie un très bon souvenir de cette année d'étude.

Aussi nous voulons remercier tous nos enseignants de l'université de SAIDA Dr Moulay Tahar pour leurs enseignements dévouements et disponibilités.

Dédicaces

On dédie se travail à nos familles, nos parents, nos frères et sœurs, notre encadreur et nos amis et à tous qui nous ont aidés de loin ou de près tout au long de ce PFE.

Hichem & Aissa

Résumé

L’évaluation de performance est un élément important dans le développement des applications logicielles. Elle vise à calculer les indices de qualité d'un système informatique. Dans ce contexte ce travail s’intéresse au paradigme SOA (Service Oriented Architecture) qui vise à mettre en place un système d'information constitué de services applicatifs indépendants et interconnectés.

L’objectif principal de notre travail est de concevoir et élaborer un environnement d'évaluation de performance des applications logicielles et particulièrement les applications basées sur l’architecture SOA.

Pour cela nous avons effectuée une étude comparative des différentes approches de sélection (approche de sélection locale et approches de sélection globale) des services web. L’outil développé est un service web implémentant les différentes approches de sélection d’un service web.

Abstract

Performance evaluation is an important element in the development of software applications. It aims to calculate the quality indices of a computer system. In this context, this work focuses on the SOA paradigm (Service Oriented Architecture) which aims to set up an information system made up of independent and interconnected application services.

The main objective of our work is to design and develop a performance evaluation environment for software applications and particularly applications based on SOA architecture.

For this we have carried out a comparative study of the different selection approaches (local selection approach and global selection approaches) of web services. The tool developed is a web service implementing the different approaches for selecting a web service.

ملخص

تقييم الأداء عنصر مهم في تطوير تطبيقات البرمجيات. يهدف إلى حساب مؤشرات الجودة لنظام الكمبيوتر. في هذا السياق ، يركز هذا العمل على نموذج SOA (الهندسة المعمارية الموجهة للخدمة) الذي يهدف إلى إنشاء نظام معلومات يتكون من خدمات تطبيقات مستقلة ومترابطة.

الهدف الرئيسي لعملنا هو تصميم وتطوير بيئة تقييم الأداء لتطبيقات البرمجيات وخاصة التطبيقات القائمة على بنية SOA.

لهذا أجرينا دراسة مقارنة لنهج الاختيار المختلفة (نهج الاختيار المحلي وأساليب الاختيار العالمية) لخدمات الويب. الأداة التي تم تطويرها هي خدمة ويب تطبق الأساليب المختلفة لاختيار خدمة الويب.

Table des matières

Liste des figures :

Liste des tables :

Introduction générale :

Aujourd'hui, le développement des applications logicielles est en perpétuelle évolution. Il est soumis à des exigences et contraintes de plus en plus fortes aussi bien non fonctionnelle que fonctionnelle.

Ces applications sont élaborées à partir des besoins fonctionnels des utilisateurs. Leur qualité ne fait pas souvent partie du processus de développement. Ainsi, face à l'immensité de l'offre disponible (des milliers de logiciels peuvent assurer les mêmes services demandés). Il est nécessaire d'avoir un mécanisme de sélection performant qui permet au moins de retrouver les applications les plus pertinentes qui répondent aux besoins fonctionnels demandés avec un certain niveau de qualité (besoins non fonctionnels).

L'évaluation de performance est un élément important dans le développement des applications logicielles. Elle vise à calculer les indices de performances d'un système informatique. Par exemple: débit, temps de réponse, taux de perte, taux d'utilisation.

Dans ce contexte ce travail s'intéresse au paradigme SOA (Service Oriented Architecture) qui vise à mettre en place un système d'information constitué de services applicatifs indépendants et interconnectés. Les services Web sont basés sur le modèle SOA. La technologie des services Web est un moyen rapide de distribution de l'information entre clients, fournisseurs, partenaires commerciaux et leurs différentes plates-formes

Avec le nombre croissant de services mis à disposition sur Internet et dont la fonctionnalité est similaire, la sélection du service le plus adéquat est un des problèmes majeurs qui peuvent faire face aux utilisateurs. Plusieurs méthodes et approches de sélection on été proposée pour résoudre ce problème

Notre objectif est de faire une étude comparative des différentes approches de sélection des services web d'une part et de concevoir et développer un environnement d'évaluation de performance des applications basée sur SOA d'autre part.

Chapitre 1

Les services Web

1. Introduction :

Un service Web est un service logiciel utilisé pour communiquer entre deux appareils sur un réseau. Plus précisément, un service Web est une application logicielle qui permet d'assurer l'interopérabilité entre des applications disparates d'une manière standardisée. Il le fait via HTTP en utilisant des technologies telles que XML, SOAP, WSDL et UDDI.
Ce chapitre présente la théorie et les principes de conception qui sous-tendent la technologie des services Web. Il explique les modèles, les spécifications et les utilisations de cette technologie comme moyen de permettre à des systèmes hétérogènes de travailler ensemble pour accomplir une tâche.

Ce chapitre vise à fournir des informations générales ainsi que des informations sur les domaines de recherche actuels dans le domaine des services Web.

2. Définition du service web :

Les services Web fournissent un moyen standard d'interopérabilité entre différentes applications logicielles, s'exécutant sur une variété de plates-formes et/ou de cadres. Ce document (WSA) est destiné à fournir une définition commune d'un service Web et à définir sa place dans un cadre de services Web plus large pour guider la communauté. Le WSA fournit un modèle conceptuel et un contexte pour comprendre les services Web et les relations entre les composants de ce modèle.

L'architecture n'essaie pas de spécifier la façon dont les services Web sont mis en œuvre et n'impose aucune restriction sur la façon dont les services Web peuvent être combinés. Le WSA décrit à la fois les caractéristiques minimales communes à tous les services Web et un certain nombre de caractéristiques nécessaires à de nombreux services Web, mais pas à tous.

L'architecture des services Web est une architecture d'interopérabilité : elle identifie les éléments globaux du réseau mondial des services Web qui sont nécessaires pour assurer l'interopérabilité entre les services Web. [1]

3. Propriétés des services Web :

Tous les services Web partagent les propriétés suivantes [2] :

- **Autonomes :** Pour lancer l'application client, il suffit l'existence d'un langage de programmation prenant en charge les clients XML et http (ne demande aucun logiciel supplémentaire) . Côté serveur, il suffit un serveur HTTP et un serveur SOAP.

- **Auto-descriptifs :** Les fichiers WSDL (Web Service Description Language) fournissent toutes les informations nécessaires à l'implémentation ou à l'appel des services Web.

- **Publics :** Les services Web peuvent être publiés, découverts et invoqués sur le Web en utilisant des protocoles standards telles que http.

Les services Web sont modulaires. Des services Web simples peuvent être agrégés pour former des services plus complexes, soit en utilisant des techniques de flux de travail, soit en appelant des services Web de couche inférieure à partir d'une implémentation de service Web. Les services Web peuvent être enchaînés pour exécuter des fonctions commerciales de niveau supérieur. Cela raccourcit le temps de développement et permet des implémentations de pointe.

Les services Web sont indépendants de la langue et interopérables
Le client et le serveur peuvent être implémentés dans différents environnements. N'importe quel langage peut être utilisé pour implémenter des clients et des serveurs de services Web.

Les services Web sont par nature ouverts et basés sur des normes
XML et HTTP sont les principaux fondements techniques des services Web. Une grande partie de la technologie des services Web a été construite à l'aide de projets open source.

Par conséquent, l'indépendance et l'interopérabilité des fournisseurs sont des objectifs réalistes.

Les services Web sont faiblement couplés. Un demandeur de service doit connaître l'interface d'un service Web, mais pas les détails de sa mise en œuvre.

Les services Web fournissent un accès programmatique
Cette approche ne fournit aucune interface utilisateur graphique ; il fonctionne au niveau du code.

Les services Web offrent la possibilité d'encapsuler des applications existantes.Les applications existantes peuvent être facilement intégrées dans l'architecture orientée services en implémentant le service Web en tant qu'interface avec l'application.

4. Architecture et fonctionnement de service web :

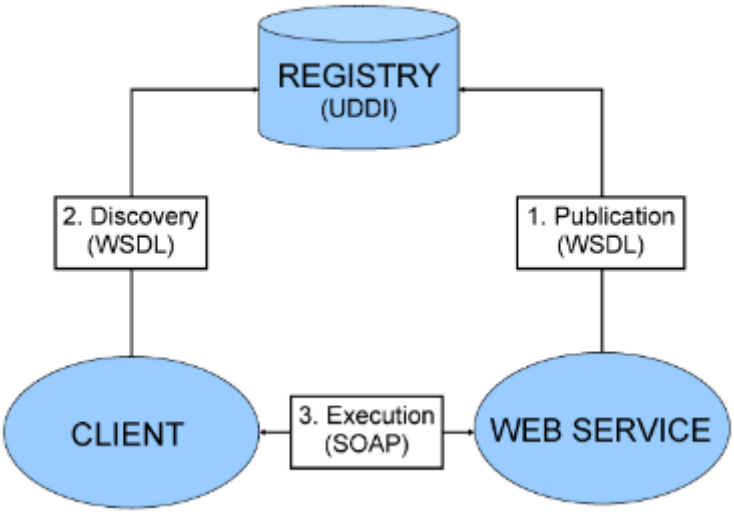

Figure 1.1. Structure fonctionnelle d'un service web [3]

Service provider : (Fournisseur de services)

Il s'agit du fournisseur du service Web. Le prestataire met en œuvre le service et le met à disposition sur Internet.

Service requester : (Demandeur de services)

Il s'agit de tout consommateur du service Web. Le demandeur utilise un service Web existant en ouvrant une connexion réseau et en envoyant une requête XML.

Service register : (Registre des services)

Il s'agit d'un annuaire de services logiquement centralisé. Le registre fournit un emplacement central où les développeurs peuvent publier de nouveaux services ou rechercher des services existants. Il sert donc de centre commercial centralisé pour les entreprises et leurs services. [3]

5. Cycle de vie d'un service web :

Après sa création, le service est déployé sur une plateforme (réseau local ou internet) puis il sera publie à l'aide de WSDL , dans l'annuaire UDDI . Comme le montre la figure suivante le cycle de vie d'un service web se compose de six étape :

- Dans un premier temps le fournisseur de service Web publie ses services web

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

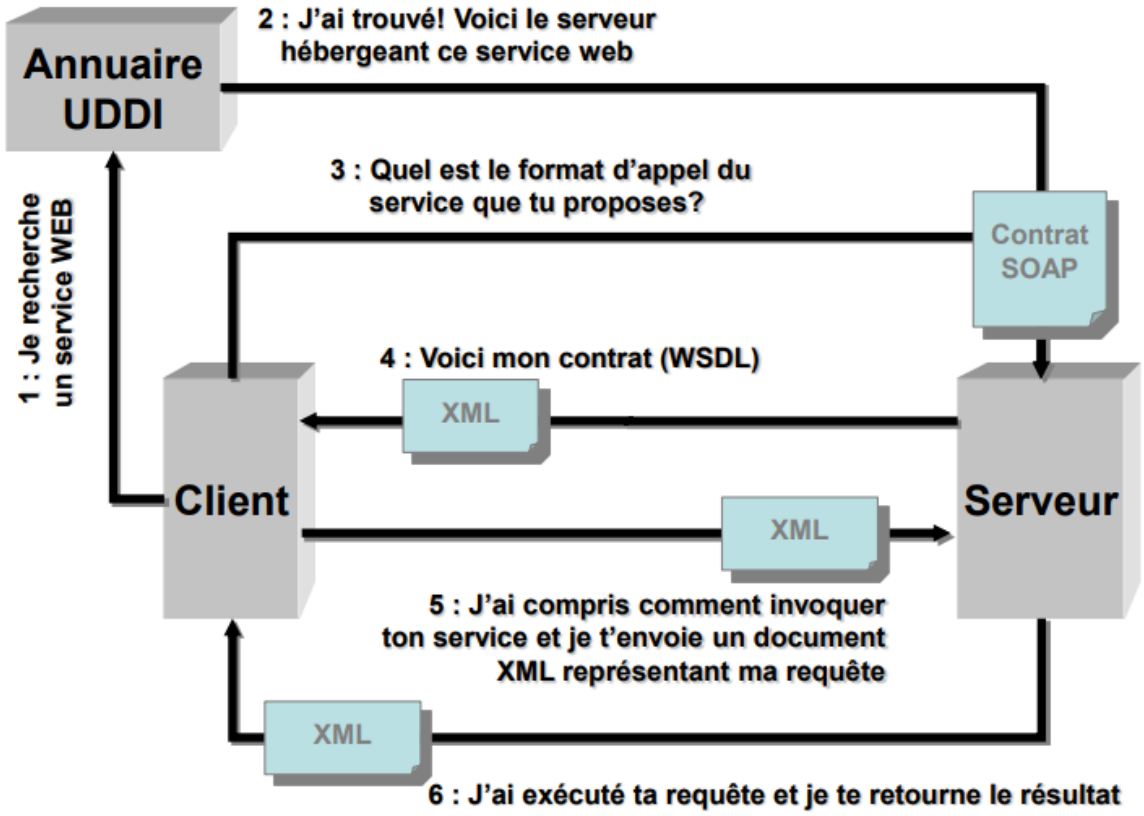

Figure 1.2. Cycle de vie d'un service web [4]

6. Architecture :

Différentes extensions de l'architecture de base ont été proposées dans la littérature. Le groupe architecture du W3C travaille activement à l'élaboration d'une architecture étendue standard. Une architecture étendue est constituée de plusieurs couches se superposant les unes sur les autres, d'où le nom de pile des Web services. La figure l.2 . décrit un exemple d'une telle pile. La pile est constituée de plusieurs couches, chaque couche s'appuyant sur un standard particulier. On retrouve, au-dessus de la couche de transport, les trois couches

formant l'infrastructure de base décrite précédemment. Ces couches s'appuient sur les standards émergents SOAP, WSDL et UDDI

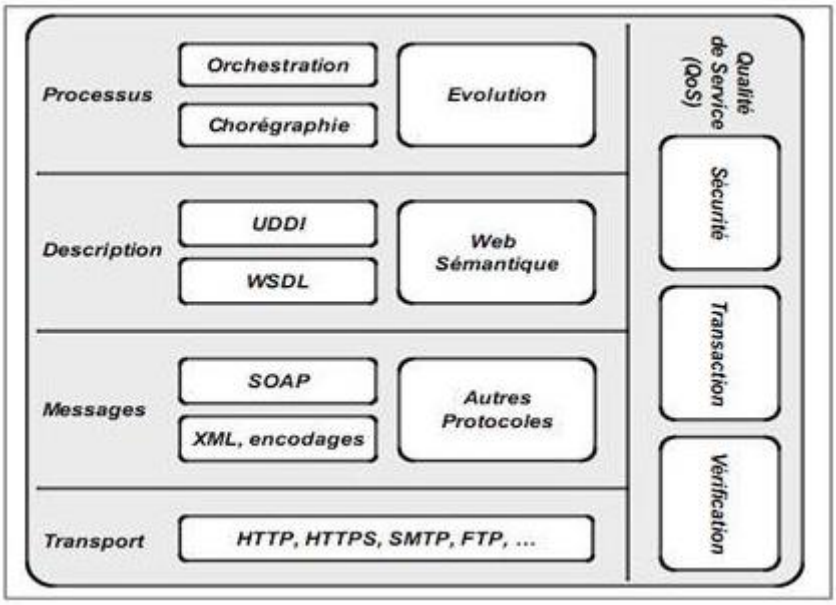

Figure 1.3. Architecture étendue [5]

6.1 SOAP :

SOAP est un protocole basé sur XML pour accéder aux services Web via HTTP. Il a des spécifications qui pourraient être utilisées dans toutes les applications.

SOAP est connu sous le nom de Simple Object Access Protocol, mais plus tard, il a été simplement abrégé en SOAP v1.2. SOAP est un protocole ou, en d'autres termes, une définition de la manière dont les services Web communiquent entre eux ou communiquent avec les applications clientes qui les invoquent.

SOAP a été développé en tant que langage intermédiaire afin que les applications construites sur divers langages de programmation puissent communiquer facilement entre elles et éviter l'effort de développement extrême.

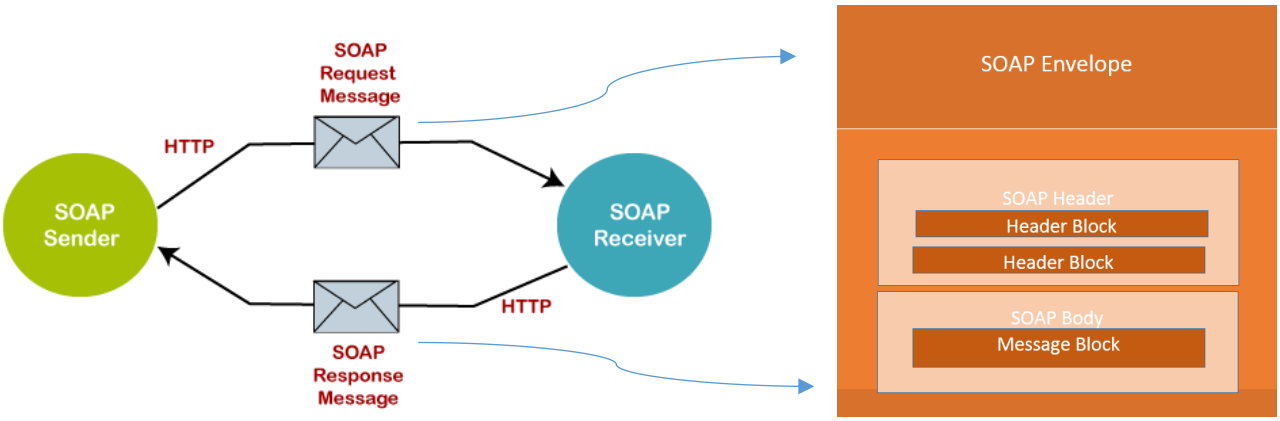

Figure 1.4. Structure d'un message SOAP avec son enveloppe [6]

Le message SOAP n'est rien d'autre qu'un simple document XML qui contient les composants ci-dessous.

SOAP Header *:*
Une enveloppe qui identifie le document XML en tant que message SOAP. Il s'agit de la partie contenante du message SOAP et est utilisé pour encapsuler tous les détails dans le message SOAP. Il s'agit de l'élément racine du message SOAP.

SOAP Body *:*
Un élément d'en-tête qui contient des informations d'en-tête – L'élément d'en-tête peut contenir des informations telles que des informations d'identification d'authentification qui peuvent être utilisées par l'application appelante. Il peut également contenir la définition de types complexes qui pourraient être utilisés dans le message SOAP. Par défaut, le message SOAP peut contenir des paramètres qui peuvent être de types simples tels que des chaînes et des nombres, mais peuvent également être un type d'objet complexe.

6.2 WSDL :

WSDL Web Service Description Langage est une notation XML pour décrire un service Web. Une définition WSDL indique à un client comment composer une demande de service Web et décrit l'interface fournie par le fournisseur de services Web.

Les opérations et les messages sont décrits de manière abstraite, puis liés à un protocole réseau concret et à un format de message pour définir un point de terminaison. Les points de terminaison concrets associés sont combinés en points de terminaison abstraits (services).

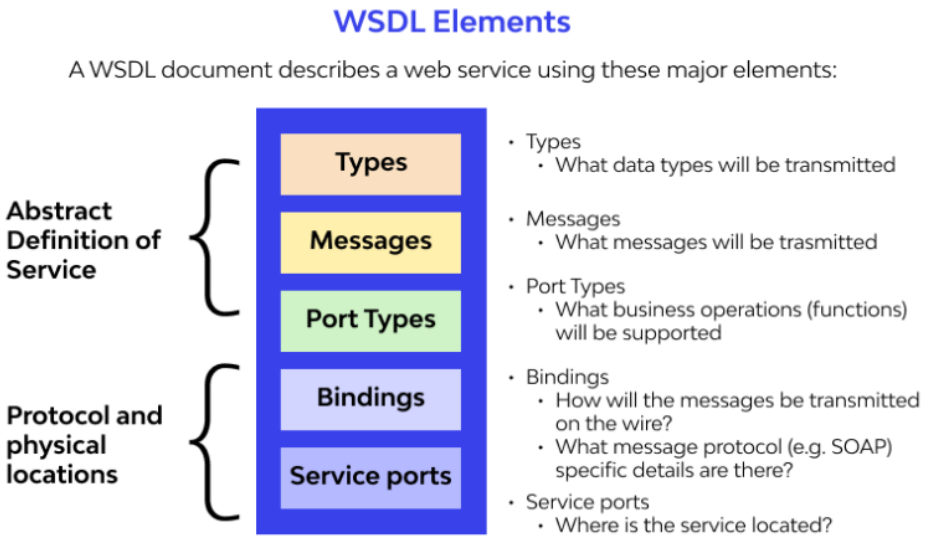

Figure 1.5. La description d'un service web a l'aide de WSDL [7]

- **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.

- **Opération :** Décrit les opération invoquées de manière distante sur le service.

- **Bindings :** Décrit la façon dont un type de port est mis en œuvre 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.

6.3UDDI :

UDDI (Universal Description, Discovery, and Integration) est un registre basé sur XML permettant aux entreprises du monde entier de se répertorier sur Internet. Son objectif ultime est de rationaliser les transactions en ligne en permettant aux entreprises de se retrouver sur le Web et de rendre leurs systèmes interopérables pour le commerce électronique. UDDI est souvent comparé aux pages blanches, jaunes et vertes d'un annuaire téléphonique. Le projet permet aux entreprises de se répertorier par nom, produit, emplacement ou services Web qu'elles proposent.

UDDI a deux parties :

- Registre de toutes les métadonnées d'un service Web, y compris un pointeur vers la description WSDL d'un service.
-
- Un ensemble de définitions de type de port WSDL pour manipuler et rechercher ce registre

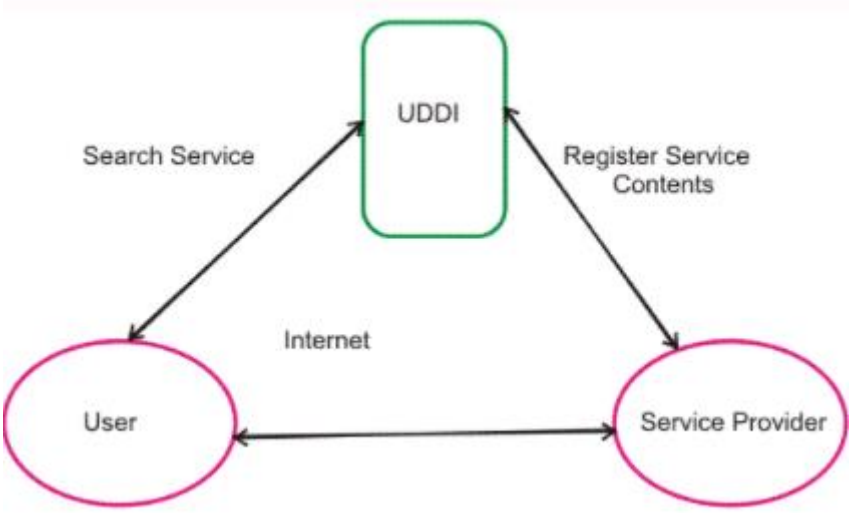

Figure 1.6. Construction d'un registre UDDI d'assignation mondiale [8]

7. Avantage :

Les services Web offrent de nombreux avantages par rapport aux autres types d'architectures informatiques distribuées :

- Interopérabilité : c'est la fonction de démarches des services web entre elle.

C'est l'avantage le plus important des services Web. Les services Web fonctionnent généralement en dehors des réseaux privés, offrant aux développeurs une voie non propriétaire vers leurs solutions. Les services développés sont donc susceptibles d'avoir une durée de vie plus longue, offrant un meilleur retour sur investissement du service développé. Les services Web permettent également aux développeurs d'utiliser leurs langages de programmation préférés. De plus, grâce à l'utilisation de méthodes de communication basées sur des normes, les services Web sont pratiquement indépendants de la plate-forme.

- Convivialité : C'est l'activité ergonomique des prestations du web.

Les services Web permettent d'exposer la logique métier de nombreux systèmes différents sur le Web. Cela donne à vos applications la liberté de choisir les services Web dont elles ont besoin. Au lieu de réinventer la roue pour chaque client, il vous suffit d'inclure une logique métier supplémentaire spécifique à l'application côté client. Cela vous permet de développer des services et/ou du code côté client en utilisant les langages et les outils que vous souhaitez.

- Réutilisabilité : renouvellement multiple et commode d'usage

Les services Web ne fournissent pas un modèle de développement d'applications basé sur des composants, mais la chose la plus proche possible du déploiement sans codage de ces services. Cela facilite la réutilisation des composants de service Web, le cas échéant, dans d'autres services. Il facilite également le déploiement de code hérité en tant que service Web.

- Deployabilité :

Les services Web sont déployés sur des technologies Internet standard. Cela permet de déployer des services Web même au-delà du pare-feu vers des serveurs fonctionnant sur Internet à l'autre bout du monde. De plus, grâce à l'utilisation de normes communautaires éprouvées, la sécurité sous-jacente (telle que SSL) est déjà intégrée.

8. les inconvénients :

Aucune technologie n'est parfaite. Bien que les services Web fassent un excellent travail pour résoudre certains problèmes, ils apportent leurs propres problèmes. Certains de ces

écueils sont inhérents aux fondations technologiques sur lesquelles reposent les services Web, et d'autres sont basés sur les spécifications elles-mêmes. Il est important de savoir quels sont ces problèmes afin de pouvoir planifier et construire autour d'eux.

Certains des plus gros problèmes sont :

- Disponibilité :

Tous ceux qui utilisent Internet savent qu'aucun site n'est disponible à 100 %. Il s'ensuit que les services Web, qui utilisent la même infrastructure que les sites Web, ne seront pas non plus disponibles à 100 %. Même si le serveur est opérationnel, votre FAI peut ne pas l'être, ou le FAI hébergeant l'autre côté de la transaction peut ne pas l'être non plus. Si vous avez besoin de 100% de disponibilité, faites autre chose. En raison de cette situation, il est souvent nécessaire de créer des mécanismes qui réessayeront la transaction ou échoueront normalement lorsque cela se produit. Certains des protocoles les plus récents pris en charge par les services Web (JMS, par exemple) gèrent cela automatiquement, mais la majorité construite sur HTTP ne le fera pas.

- Besoins correspondants :

Chaque fois que vous créez un service général qui s'occupera d'une variété de clients, vous rencontrerez des besoins spécifiques. Certains clients peuvent avoir besoin d'une petite fonctionnalité supplémentaire dont personne d'autre n'a besoin. Les services Web sont envisagés comme une technologie "taille unique pour de nombreux clients". Si votre entreprise ne peut pas s'adapter à ce modèle, vous devriez envisager d'autres solutions.

- Interfaces immuables :

Si vous investissez dans la création d'un service Web pour vos clients, vous devez éviter de modifier les méthodes que vous fournissez et les paramètres attendus par vos clients. Vous pouvez créer de nouvelles méthodes et les ajouter au service, mais si vous modifiez celles qui existent déjà, les programmes de vos clients seront interrompus. C'est facile à faire jusqu'à ce que vous trouviez que l'une de vos méthodes existantes renvoie de mauvaises réponses et ne peut pas être réparée car l'approche est fondamentalement défectueuse. Les premières versions de Java contenaient des méthodes de calcul des différences de date et d'heure qui ne pouvaient pas fonctionner correctement. Ils n'avaient d'autre choix que de jeter le premier ensemble de routines et d'en écrire de nouvelles. Cela a provoqué l'arrêt des programmes, mais il n'y avait pas d'autre choix car les programmes recevaient de mauvaises réponses des anciennes routines. Bien que ce type de problème se produise dans tous les systèmes, il est particulièrement vrai dans les services Web. Vous ne savez peut-être pas qui utilise votre service et, par conséquent, vous n'avez aucun moyen d'informer ces utilisateurs du changement. Dans la plupart des autres systèmes, en raison du couplage étroit, il y a

généralement plus de coordination verbale ou écrite entre les producteurs d'un service et les consommateurs de celui-ci.

- Exécution garantie :

L'idée derrière le fait d'avoir un programme informatique au lieu de faire un travail à la main est que le programme peut fonctionner sans surveillance. HTTP n'est pas un protocole fiable dans la mesure où il ne garantit pas la livraison ou une réponse. Si vous avez besoin de ce type de garantie, écrivez vous-même le code pour réessayer les demandes ou faites en sorte d'envoyer vos demandes par l'intermédiaire d'un intermédiaire qui effectuera ces tentatives pour vous. Encore une fois, les nouvelles versions de la spécification permettent d'utiliser des protocoles tels que JMS pour résoudre ce problème, mais la majorité des services utilisent toujours HTTP, ce qui n'est pas le cas.

9.Conclusion :

Les services Web sont l'un des éléments clés du Web dit programmable. Ce sont des éléments logiciels extrêmement polyvalents qui ont vraiment le potentiel d'ouvrir une nouvelle ère dans le logiciel : l'ère de l'interopérabilité. Les services Web peuvent être utilisés efficacement pour participer et mettre en place des transactions interentreprises (B2B). Ils sont excellents pour exposer les fonctionnalités logicielles aux clients et intégrer des plates-formes hétérogènes.

Chapitre 2

L’évaluation de performance et la qualité logicielle

1. Introduction

Après la multiplication des services web, l'´évaluation de la performance de ce dernier devient très utile pour ces client qui veulent bénéficier un service de qualité. Il est donc conseillé d'étudier le comportement de ces systèmes, avant de les déployer au sol, a fin de comprendre et de résoudre les problèmes qui pourraient affecter ces systèmes.
La qualité des logiciels consiste à évaluer un logiciel en fonction de divers critères. l y a de nombreux critères d'évaluation d'un logiciel, par exemple, l'exactitude des résultats, la portabilité sur différentes plates-formes ou la facilité de modifier le logiciel [9].
Dans ce contexte, plusieurs méthodes d'évaluation de la qualité du logiciel sont suggérées. Dans ce chapitre, nous présentons les différents concepts relatifs à la qualité, les différentes terminologies, l'importance de la qualité logicielle et les différents types et niveaux de qualité, puis nous fournissons des définitions de facteurs, de critères et de paramètres de qualité.

2. Définition de L'évaluation de performance

L'évaluation de la performance d'un système est un domaine qui gagne en importance, Cela devient inconcevable de construire un système sans avoir réalisé au préalable une analyse de performance pour construire un système adapté conformément aux objectifs des spécifications [10].
L'évaluation de la performance met l'accent sur le calcul des paramètres de performance du système (indices) [11]. Les paramètres de performance que nous voulons obtenir sont des ordres différents, selon le système pris en considération [12].

L'évaluation de performance se fait à deux niveaux :

- **En conception :** Le système n'existe pas encore, l'objectif est d'évaluer le système avant sa construction.
- **En exploitation:** A ce niveau, on évalue souvent les performances d'un système à des fins de modification (extension) ou de test au-delà de son point de fonctionnement normal. L'objectif est d'améliorer le système.

3. Les objectifs de performance:

Souvent, une exigence de performance exprimée par un propriétaire de projet par exemple le propriétaire spécifier le besoin suivant : « ***Le logiciel doit être rapide »*** . Cet exemple est caricatural, mais il met l'accent sur la nécessité de l'objectivité qui peut être attendus de ce type d'exigence au moment de la vérification. Pour l'équipe de développement, il est évident que les exigences de performance exprimées dans un style vague et subjectif . Cette première constatation nous conduit à proposer et demander des indicateurs strictement définis et mesurables. Par la suite, les cibles de performance devraient être établies en fonction de ces indicateurs. Bien entendu, ces indicateurs doivent être importants pour l'autorité contractante. La définition d'indicateurs qui quantifient le confort des utilisateurs et l'utilisation des ressources est important [13]:

- Le temps de réponse : le temps requis pour effectuer une opération au moyen de l'application. (par ex. le temps d'affichage d'une page surun projet web). En lieu et place d'un indicateur brut, il est plus pratique d'utiliser des classes de qualité : sous-optimal, optimal, sur-optimal. Pour un projet web moyen, nous pouvons affirmer : plus de 4 secondes

- c'est sous-optimal, entre 3 et 4 secondes c'est optimal, moins de 3 secondes c'est sur-optimal.

- Le débit : Il s'agit de la mesure du chargement instantané pris en charge par le logiciel et son infrastructure. nous pouvons suivre le nombre de requêtes que l'application peut servir par seconde.

- L'utilisation des ressources : Il s'agit du niveau d'usage des sous- systèmes logiciels.(files d'attente, base de données, caches, etc.).

- L'ensemble des objectifs définis et leur méthode de mesure sont regroupés dans undocument, le plan de performance, accessible et connu de l'ensemble de l'équipe de conception.

4. Les étapes d'évaluation de performances

La performance d'un système peut être évaluée en troisétapes [14] :

1. **Etape1** : Comprendre de quelle façon le système fonctionne.
2. **Etape2**: Élaborer un modèle plus fidèle aux caractéristiques et fonctionnements du système.
3. **Etape3 :** Evaluer la performance du dispositif en fonction du formalisme du modèle.

Il est possible de schématiser l'évaluation du rendement d'un système de la façonsuivante [15]:

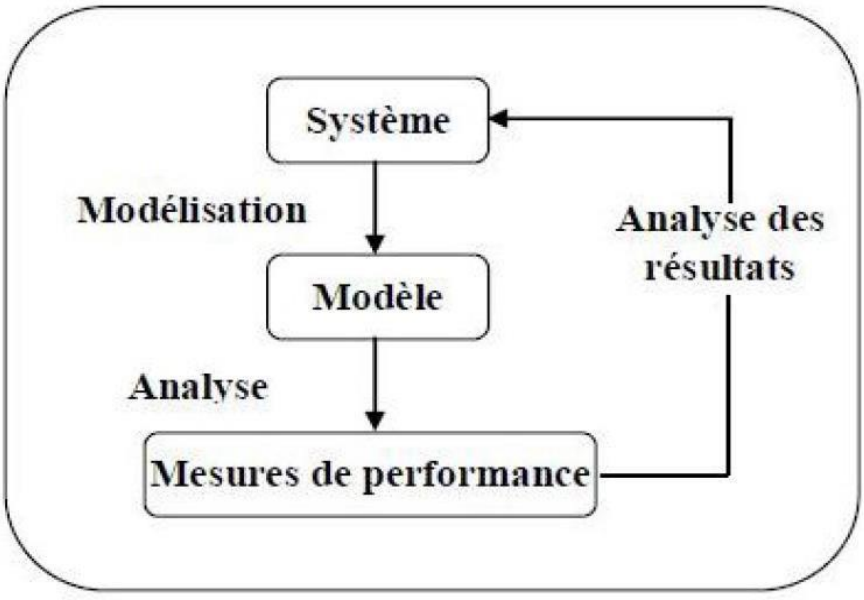

Figure 2.1. : Processus d'évaluation de performance [18].

5. Techniques d'évaluation de performance

Les Techniques d'évaluation de performance sont classé en deux catégories : l'acquisition de mesures directes sur le système (évaluation en exploitation) , la modélisation (évaluation en conception) [16] .

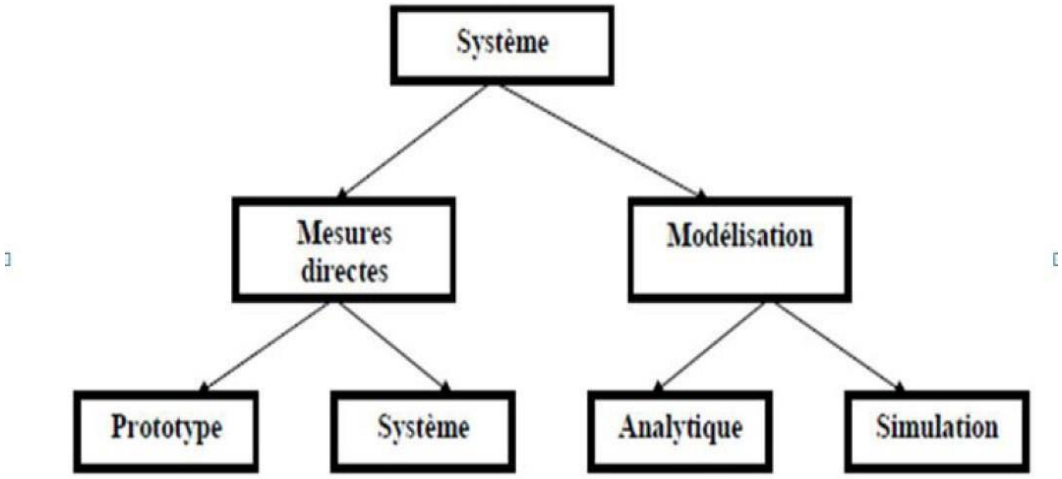

Figure 2.2 : Technique d'évaluation de performance
[23]

5.1. Mesures directes

Les techniques de mesure ou l'analyse opérationnelle des systèmes méthodes très utile car elle offre une image réel , l'état d'un système réel prenant en considération toutes les caractéristiques de ces Derniers . Mais elle comporte plusieurs inconvénients. En fait, les caractéristiques du système sont extrêmement variables et imprévisibles et, en conséquence, les données obtenues à un instant donné sur le système ne permettent pas toujours de prédire le comportement du système dans d'autres conditions. Cette première technique ne peut pas être employée là ou` une partie du système n'existe pas encore.

5.2. Modélisation

La modélisation sert à décrire simplement les éléments du système que nous voulons analyser [20]. Le processus de modélisation consiste à décomposer le système à étudier en plusieurs taches, afin de le simplifier et de l'examiner de plus près ,Cette représentation symbolique dépend aussi d'outils théoriques et mathématiques[24]. La modélisation analytique est un formalisme mathématique visant à créer un modèle qui traduit le comportement et intègre les paramètres du véritable système. Il existe de nombreuses méthodes de modélisation, telles que: Les chaines de Markov, les files d'attente, les réseaux de files d'attente qui sont orientés évaluation de performance et les réseaux de Pétri.
Les modèles Markov sont souvent utilisés pour l'analyse des performances des systèmes limitée parce que cette technique exige l'identité de tous les états du système et la connaissance des vélocités ou des probabilités de changement d'état.
Les réseaux de files d'attente et les réseaux de Petri sont les méthodes les plus utilisés dans un but d'évaluation de performance , Leur avantage réside dans le fait qu'ils permettent une construction plus naturelle du modèle basée sur la description des composants du système et de leur comportement. Plutôt que d'employer une forme mathématique, les modèles sont ensuite décrits graphiquement, plus proche des mécanismes employés par le système réel. Les réseaux de files d'attente et les réseaux de Pétri peuvent être analysées grâce à l'étude de leur modèle markovien.

- **Les méthodes analytiques :** Permettre de dériver les paramètres de performance en

fonction des paramètres de temps du modèle. Ces méthodes sont très efficaces et bien plus rapides que la simulation, mais la solution existe seulement pour une classe très limitée de systèmes. Dans les méthodes analytiques, L'évaluation des performances est ensuite réalisée, en déterminant les équations analytiques, qui sont suivie par une résolution numérique [17].

- **Simulation :** La simulation permet d'imiter le comportement d'un système par le traitement de modèles généralement plus proches de la réalité, et elle est souvent utilisée pour valider les modèles de ce dernier.

La simulation est répétée plusieurs fois (en général plusieurs millions de fois), de sorte que les variables aléatoires représentent la répartition des probabilités et son coefficient de variation. De plus, la simulation présente l'avantage d'être insensible aux dimensions de l'espace étatique. Mais, il est difficile d'utiliser directement la simulation et sa complexité temporelle augmente avec les informations investies dans le modèle et le degré d'exactitude.

Comme le montre la figure suivante , il existe plusieurs types de simulation :

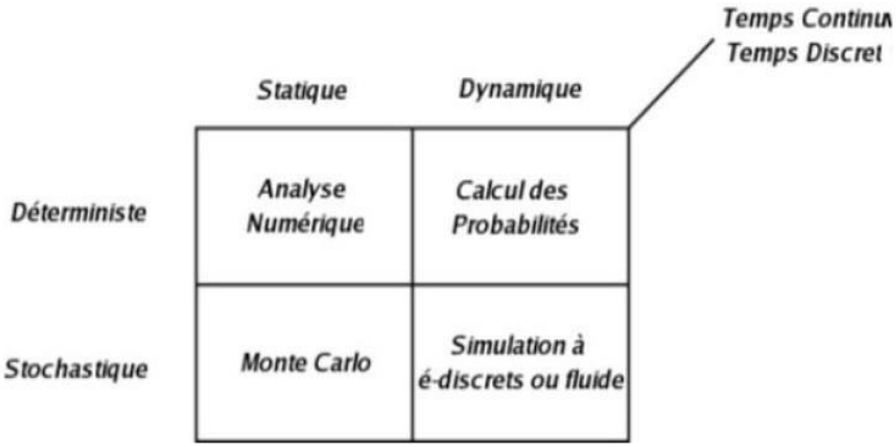

Figure 2.3 : Types de simulations [19].

Simulations statiques : La méthode ne s'applique que si le temps n'influence pas, utilise des impressions aléatoires et souvent uniformes. Des valeurs sont arbitrairement attribuées aux variables et des évaluations et des mesures sont effectuées dans ces conditions [19].
Simulations dynamiques : Un système qui évolue avec le temps. Les variables de système peuvent changer les valeurs lorsque certains événements se produisent.il y a deux sortes de simulation dynamique : avec horloge à incrémentation fixe, et variable.
Simulations à événements discrets : les variables ne se changent qu'au temps d'un événement , par contre restent constantes entre les temps des événements .La simulation d'événement discret permet de modéliser un système réel dont le comportement peut changer selon l'apparence des événements dans le temps. [20].

6. Passerelle entre L'évaluation de performance et la qualité

L'évaluation de performance contient de partie : L'évaluation de la performance fonctionnelle et non fonctionnelle.

- **L'évaluation de la performance fonctionnelle :** Un software est créée pour répondre tout d'abord , aux besoins fonctionnels des entreprises, S'il fournit les résultats voulus et fait la tache qu'il est construit pour, on dit que le logiciel présent les exigences fonctionnelles.

- **L'évaluation de la performance non-fonctionnelle :** Ces exigences définissaient la qualité pour travailler dans de bonnes conditions (Ex. Temps de réponse, Sécurisé, Portabilité, cout de logiciel...) .

7.La qualité logiciel

Aujourd'hui, l'information est une ressource stratégique pour la plupart des entreprises, dont un grand nombre d'activités sont fondées sur l'exploitation d'applications informatiques. Pour ces sociétés, la fiabilité de leurs systèmes informatiques et la qualité des logiciels utilisés sont cruciales. Parallèlement, on constate que [21] :

- la part des logiciels est désormais prédominante dans le coût global d'un système informatique.
- la demande de nouvelles applications de plus en plus complexes continuede s'accroitre.
- les utilisateurs exigent toujours davantage en termes de fiabilité etde sécurité.

La qualité des logiciels consiste à évaluer un logiciel en fonction de divers critères. Il y a de nombreux critères d'évaluation d'un logiciel, par exemple, l'exactitude des résultats, la portabilité sur différentes plates-formes ou la facilité de modifier le logiciel [20].

7.1 Définition

La qualité du logiciel est définie en fonction de sa capacité à satisfaire les besoins des utilisateurs. Elle est définie par l'ANSI comme "l'ensemble des attributs et caractéristiques d'un produit ou d'un service qui portent sur sa capacité à satisfaire des besoins donnés" [21]. Selon le standard ISO 9000 " la qualité d'un logiciel est l'ensemble des caractéristiques intrinsèques qui lui confère l'aptitude
à satisfaire les besoins et attentes exprimés ou implicites des clients et autres parties intéressées."[22]

7.2. Types et niveaux de qualité

Généralement, il y'a deux visions pour la qualité [17]:

- Vision du développeur (organisme).

- Vision du client.

Figure 2.4. : Différent type de qualité [17].

7.3. Concepts de base :

- **Qualité de service QOS (Quality of Service) :** Les services Web sont des systèmes logiciels auxquels on répond de façon générale à des besoins fonctionnels et non fonctionnels, les exigences non fonctionnelles des services Web sont le plus souvent exprimées par la qualité de service. Un service est le résultat des activités menées à l'interface entre le fournisseur et le client et des activités internes des fournisseurs afin de répondre aux besoins des clients. Il s'agit de la série de caractéristiques fournies et accessibles aux clients [23].

- **Mesure** : Un processus de description des entités du monde réel selon des critères prédéfinis, Les attributs des entités se voient attribuer des numéros ou des symboles (ex. La taille d'un logiciel –> nombre de ligne de code). Une définition de la notion de mesure est proposée par Fenton [23]
  :" ¡ Formally , we define measurement as a mapping from the empirical world to the formal, relational world. Consequently, a measure is the number or symbol assigned to an entity by this mapping in order to characterize an attribute".

- **Facteur de qualité :** est une caractéristique du logiciel, du procédé ou du service qui contribue à la qualité du logiciel selon la perception de l'utilisateur.

- **Critère de qualité :** est un attribut du logiciel grâce auquel il est possible d'obtenir un facteur. C'est aussi une caractéristique du logiciel qui permet au développeur d'agir. (Par exemple, sa simplicité).

- **Métrique de qualité :** désigne la mesure d'une propriété d'un critère (ex : la

taille d'un module pour le critère de simplicité.

7.4. Modèles de qualité :

Le modèle de qualité modèle détermine la qualité du logiciel en fonction de la qualité du pro- duit, du processus et du service fourni. Ce modèle peut être présenté comme une structure arborescente [21]. Une définition plus concise d'un modèle de qualité est présentée par Donald [17], « *Un modèle de qualité est défini comme étant un modèle qui a pour objectif de décrire, d'évaluer et/ou de prédire la qualité logicielle* »

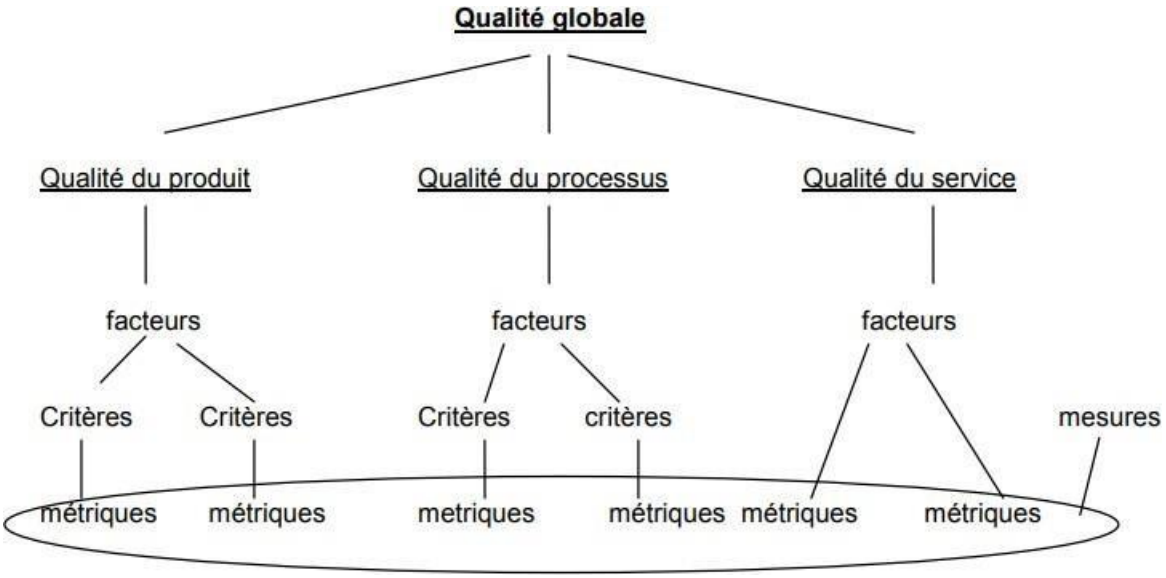

Figure 2.5: Modèles de qualité [21].

7.5. Modèle de qualité interne et externe :

La figure 2.6 illustre le modèle de qualité interne et externe de la norme ISO 9126-1. Ce modèle distingue six caractéristiques qualitatives logicielles (capacité fonctionnelle, Fiabilité, facilité d'utilisation, rendement, maintenabilité et portabilité), qui se subdivisentchacune dans les fonctionnalités secondaires. Ceux-ci peuvent se mesurer, en fonction du contexte, par les paramètres externes et internes.
La sécurité est l'une des mesures qu'on peut également trouver parmi les mesures de fonctionnement. La sécurité des systèmes d'information comprend l'intégrité, la confidentialité et la protection. On peut se servir de la norme ISO 15026 pour définir lesniveaux d'intégrité du système et du logiciel.

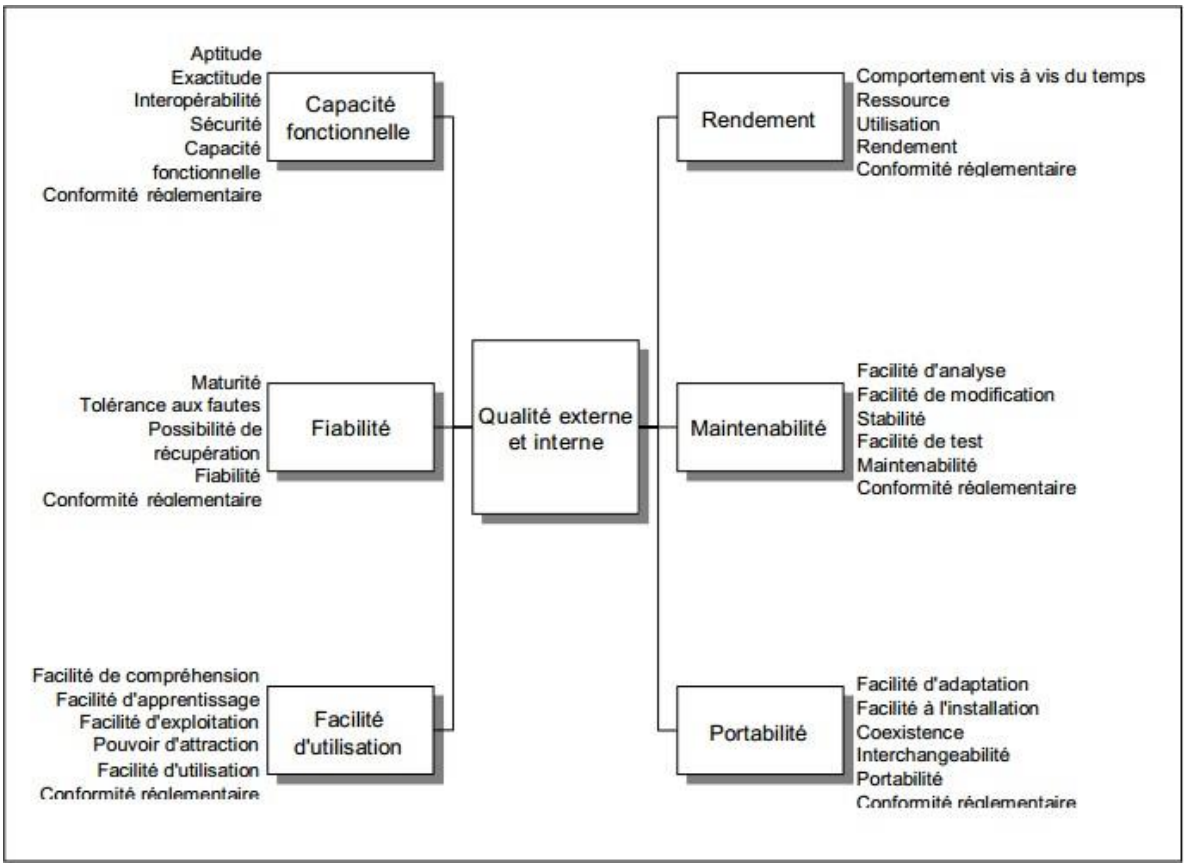

Figure 2.6: Modèles de qualité externe et interne [21].

Le groupe W3C propose un guide [45] pour définir les Qos et leurs métriques. Le guide proposé regroupe les Qos en 4 catégories : les attributs de performance, les attributs qui assurent la sureté de fonctionnement (Dependability), les attributs de sécurité et les attributs spécifiques au domaine d'application. Le tableau suivant illustre cette catégorisation :

Catégorie de QoS	Attributs de QoS
Performance	Temps d'exécution, débit, temps de réponse, ...
Sureté de fonctionnement	Disponibilité, accessibilité, fiabilité, ...
Sécurité	Authentification, non-répudiation, confidentialité, intégrité, ...
Attributs spécifiques aux domaines d'application	Prix, mode de payement, taux de pénalité, ...

Figure 2.7 : Catégorisation des attributs de Qos

7.6. Des outils pour mesurer la qualité :

Un modèle de qualité se compose d'un certain nombre de règles et de principes, réparties en différentes catégories. La mise en œuvre d'un modèle de qualité im- plique de mesurer ces règles à cette fin, le modèle recueille une certaine quantité d'information du projet, y compris des mesures du code.

- **Les métriques de code :**

Il existe un nombre important de métriques et certains d'entre eux ne sont pas calculés de la même façon selon l'outil utilisé. C'est la raison pour laquelle il est nécessaire de déterminer quelles métriques seront utiles et comment ils devraient être calculés avec précision [24].
Cependant, les mesures peuvent être divisées en deux catégories générales :

1. **Les métriques primitives :** Ce sont des métriques qui mesurent les Propriétés élémentaires du code source comme le nombre de lignes de code, la complexité cyclomatique [24].
2. **Les métriques de design :** Ils déterminent si le code source est conformeaux principes de conception établis précédemment [24].

8.Conclusion

L'évaluation des performances et la qualité des logiciels sont deux concepts très prisés dans ledomaine de l'ingénierie logicielle. La qualité des logiciels rassure un logiciel performant, et ce dernier est très unique parmi tous les logiciels. Dans ce chapitre nous avons présenté les concepts les plus importants liés à la qualité et à l'évaluation de performance.

Chapitre 3

Les approches de sélection

1. Introduction :

Pour un besoin fonctionnel donné, nous pouvons trouver plusieurs services qui assurent les mêmes tâches. Il est donc possible d'avoir plusieurs services équivalents. Cette contribution a pour but de répondre à la question suivante : **« Quel est le service le plus performant entre plusieurs services équivalents? »**

Le caractère dynamique des services Web augmente le besoin des applications en dynamisme. Les méthodes existantes sont bâties sur des protocoles de découverte de services web, des descriptions de services et de contrôle de services. Toutefois, la sélection des services Web selon des critères liés au contexte n'est pas complètement traitée par ces protocoles.

La sélection de services Web consiste à sélectionner le service le plus adéquat selon un besoin. La sélection se base principalement sur la description de service qui est une interface de service et qui identifie un ensemble d'opérations et un ensemble de propriétés pour qualifier le service.

Les mécanismes nécessaires pour la sélection de services Web sont complexes en raison de la richesse et de la dynamique du contexte.

Dans ce chapitre nous définissons le principe de la sélection de service, en présentant une liste des approches de sélection des services web qui existent actuellement.

2. Les approches de sélection :

Plusieurs travaux ont été proposés pour résoudre le problème de sélection des Web de services. Dans la littérature, on trouve plusieurs classifications des approches : selon la fonction à optimiser mono-objectif, multi-objectif ou hybride [25] , Selon la stratégie de sélection utilisée soit locale ou globale [26] .

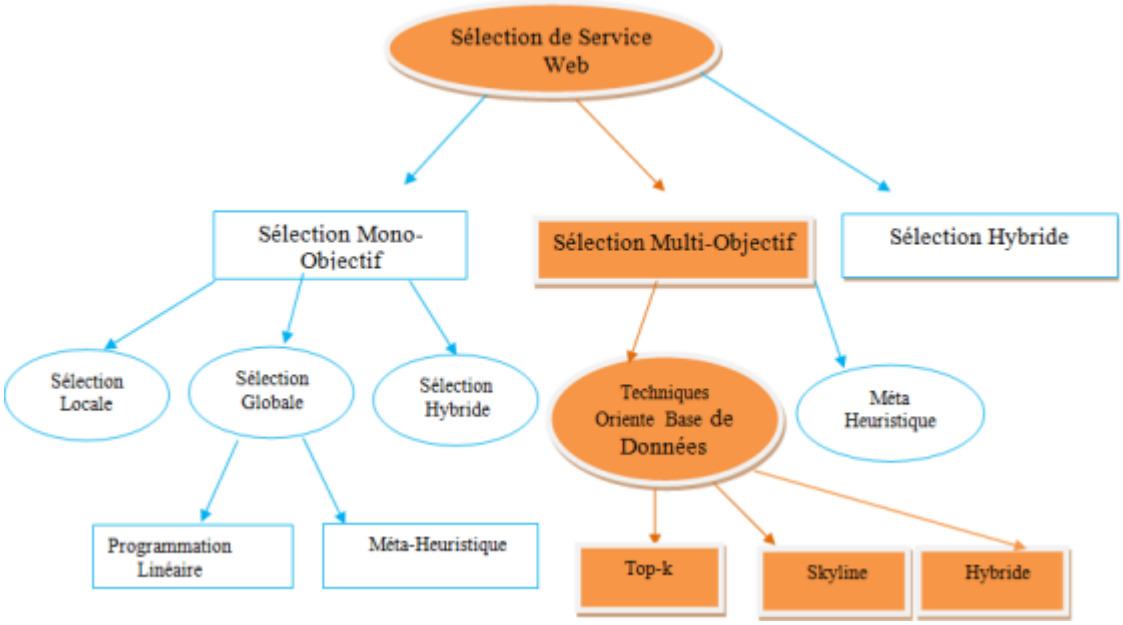

Figure 3.1 : Approches de sélection de services web à base Qos. [44]

2.1. La sélection locale :

Elle a pour objectif de choisir le meilleur Web service pour chaque tâche individuelle [3]. Dans cette approche [4] la sélection du service Web qui exécutera une tâche donnée d'une spécification de service composite se fait au dernier moment possible et sans tenir compte des autres tâches impliquées dans le service composite. Lorsqu'une tâche doit réellement être exécutée, le système collecte des informations sur la qualité de service de chacun des services Web qui peuvent exécuter cette tâche. Après avoir collecté ces informations de QoS, un vecteur de qualité est calculé pour chacun des services Web candidats, et sur la base de ces vecteurs de qualité, le système sélectionne l'un des services Web candidats.

2.1.1 Les méthodes de sélection locale :

- **SAW (Simple Additive Weitling) :**

La somme Pondérée (SAW) [5,6] est une méthode d'agrégation courante plus utilisé dans le cas de la prise de décision. La méthode est basée sur la moyenne pondérée. Un score d'évaluation est calculé pour chaque alternative en multipliant la valeur mise à l'échelle donnée à l'alternative de cet attribut par les poids d'importance relative directement attribués par le décideur, puis en additionnant les produits pour tous les critères. Cette méthode [32] est plus efficace que les autres méthodes car le temps nécessaire au calcul est plus court. L'avantage de cette méthode est que les degrés (les poids) de préférence entre les différentes alternatives et les priorités préférentielles indiquant le classement préféré des alternatives pour chaque décideur.

Algorithme de la somme pondérée

```
1 : procedure calculQoScore (Services,Requete,Poids)
    2 : Normalisation (Requete)
    3 : pour tout Service dans Services
        4 : score = Σ_{i=1}^{Requete.length} Service.atti * Poids.atti ;
        5 : Service.QoScore = score ;
    6 : fin pour;
  7 : fin procedure ;
```

Figure 3.2 : Algorithme de la somme pondérée [41]

- **MAGIQ :**

La technique MAGIQ [33] utilise le concept ROC (Rank Order Centroids) qui arrange les critères selon leur priorité par rapport à l'objectif de décision et les alternatives sont classées en fonction de leur degré de satisfaction à chaque critère. Ces ordres de classement sont ensuite convertis en échelles numériques à l'aide de Centroids d'ordre de classement, en utilisant cette formule [9].

$$W(A_k) = \frac{1}{n}\sum_{i=k}^{n}\frac{1}{i}$$

L'avantage de cette méthode est que les poids sont générés d'une façon automatique en classant les préférences par priorité en utilisant la formule au-dessus.

Algorithme MAGIQ

1 : **procedure** ComputeMagiqScore (Services, Requête)
2 : poids = ComputeWeights (| Requête|) ;
3 : **pour** tout Service **dans** Services;
4 : score = $\sum_{i=1}^{|Requête|}$ poids . att_i * Service. att_i ;
5 : Service.QoSscore = score ;
6 : **fin pour**
7 : **fin procedure**

1 : **function** ComputeWeights (|Attributs|)
2 : **pour** i = 1 **à** | Attributs | **faire**
3 : w[k] = $\sum_{i=k}^{|Attributs|} \frac{1}{i}$;
4 : w[k] = $\frac{w[k]}{|Attributs|}$;
5 : **fin pour**
6 : retourner w ;
7 : **fin function**

Figure 3.3 : Algorithme MAGIQ [42]

- **La Distance Euclidienne :**

Algorithme Distance euclidienne

1 : **procedure** ComputeQosScore (Services, Requête)
2 : normalise (Requête);
3 : **pour** tout Service **dans** Services;
4 : distance = 0 ;
5 : **pour** i **de** 1 **à** |Requête| **faire**
4 : distance = distance + (Requête.att_i – Service.att_i) ^{2};
5 : **fin pour**
6 : distance = $\sqrt{distance}$;
7 : Service.distance = distance;
7 : **fin pour**
8 : **fin procedure**

Figure 3.4 : Algorithme Distance euclidienne [35]

Dans [10], les auteurs proposent une méthode basée sur la mesure d'une distance entre les préférences de l'utilisateur en termes de QoS et les annonces des fournisseurs de services. L'algorithme détermine quel Web service wsi d'un ensemble de services WS = {ws1, ws2, wsn} qui sera choisi pour exécuter la demande du client selon ses spéciations de qualité de services. Ils construisent dans un premier temps une matrice QoSnk, ou n représente le nombre total de

services qui ont la même propriété fonctionnelle, et k représente le nombre total de propriétés de QoS. La matrice obtenue est la suivante :

$$QoS=\begin{bmatrix} qp_{11} & qp_{12} & \cdots & qp_{1k} \\ qp_{21} & qp_{22} & \cdots & qp_{2k} \\ \vdots & \vdots & \cdots & \vdots \\ qp_{n1} & qp_{n2} & \cdots & qp_{nk} \end{bmatrix}$$

Chaque ligne dans la matrice correspond à un Web service candidat, et chaque colonne représenté une propriété de QoS. Suivant les indications de la matrice QoS, chaque service est modélisé en tant que vecteur de k-dimensions, chaque Web service est représenté comme un point dans les k-dimensions. Les auteurs emploient la distance euclidienne pour calculer la distance entre le vecteur de QoS spécifié par l'utilisateur et les propriétés existantes de QoSpour chaque vecteur dans la matrice suivant la formule suivante [35]:

$$distance(QP_{c,}QP_i) = \sqrt{\sum_{j=1}^{k}(qpc_j - qp_{ij})^2}$$

Ensuite, les auteurs choisissent le service Web qui a la plus petite distance euclidienne.

2.1.2. Avantages est inconvénients :

Ce tableau illustre les avantages et les inconvénients de chaque méthode :

	Avantages	Inconvénients
WSM	- C'EST une méthode simple à mettre en œuvre - L'utilisateur peut exprimer une préférence d'un paramètre par rapport un autre (importance d'un paramètre sur,un autre) en utilisant les poids.	- il n'est pas possible de considérer une contrainte globale sur toute la composition - C'est une méthode de sélection locale ce qui n'entraine pas forcément l'optimisation de la composition globale
Distance euclidienne	- La méthode permet de choisir le meilleur service qui se rapproche le plus des préférences de l'utilisateur - Simple à mettre en œuvre	- L'utilisateur ne pourra pas optimiser un critère - Ne permet pas de spécifier la préférence de l'utilisateur d'un paramètre par rapport à un autre
Sémantique	- L'utilisation d'une ontologie de QoS s'assure de l'emploie d'un moyen standard par les fournisseurs pour exprimer leurs propriétés de QoS et s'assure que les utilisateurs emploient le bon moyen pour exprimer leurs préférences	- Ne prend pas en compte les services composites

Tableau 3.1 : Avantages et inconvénients des méthodes locales [43]

2.2. Sélection globale :

La stratégie de sélection globale a pour but de choisir la combinaison de services web qui garantit la meilleure qualité globale en tenant compte des contraintes de QOS et des préférences globales assignées pour l'ensemble des tâches.

2.2.1 Les méthodes de sélection globale :

- **Le chemin optimal :** consiste à modéliser la composition des Web services sous forme d'un graphe. Ainsi, la résolution du problème de sélection revient à trouver le chemin le plus optimal qui satisfait les contraintes de QoS de l'utilisateur. Dans [33], les auteurs proposent une méthode basée sur la somme pondérée pour trouver le chemin le plus optimal dans la composition. Le problème est décrit comme suit, soit un ensemble de tâche T = (*t*1, *t*2,..., *tn*) qui doit être exécuté, un ensemble de services S = (*s*1, *s*2,..., *sm*) disponibles pour chaque tâche, et un ensemble de QoS

Q = (*q*1, *q*2,..., *qz*) pour chaque tâche. Les auteurs ajoutent deux états C et C' qui représentent respectivement l'état initial et l'état final. Pour ordonner les différentes tâches, un graphe de précédence est utilisé en partant de l'état initial C vers l'état final C' comme suit :

$$C \rightarrow t_1 \rightarrow t_2 \rightarrow \ldots \rightarrow t_n \rightarrow C'$$

Pour résoudre le problème de sélection, les auteurs modélisent le problème sous forme d'un arbre qui est construit comme suit, on commence par la tâche *t*1, on crée un nœud pour chaque service disponible pour cette tâche, de même que pour toutes les autres tâches, ensuite, on relie les différents nœuds de *t*1 avec tous les nœuds de *t*2 avec des arcs allant de *t*1 vers *t*2, on refait l'opération jusqu'à ce qu'on arrive à la tâche *tn*. On relie ensuite l'état initial C avec le s nœuds de *t*1 et chaque nœud de la tâche *tn* est considéré comme un état final C'. La Figure 2.5 montre un exemple d'un arbre d'une composition constituée de trois tâches dont chaque tâche possède deux services candidats. Ainsi, le problème de sélection revient à choisir le chemin optimal dans l'arbre qui va de la racine vers l'une des feuilles.

Pour pouvoir calculer le coût d'un chemin dans l'arbre, les auteurs associent pour chaque nœud un score qui est calculé par une somme pondérée de ses différents paramètres de QoS. Chaque paramètre est sujet à une phase de normalisation avant de calculer le score du nœud, car ils sont définis dans des échelles et des dimensions différentes, ensuite les auteurs affectent à chaque paramètre un poids qui représente la préférence de l'utilisateur d'un paramètre sur un autre. Le calcul du score d'un chemin est réalisé en faisant une addition des scores des différents nœuds qui composent ce chemin. Le chemin qui a le score le plus élevé est choisi pour exécuter la composition.

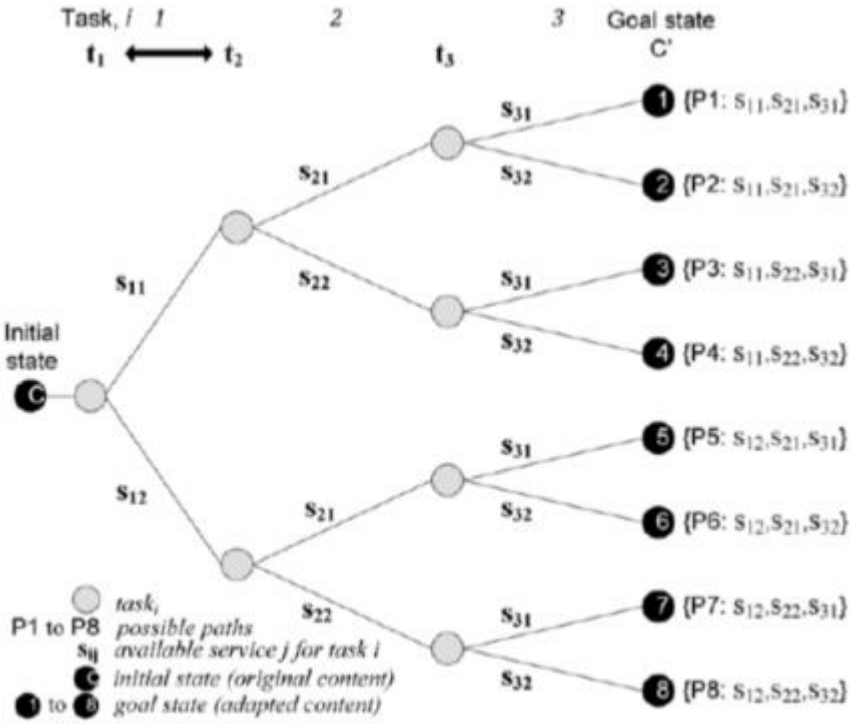

Figure 3.5 : Arbre de composition

- **Programmation mathématique :** On peut considérer le problème de sélection des services Web comme un problème multi critère concernant la QoS en tant que programme multicritères. Surtout, la programmation par objectif [37] résout le problème conformément aux préférences des clients et peut trouver une solution compromise lorsque le modèle de programmation multicritères normal n'a pas de solution. Dans [37] les auteurs modélisent le problème sous la forme d'un problème de programmation mathématique défini par un ensemble de variables, une fonction objective et un ensemble de contraintes comme suit:

$$\text{Fonction Objective}(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} c_i x_i$$

$$\text{Contraintes} \begin{cases} a_{11}x_1 + \cdots + a_{1n}x_n \leq b_1 \\ \vdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n \leq b_m \end{cases} \quad \text{Les } x_i \text{ Sont les variables.}$$

Pour résoudre le problème de sélection les auteurs supposent que la composition est constituée de m niveaux (tâches). Pour chaque tâche il y a n services candidats pour exécuter celle-ci. Les données du problème sont les suivantes :

Les variables du problème [38] : l'ensemble des services Web sij qui sont définis comme suit :

$$S_{ij}=\begin{cases}1 \text{ si le service i est sélectioné pour exécuté la tache}\\ 0 \text{ sinon}\end{cases}$$

Les contraintes [38] : un service choisi pour exécuter une tâche. Par conséquent, les variables précédemment définies ont la contrainte suivante :

$$\sum_{i=1}^{n} S_{ij} = 1 \; avec \; 1 \leq j \leq m$$

La fonction objective : Dans [38], les auteurs prennent en compte quatre critères différents (temps de réponse, fiabilité et prix). La fonction objective [34] est ainsi définie:

$$f(s_{11}, \dots, s_{nm}) = Q_{temps_{de_{réponse}}} + Q_{fiabilité} + Q_{prix}$$

Pour résoudre ce problème, les auteurs appliquent la méthode Branch-and-Bound [14] (évaluation / séparation) qui consiste à dénombrer les solutions possibles d'une manière intelligente. Cette technique élimine les solutions partielles qui ne conduisent pas à ne pas aboutir à la solution recherchée.

- **WSCEP : Web service Composition Evaluation Process [40] :**

Ce processus est basé sur le processus contient 4 étapes :

1. **Etape N °1 : Spécification des besoins /calcul des poids**
   Dans cette étape l'utilisateur doit :
   - Classer les caractéristiques de qualité (attribut).
   - Spécifier le type de la recherche (Exact, Meilleurs Parmi)
2. **Etape N °2 : Composition :** Assembler les services Web afin de construire le service demandé : déterminer toutes les compositions possibles.
3. **Etape N °3 : Evaluation** : Évaluer la qualité fournie (PQ : Provided Quality) par chaque système. L'évaluation est décomposée en deux phases :

- ✓ **Phase N°1 :** calcul des vecteurs des attribut (Ai) d'une manière incrémentale : La valeur des attributs d'un vecteur Ai est dérivée à partir des attributs de 2 autres services appartient à la composition sélectionnée en respectant les règles suivantes :

Temps d'exécution :

Si (Parallèle) :

$$A_{executonTime} = MAX(Sn_{executionTime} + Sm_{executionTime})$$

Si (Séquentielle)

$$A_{executonTime} = Sn(executionTime) + Sm(executionTime)$$

Prix :

Dans le deux cas soit séquentielle soit parallèle :

$$A_{Price} = Sn\ (Price) + Sm\ (Price)$$

Sécurité :

Si (Parallèle)

$$A_{sécurité} = (Sn_{sécurité} + Sm_{sécurité}) / 2$$

Si (Séquentielle)

$$A_{Sécurité} = MIN\ (Sn_{sécurité} ; Sm_{sécurité})$$

Réputation :

Dans les deux cas soit séquentielle soit parallèle :

$$A_{Reputation} = AVG(Sn_{Reputation}, Sm_{Reputation})$$

- ✓ **Phase N°2 :** La production de la valeur globale de qualité : Chaque composition est considéré comme un service. Après la production des valeurs des vecteurs A_i.

$$QV = \sum_{i=1}^{n} (V(A_i) * W(A_i))$$

Etape N°4 : Choix de l'application optimale

3. Conclusion :

Nous avons vu dans ce chapitre, les différents aspects liés à la sélection de services web, les méthodes de sélection locale et les méthodes globales. L'ensemble des points cités dans ce chapitre, nous donne l'opportunité pour faire un bon choix des techniques à adopter, pour la réalisation de notre travail.

Chapitre 4 :

Comparaison des processus de sélection et implémentation

1. Introduction :

La sélection de services Web consiste à choisir le service le plus adéquat selon un besoin. Elle se base principalement sur la description de service qui identifie un ensemble d'opérations et un ensemble de propriétés pour qualifier le service. Dans ce contexte plusieurs méthodes de sélection ont été proposées.

Les méthodes de sélections existantes ont en commun un certain nombre de phases qui sont présentées dans [45] sous l'appellation de « processus de sélection général » (GCS pour General COTS Sélection) :

1- Définir les critères d'évaluation basés sur les exigences et les contraintes du développeur;
2- Rechercher les composants (services) à partir des référentiels (Annuaires)
3- Filtrer les résultats de recherche basée sur les exigences. Cela permet la définition d'une liste des composants candidats prometteurs qui doivent être évalués de manière plus détaillée,
4- Évaluer les composants candidats de la liste restreinte.

Notre contribution est divisée en deux grandes parties : la première théorique et la deuxième pratique :

- ✓ Dans la partie théorique nous avons réalisé une comparaison entre les méthodes de sélection reconnu dans le contexte SOA .

- ✓ La deuxième partie, représente une conception et implémentation d'un prototype pour l'évaluation de performance des services web. L'outil réalisé peut être considéré comme un environnement de simulation. IL implémente plusieurs méthodes de sélection .

2. Comparaison des méthodes de sélection

Cette partie représente une étude comparative de trois processus de sélection de composants logiciels selon deux points de vue :

1. la couverture des phases principales de processus de sélection général(PSG) .
2. l'évaluation des services au cœur du processus de sélection général (PSG) :

2.1 Comparaison selon la couverture des phases principales de processus de sélection général (PSG) :

Le tableau ci-dessous (tableau 4.1) montre si le processus de sélection ne couvre pas ou couvre partiellement ou totalement les différentes phases de PSG :

	Spécification des besoins description des exigences fonctionnelles	Description des exigences non fonctionnelles	Sélection du fournisseur : évaluation des candidats	Sélection du fournisseur : analyse des résultats et choix
SAW	/	Totale	/	Totale
MAGIQ	/	Totale	/	Totale
Distance Eucludienne	/	Totale	/	Totale
Chemin optimal	/	Totale	/	Totale
WSCEP	Totale	Totale	Totale	Totale

Tableau 4.1. Comparaison selon la couverture des phases PSG

Comme le montre le tableau ci-dessus, toutes les méthodes offre une couverture totale du processus de description des exigences non fonctionnelles et du processus d'analyse des résultats et choix. Mais seul le processus WSCEP couvre la spécification des besoins fonctionnelle et il permet de filtrer les résultats de recherche basée sur les exigences.

2.2 Comparaison selon L'évaluation du service web au cœur du processus de sélection général :

Kontio et al [46] suggèrent de définir les critères d'évaluation de manière hiérarchique, où un ensemble d'objectifs abstraits sont progressivement affinés, en fonction de facteurs tels que les exigences des applications logicielles, l'architecture des applications et les capacités des produits existants.

En général, les critères suivant sont définis pour l'évaluation [46]:

- ✓ **PSG** : Conformité à la méthode PSG.
- ✓ **EVAL** : Evaluation Filtrage progressif (PF) qui commence par un grand groupe de produits, puis définit progressivement des critères discriminants par itérations successives des produits, où les produits moins adaptés sont éliminés.
- ✓ **UNIQ**: Convient pour une sélection d'un service unique.
- ✓ **MULT**: Aptitude à la sélection de plusieurs service.
- ✓ **ADAPT:** Adaptabilité du processus aux différents contextes.

✓ **OUTIL**: Disponibilité d'un support d'outils pour faciliter l'application de la démarche

Ces critères peuvent prendre les valeurs suivantes :

- PF : Filtrage progressif
- √ : Satisfaction complète de critère
- × : Ne satisfait pas au critère
- ~ : Satisfait partiellement, officieusement ou implicitement au critère

Approche	Facteurs de comparaison					
Nom	PSG	EVAL	UNIQ	MULT	ADAPT	OUTIL
SAW	~	X	√	×	√	×
MAGIQ	~	X	√	×	√	x
Distance Euclidienne	~	x	√	x	√	x
Chemin optimal	~	x	√	√	~	x
WSCEP	√	PF	√	√	~	x

Tableau 4.2 : 2 Comparaison selon L'évaluation du service web au cœur du PSG

Le tableau 4.2 illustre que ces méthodes peuvent être utilisées dans d'autres contexte et qu'il n'existe pas un outil implémentant les méthodes de sélection. Cela nous a conduits à concevoir et développer un environnement de simulation implémentant les approches de sélection locales et globales.

3. Conception et implémentation

Dans cette partie nous présentons un prototype d'un environnement sélection de service web nommé ESWS acronyme de : Evaluation et Sélection des Services Web. Ce prototype est **un service web** qui implémente plusieurs méthodes de sélection et peut être considéré comme un environnement d'évaluation de performance des applications basée sur les services web.

3.1. Langage de programmation :

Java est un langage de programmation à usage général , évolué et orienté objet dont la syntaxe est proche du C++. Il a été mis au point en 1991 par la firme Sun Microsystems. Il s'agissait de concevoir un langage bien adapté aux environnements de travail en réseau et capable de gérer des informations de nature variées (données numériques, informations sonores et graphiques). Java est devenu aujourd'hui une direction incontournable dans le monde de la programmation, parmi les différentes caractéristiques qui sont attribuées à son succès :

- L'indépendance de toute plate-forme : le code reste indépendant de la machine sur laquelle il s'exécute. Il est possible d'exécuter des programmes Java sur tous les environnements qui possèdent une Java Virtual Machine.

- Java est également portable, permettant à la simulation d'être distribuée facilement sans avoir à recompiler le code pour les différents systèmes.
- Le code est structuré dans plusieurs classes, dont chacune traite une partie différente de la simulation.
- Il assure la gestion de la mémoire.
- Java est multitâches : il permet l'utilisation de Threads qui sont des unités d'exécution isolées.

Environnement Logiciel :

On présentera dans ce paragraphe les différents logiciels utilisés pour la réalisation de notre projet de fin d'étude :

ECLIPSE

Eclipse est l'environnement de développement (spécialisé pour le langage Java) qui sera utilisé dans ce projet. Le choix d'Eclipse repose essentiellement sur sa gratuité, sa facilité d'utilisation, sa puissance de développement et sur tout ses nombreux plugins (bibliothèques additives).

JavaFX

Avec l'apparition de Java 8 en mars 2014, JavaFX devient la bibliothèque de création d'interface graphique officielle du langage Java, pour toutes les sortes d'application (applications mobiles, applications sur poste de travail , applications Web), le développement de son prédécesseur Swing étant abandonné (sauf pour les corrections de bogues). JavaFX est désormais une pure API Java (le langage de script spécifique qui a été un temps associé à JavaFX est maintenant abandonné). JavaFX contient des outils très divers, notamment pour les médias audio et vidéo, le graphisme 2D et 3D, la programmation Web, la programmation multi-fils etc. Le SDK de JavaFX étant désormais intégré au JDK standard Java SE , il n'y a pas besoin de réaliser d'installation spécifique pour JavaFX

3.2. Comportement fonctionnel du système développé

Notre application est un **service web** qui permet de faire trois grandes tâches : Ajouter des services Web, faire une sélection globale et sélection locale. La vision globale du comportement fonctionnel du système développé est présentée dans le diagramme de cas d'utilisation ci-dessous.

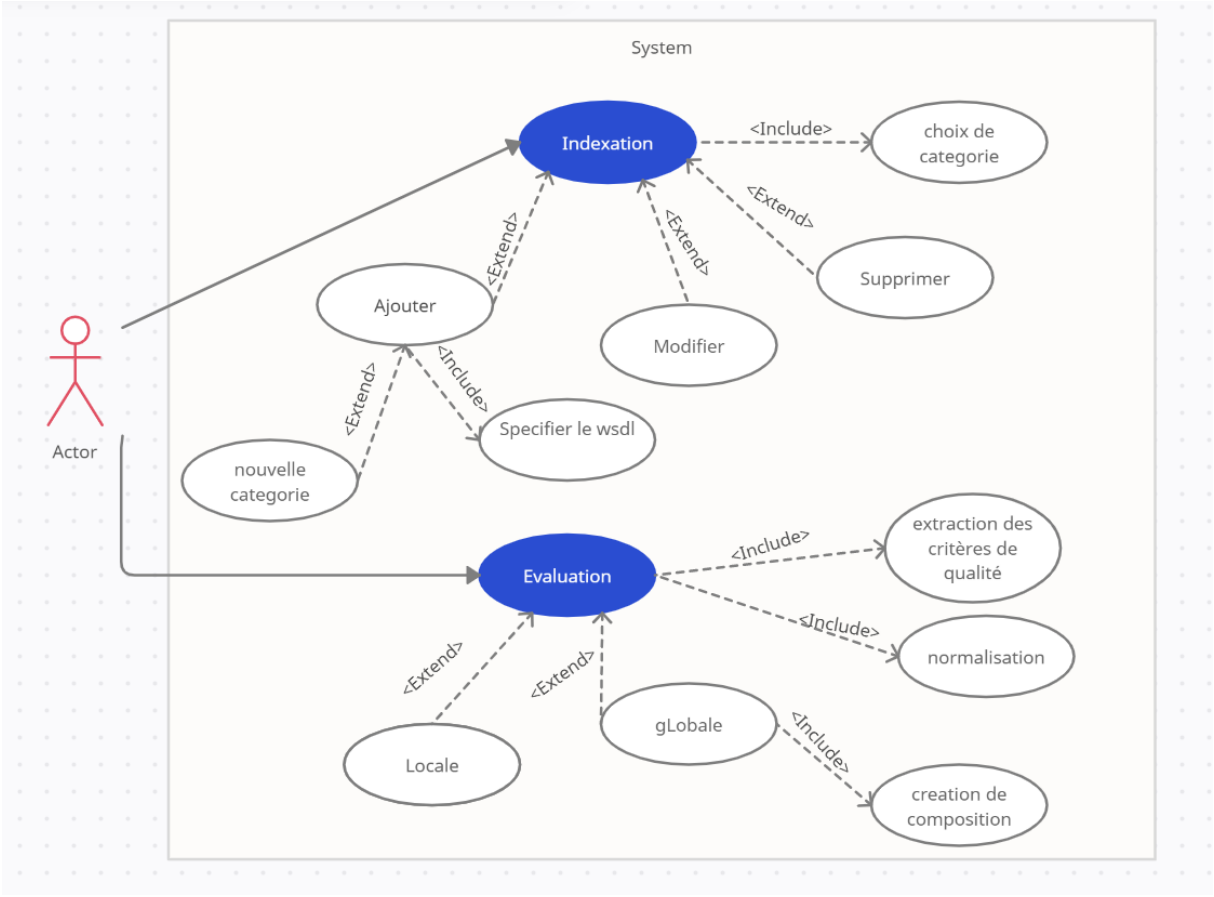

Figure 4.1 Diagramme de cas d'utilisation de l'application

La figure suivante montre l'interface principale de notre système :

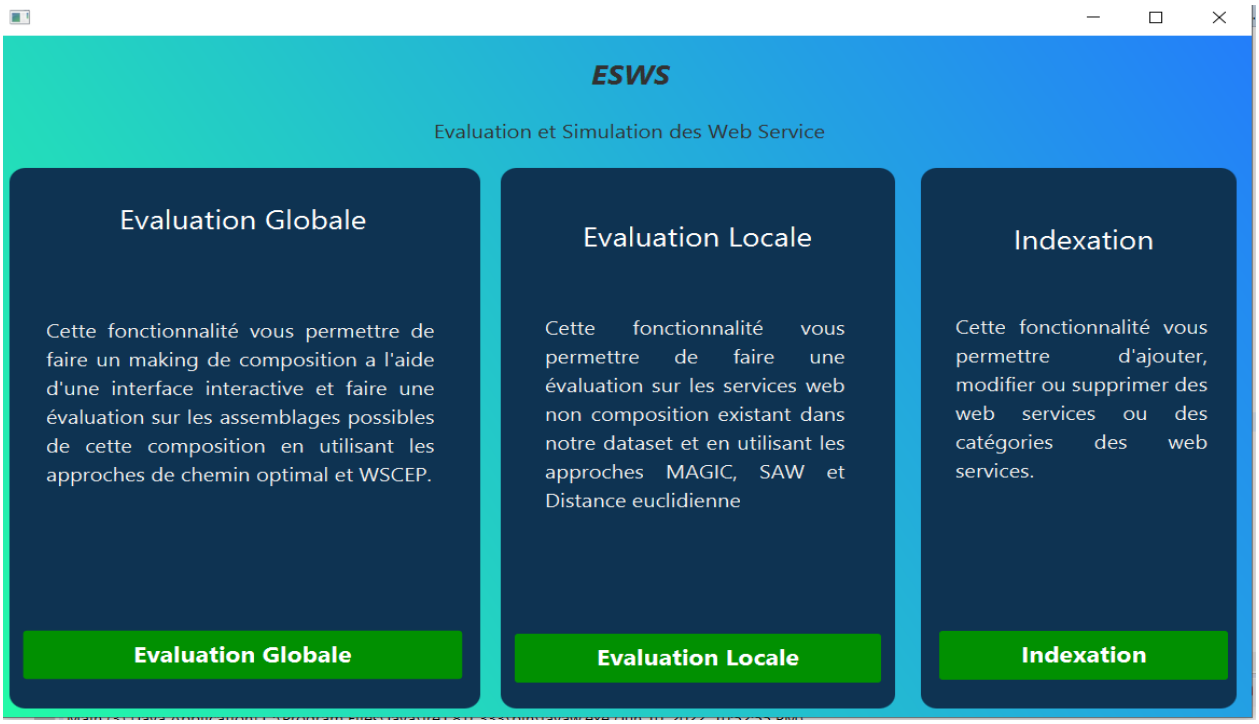

Figure 4.2 L'interface d'accueil

Pour tester les différentes approches de sélection nous avons fait une extension de la base OWLS-TC [47] en ajoutant la balise <wsdl:qos> qui permet la spécification des valeurs des attributs de QoS (le temps d'exécution, le prix, la sécurité et la réputation).

L'environnement développer permet à l'utilisateur de créer ses descriptions pour des services web et ce à l'aide du cas indexation..

La structure de WSDL :

Les figures suivantes présentent des captures de la nouvelle structure du fichier WSDL :

- La partie fonctionnelle:

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns="http://127.0.0.1:8000/" xmlns:apachesoap="http://xml.apache.org/xml-soap" xmlns:impl="http://127.0.0.1:8000
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/ " xmlns:tns="http://127.0.0.1:8000/wsdl/Academic-degreeFundingduration" xmlns:xsd="
http://www.w3.org/2001/XMLSchema" xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:intf="http://127.0.0.1:8000" xmlns:
SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" name="Academic-degreeFundingduration" targetNamespace="http://127.0.0.1:8000/n">
    <wsdl:types>
        <xsd:schema version="OWLS2WSDL Mon May 25 01:52:23 CEST 2009" targetNamespace="http://127.0.0.1:8000/wsdl/
        Academic-degreeFundingduration" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
            <xsd:annotation>
                <xsd:documentation source="Translation (OWL2XSD-SimpleType) of http://127.0.0.1:8000/ontology/portal.owl#Academic-Degree"
                />
            </xsd:annotation>
            <xsd:element name="Academic-Degree" type="Academic-DegreeType"/>
            <xsd:complexType name="DurationType">
                <xsd:choice>
                    <xsd:element name="durationID" type="DurationDATA"/>
                    <xsd:sequence>
                        <xsd:element minOccurs="0" name="name" type="xsd:string"/>
                        <xsd:element name="has-unit-of-measure" type="Time-Measure"/>
                    </xsd:sequence>
                </xsd:choice>
                </xsd:restriction>
            </xsd:simpleType>
            <xsd:simpleType name="FundingType">
                <xsd:restriction base="xsd:string"/>
            </xsd:simpleType>
            <xsd:simpleType name="DurationDATA">
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="twelve-month-duration"/>

                </xsd:restriction>
            </xsd:simpleType>
            <xsd:simpleType name="Time-Measure">
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="time-measure-day"/>
                    <xsd:enumeration value="time-measure-hour"/>
```

Figure 4.3 WSDL Partie fonctionnelle

- La partie non-fonctionnelle:

```
<wsdl:qos>
<wsdl:ExecutionTime>31.45</wsdl:ExecutionTime>
<wsdl:Price>23968.98</wsdl:Price>
<wsdl:Security>81.70</wsdl:Security>
<wsdl:Reputation>0.80</wsdl:Reputation>
</wsdl:qos>
</wsdl:definitions>
```

Figure 4.4 WSDL Partie Non-fonctionnelle

Les principales balises sont :

- <wsdl:definition> : contient les informations générale sur le service , comme le nom de service et l'adresse… etc.
- <types> : définit les types de données (schéma XML) utilisés par le service Web
- <message> : définit les éléments de données pour chaque opération.
- <portType> : décrit les opérations qui peuvent être effectuées et les messages impliqués.
- <binding> : définit le protocole et le format de données pour chaque type de port.
- <wsdl:qos> : décrit la partie non fonctionnelle ajoutée . Elle permet la spécification des valeur de qualité .

3.3.1 Indexation :

Sur Cette interface nous avons donné accès à l'utilisateur pour qu'il puisse ajouter /modifier /supprimer des services Web on spécifiant les informations du service, comme on peut ajouter des catégories supplémentaires.

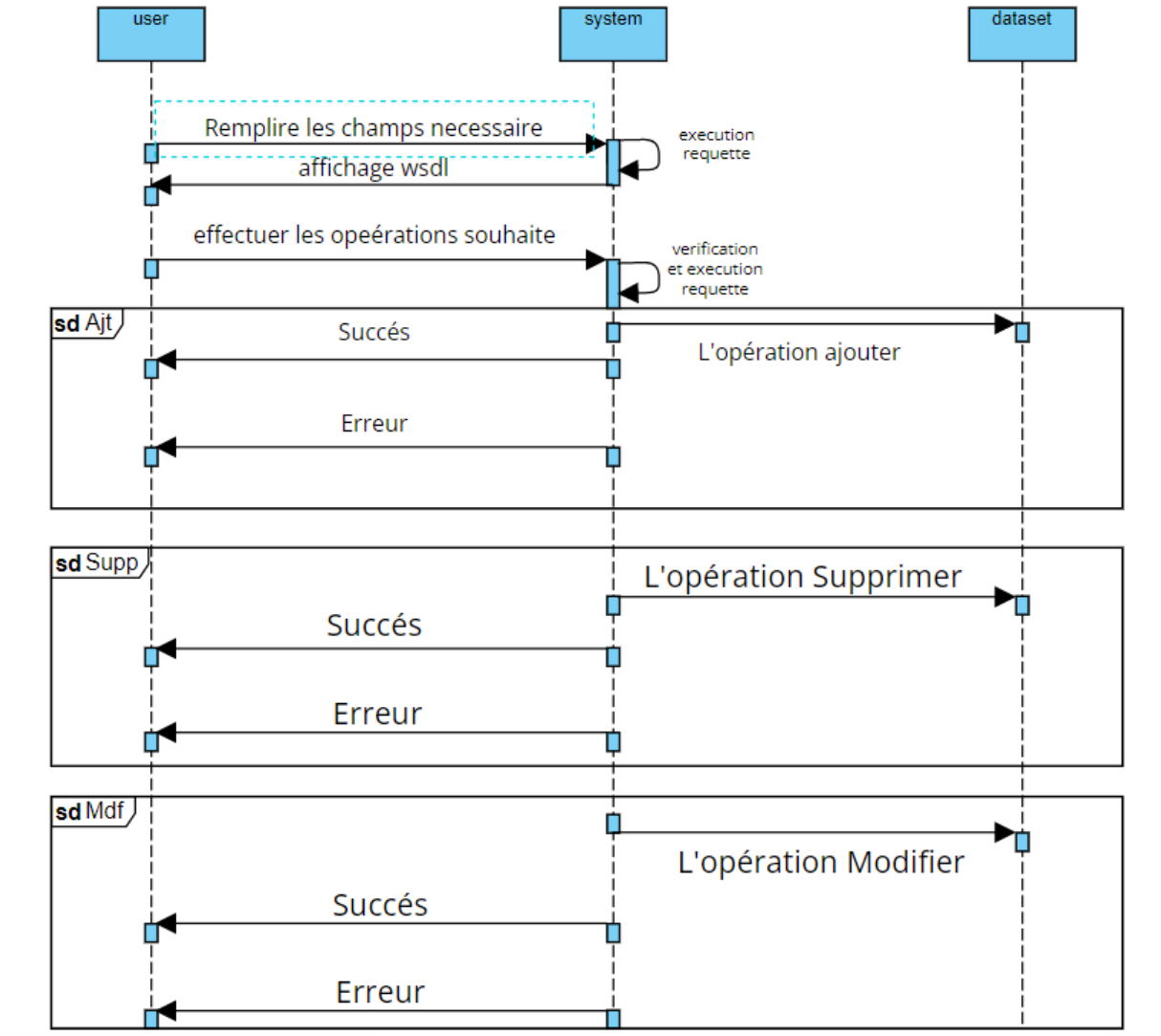

Figure 4.5 Diagramme de séquence pour l'indexation des services web

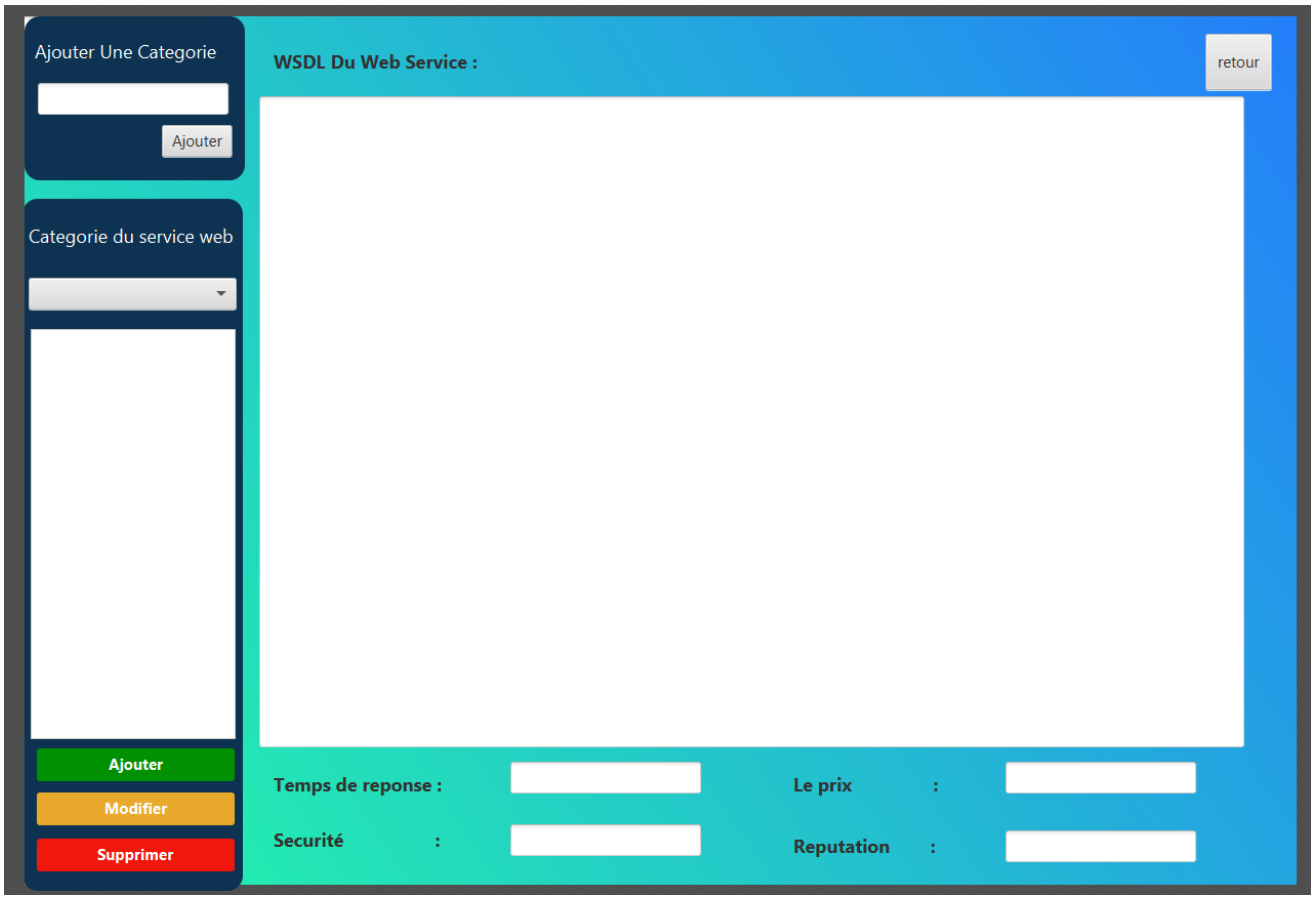

Figure 4.6 L'interface d'indexation

3.3.2. Evaluation

- **Evaluation locale :**

Ce type d'évaluation consiste à évaluer les services web non-composé en tenant compte de ces valeurs de qualité à l'aide des méthodes suivantes : **MAGIQ, SAW et Distance euclidienne .**

L'utilisateur choisie la catégorie qu'il veut et classe les critères de qualité selon ces besoins. Après l'exécution de la méthode sélectionnée (MAGIC , SAW ou Distance euclidienne) le système affiche le classement décroissant des services web (score et les critères de qualité) par rapport au besoins de client . Finalement l'utilisateur peut sélectionner le service web qui est compatible a ses besoins .

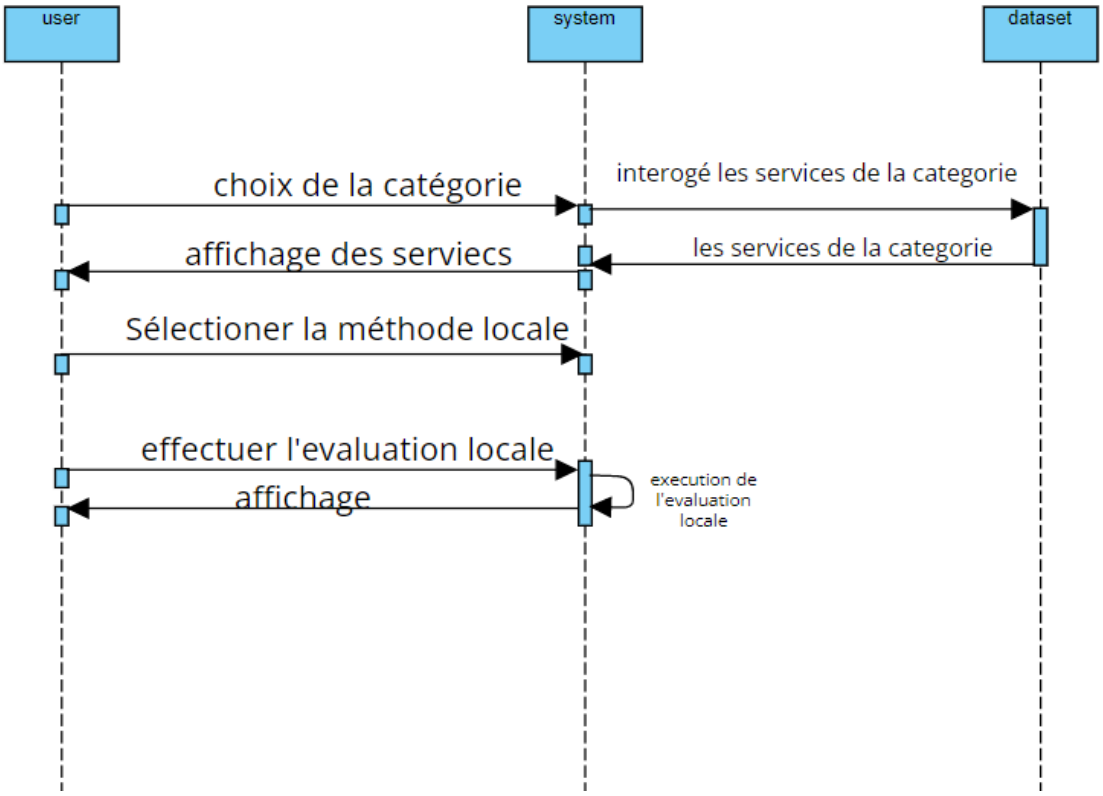

Figure 4.7 Diagramme de Séquence de la Sélection Locale

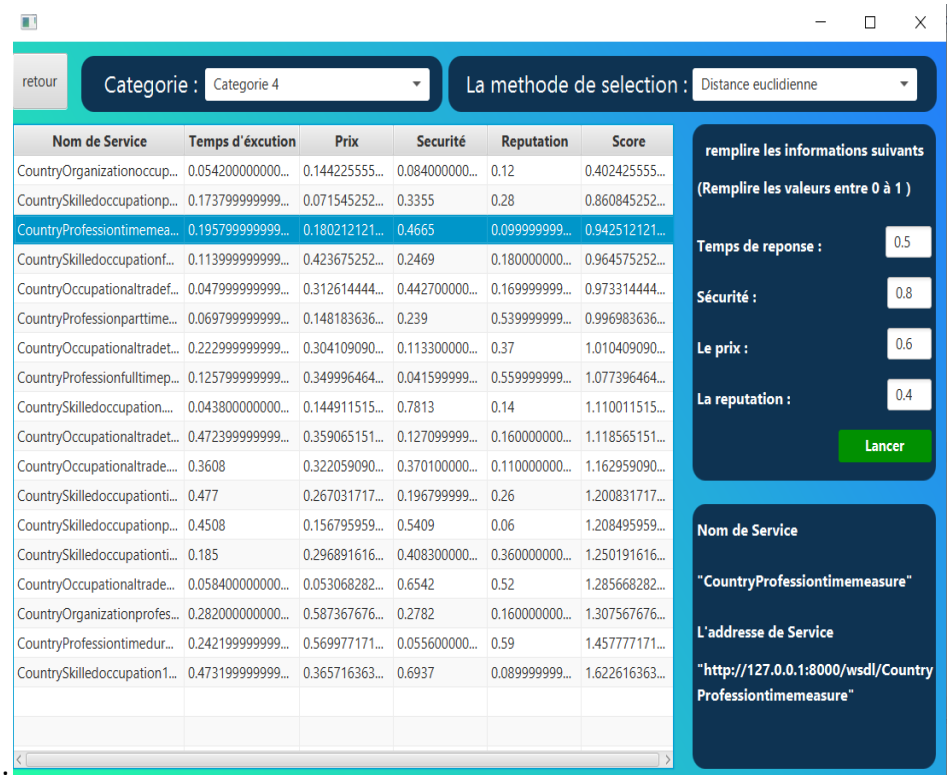

Figure 4.8 Sélection locale « méthode Distance euclidienne »

- **Evaluation globale :**

L'évaluation globale consiste a crée une composition de plusieurs services web selon les besoins d'utilisateur (figure 4.6). Le service demandé peut être réalisé avec plusieurs assemblage de services web simple. Notre système évalue toutes les compositions possibles.

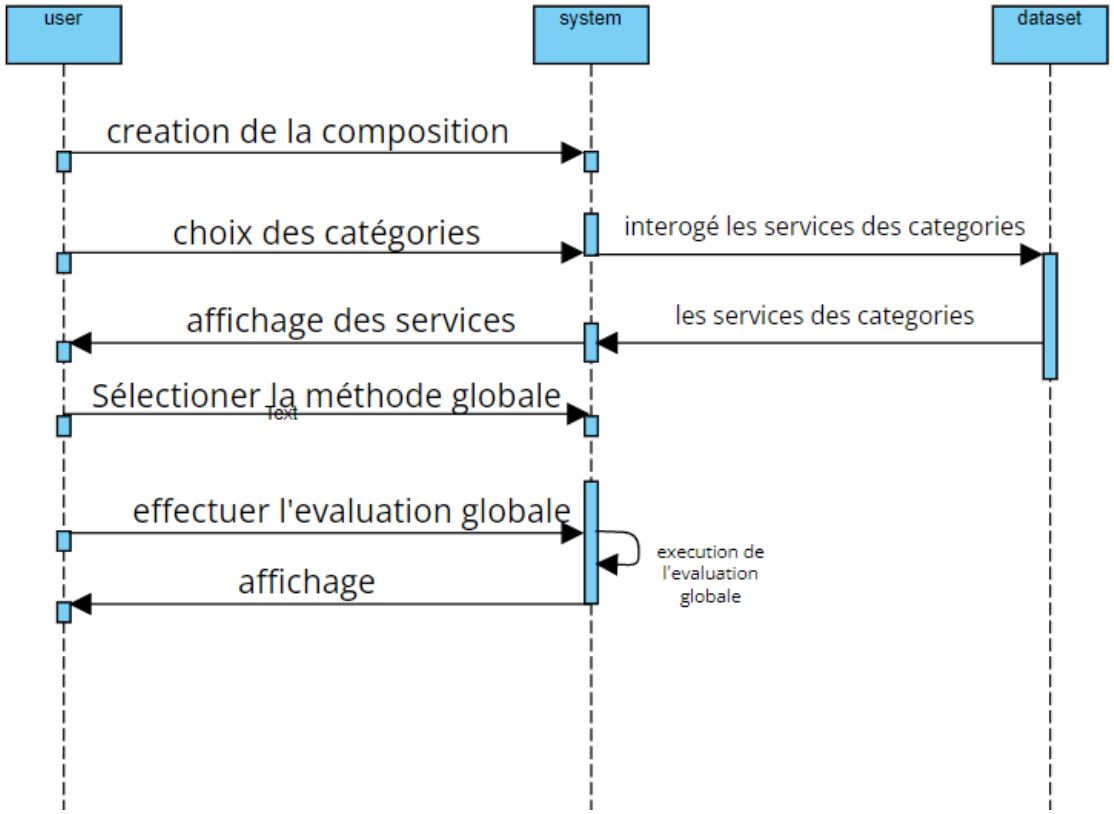

Figure 4.9 Diagramme de séquence de la méthode globale.

- **Création d'une composition :**

A l'aide de l'interface composition l'utilisateur peut ajouter des services web en précisant la catégorie et le nom de ce dernier. Le service globale et réaliser d'une manière incrémentale et ce en utilisant les deux relations : composition parallèles et composition séquentiels.

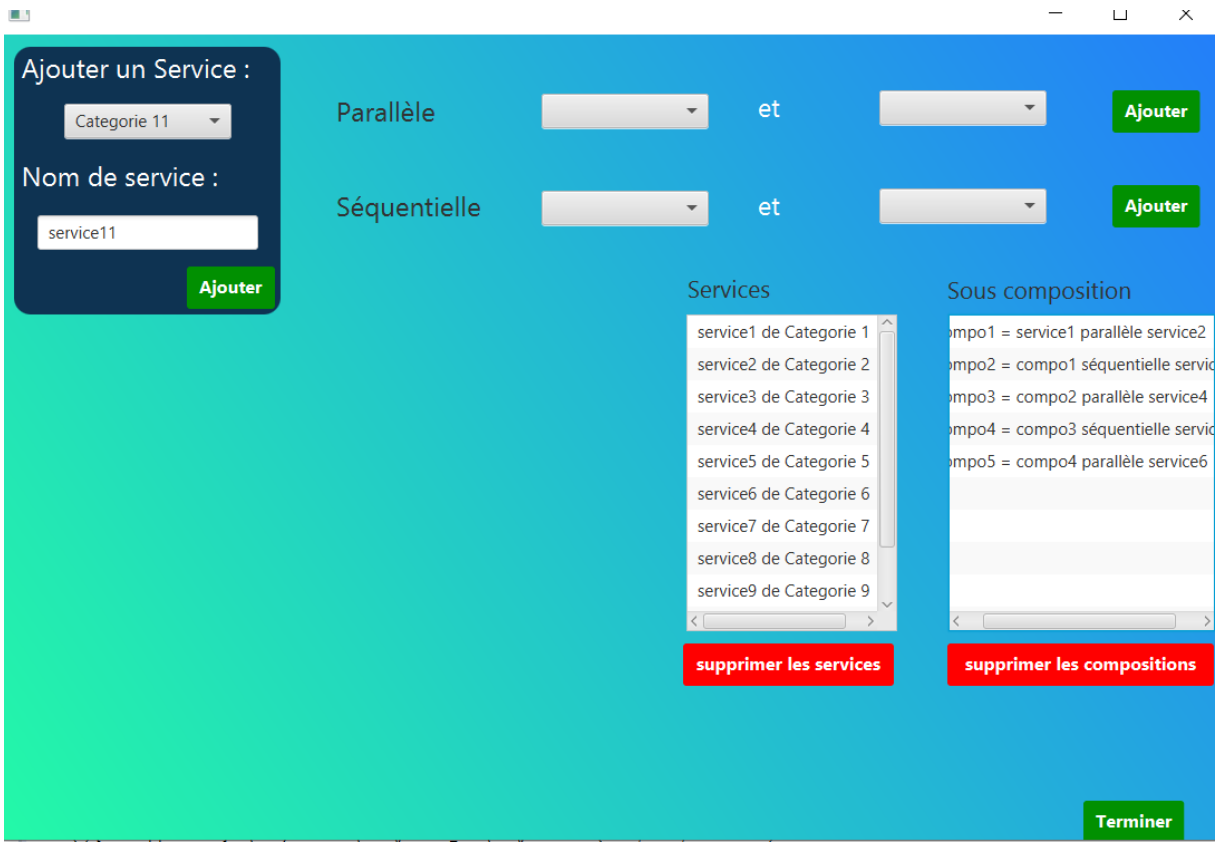

Figure 4.10 Création des compositions

Le résultat de la composition s'affiche comme suit :

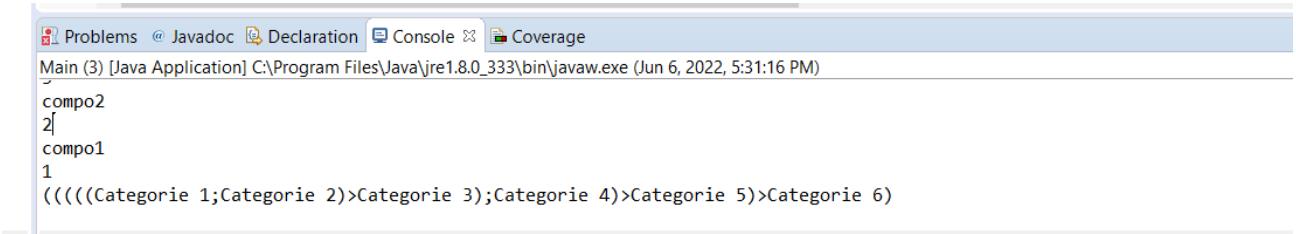

Figure 4.11 résultat final de la composition

L'application affiche à l'utilisateur une nouvelle interface pour qu'il puisse sélectionner la méthode de sélection globale voulue (WSCEP ou Le plus court chemin) si sont choix et le WSCEP donc il doit choisie soit le type de recherche "Exacte" ou le type "le meilleur parmi" , sinon la méthode SAW sera sélectionner par défaut pour l'exécution du plus court chemin.

Les figures suivantes montrent l'implémentation :

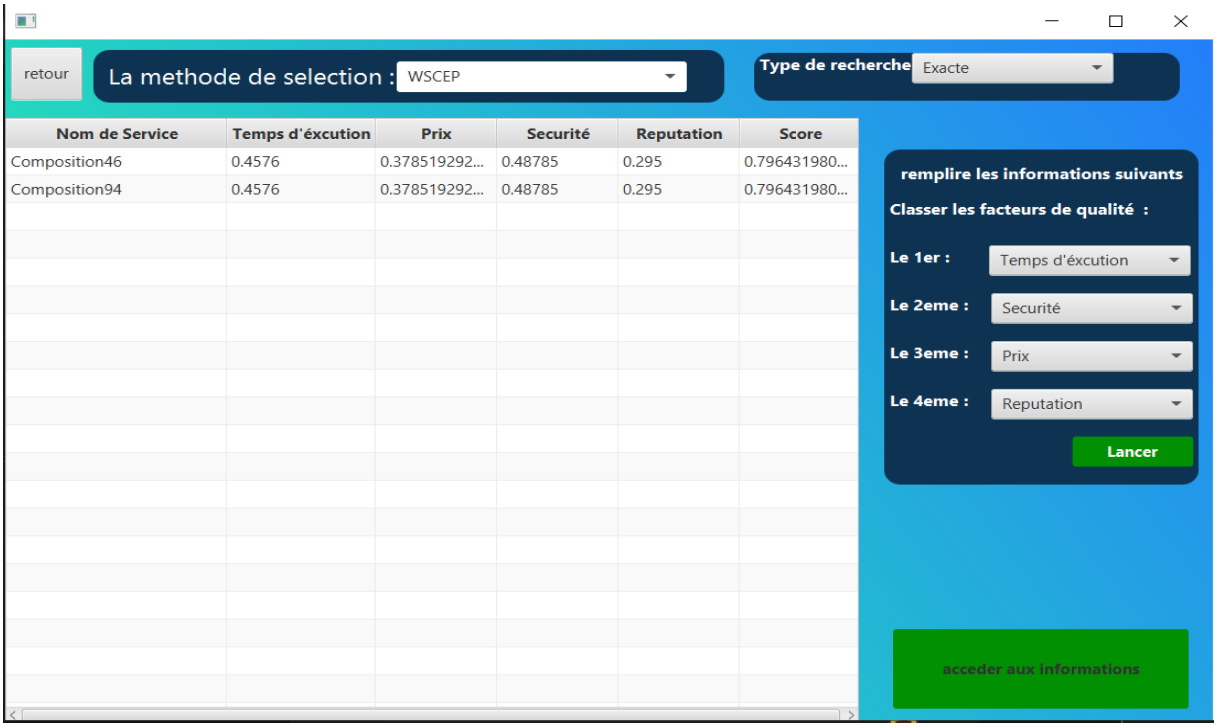

Figure 4.12 Méthode Sélection WSCEP, Type de recherche Exacte

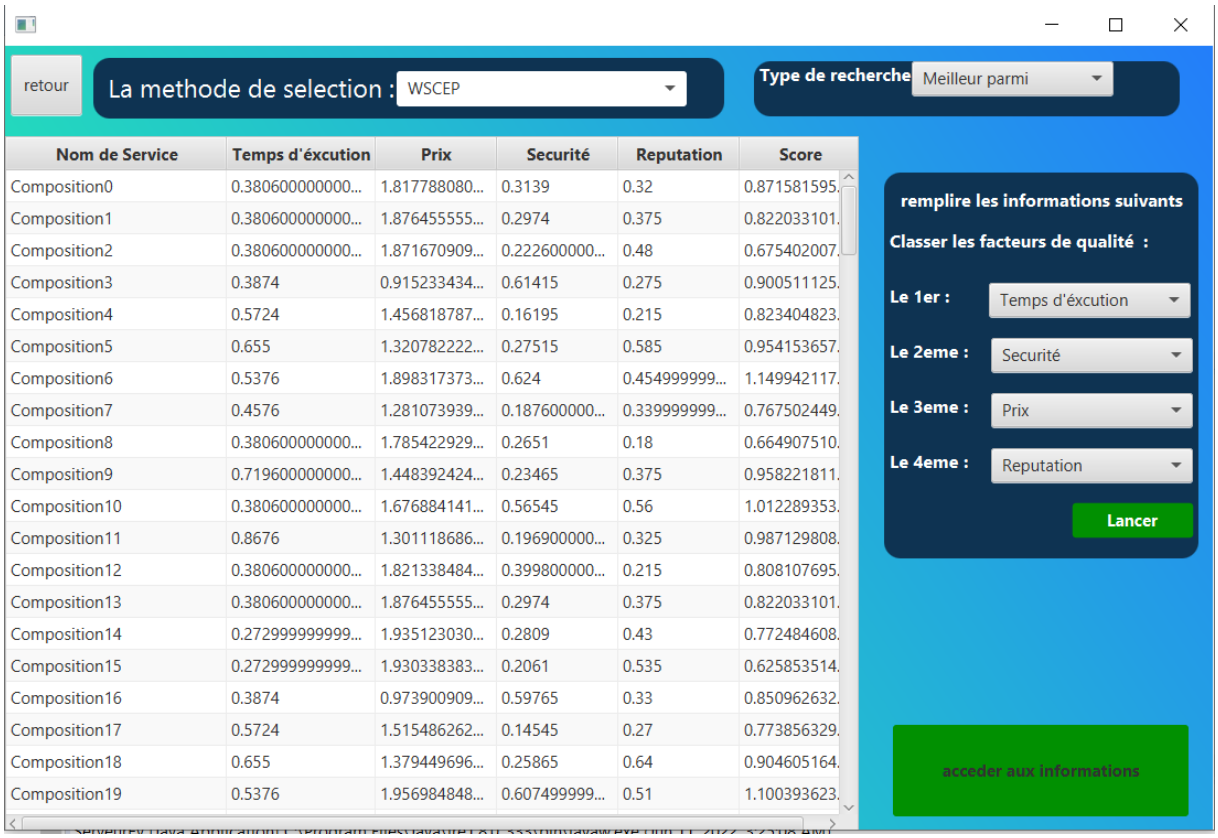

Figure 4.13 Méthode de sélection WSCEP , Type de recherche meilleur parmi

- **Normalisation des attribues :**

Chaque service doit passer par une étape de normalisation pour rendre toutes les valeurs des attributs QoS dans l'intervalle [0,1]. Cette étape est réalisée à l'aide des règles suivantes [41]:

Si l'attribut est un attribut à minimisé comme le temps d'exécution ou le prix, on utilise la règle suivante :

$$Normalise(Att_i) = \frac{Att_i^{max} - Att_i}{Att_i^{max} - Att_i^{min}} \qquad \text{(R1)}$$

Si l'attribut est un attribut à maximisé comme la sécurité ou la réputation, on utilise la règle suivante :

$$Normalise(Att_i) = \frac{Att_i - Att_i^{min}}{Att_i^{max} - Att_i^{min}} \qquad \text{(R2)}$$

$$Avec \ \ Att_i \in [Att_i^{min}, Att_i^{max}]$$

Exemple :

Si on prend l'attribut temps d'exécution (attribut à minimisé) :

Valeur avant normalisation = 50

On suppose que la valeur maximale et la valeur minimale dans l'ensembles des services sont :

- Valeur maximale : Att^{max}= 120
- Valeur minimale : Att^{max}=1

On utilisant R1

Normalisation(ExecutionTime)=$\frac{50 \ - \ 1}{120 \ - \ 1} = 0.44$

4. Conclusion

Dans ce chapitre nous avons présenté une étude comparative des processus de sélection et une conception et implémentation d'un environnement pour l'évaluation de performances des services web. Un bilan des travaux réalisés et les perspectives que l'on peut qualifier de court ou long terme seront présenté dans la section suivante.

Conclusion générale

Au terme de ce mémoire, nous procédons dans les lignes qui suivent à un récapitulatif du travail réalisé. Rappelons que l'objectif principal est de concevoir et développer un environnement d'évaluation de performance des applications logicielles et particulièrement celles basées sur l'architecture SOA.

Pour accomplir cet objectif nous avons étudié les différentes approches de sélection de service web. Après cette étude nous avons constaté que Les processus de sélection existants partagent un certain nombre de phases présentées dans PSG : 1- Définir les critères d'évaluation basés sur les exigences et les contraintes du développeur;-*2*-: Rechercher les services à partir des référentiels,-3- Filtrer les résultats de recherche basée sur les exigences. et -*4*-: Évaluer les services candidats de la liste restreinte.

Premièrement nous avons fait deux études comparatives :

1. Comparaison selon la couverture des phases principales du PSG
2. Comparaison selon l'évaluation des composants est au cœur du PSG

Après cette étude, nous avons constaté que :

- toutes les méthodes offre une couverture totale du processus de description des exigences non fonctionnelles et du processus d'analyse des résultats et choix. Mais seul le processus WSCEP couvre la spécification des besoins fonctionnelle et il permet de filtrer les résultats de recherche basée sur les exigences.
- ces méthodes peuvent être utilisées dans d'autres contexte et qu'il n'existe pas un outil implémentant les méthodes de sélection.

La deuxième étape de ce travail consiste à concevoir un service web pour d'évaluation et la sélection des services web. Le comportement fonctionnel du système développé a été présenté par diagrammes UML (Unified Modeling Language). L'implémentation nous à permet de constater que :

- le processus WSCEP maintient les valeurs détaillée des attributs de qualité durant la composition .Par exemple l'utilisateur peut connaitre le cout (en terme de prix) de la composition. Par contre le calcul du score à chaque nœud de la méthode « plus cout chemin » cache les valeurs des attributs.
- WSCEP permet de minimiser le nombre de comparaison d'une part et de prendre l'architecture de composition (séquentielle / parallèle) en considération d'une autre part.

Comme perspectives, nous envisageons un ensemble de travaux futurs qui concernent les aspects suivants :

- La prise en compte des propriétés de QoS de type qualitatives
- Traiter la problématique d'hétérogénéité des modèles de qualité et d'intégrer la vérification syntaxique et sémantique des compositions.
- Etudier d'autres paradigmes de programmation

Références Bibliographiques :

- [1] : W3C World Wide Web Consortium; “Web Services Architecture”; W3C Working Group Note 11; February 2004; http://www.w3.org/TR/ws-arch
- [2] : https://www.ibm.com/docs/en/txseries/8.1.0?topic=overview-properties-web-services , 13/04/2022 .
- [3] : Clement Jonquet and Stefano A. Cerri PhD in Informatics Senior Researcher – INRAE (MISTEA) .
- [4] : Cours de Dr. Faiçal Felhi Faculté des Sciences de Gabès .
- [5] : https://www.institut-numerique.org/ les services web 15/03/2022.
- [6] : ‘ ’ Structure d’un message SOAP avec son enveloppe’ ’ by Alyssa Walker 01/02/2022.
- [7] : https://www.wallarm.com/what/what-is-web-services-description-language-wsdl 18/02/2022.
- [8] : https://www.r4r.co.in/webservice/01/tutorial/basic/UDDI.shtml,UDDI (2002). Universal Description, Discovery, and Integration – 13/04/2022.
- [9] : Divan, S. Evaluation des performances des architectures logicielles a l’aide d’un réseaux de les d’attente. Université de Paris, 2008- 2009.
- [10] : USTHB. Cour d’évaluation de performance des réseaux de le d’attente. www.jmct.123.fr, 18/12/2008.
- [11] : Baynat, B. Théorie des les d’attente, vol. 8. HERMES Science publications, Paris, 2000.
- [12] :Atrouche Sabeha , Oulhaci Massinissa ,thèse: Evaluation des Performances d’un Web Service à l’aide d’un Réseau de Files d’attente , 2014/2015
- [13] : Aoumer, S., and Asli, L. Performance de la methode de stabilité forte dans des système d’attente avec priorité. université de Bejaia, 2007.
- [14] : Constantin Samoila Expert Cloud et Infogérance . Février 5, 2013 https://www.pentalog.fr/blog/developpement-technology/vue-densemble-de-la-performance-des-applications-logicielles, 16/01/2022
- [15] : Sad, S. Cour évaluation et simulation, Décembre 2009.
- [16] : Nakannishi, T. The development of aread traffic simulation system in broad areas mathematics and computers insimulation, vol. 6. 1995, pp. 207212.
- [17] : Martin, F. Modelisation et evaluation de performance

prévisionnelle d'architectures avionique modulaire integrée. l'onera toulouse, 1999.

- [18] : Braunschweig, B. La simulation sur micro-ordinateur(les modeles dedynamique des systeme,vol. 169 pp. Eyrolles, 1985.
- [19] : ISO/ IEC 9126-1 : Software engineering - Product quality - Part 1 Quality model ; Tech. Rep, International Organization for Standardization, 2001.
- [20]:http://mariepascal.delamare.free.fr/IMG/pdf/coursQualite.pdf 04/23/2022
- [21] : ISO 9000 :2000 :Quality management systems Fundamentals and vo- cabulary ;Geneva,Switzeriand : Intemafional Organization for Standardization 01/02/2022 .
- [22] : YAHLALI AZZA, CHAABEN DJAMILA, Etude comparative des approches d'évaluation de la qualité des applications basées sur l'assemblage de composants logiciels, 2020/2021 .
- [23] :https://rmod-files.lille.inria.fr/Team/Texts/Papers/Mord14a-Chapitrequalite.pdf 17/02/2022 .
- [24] : YAHLALI.M , CHOUARFIA Abdallah **:** Software Component Quality Evaluatio ;International , January 2014
- [25] : Hadjila, F. (2014). Composition et interopération des services web sémantiques. PhD thesis.
- [26] : 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).
- [27] : BOUDJELABA Hakim. Sélection des services Web sémantiques Université A / Mira de Béjaia, juin 2012(dernière consultation 27 mars 2020)
- [28] https://eprints.qut.edu.au/377/1/377.pdf (dernière consultation 31 mars 2020)
- [29] : Lazim Abdullah, C.W. Rabiatul Adawiyah Simple Additive Weighting Methods of Multi criteria Decision Making and Applications: A Decade Review.
- [30] : Alireza Afshari, Majid Mojahed and Rosnah Mohd Yusuff Simple Additive Weighting approach to Personnel Selection problem (dernière consultation 1 avril 2020)
- [31] : Lazim Abdullah, C.W. Rabiatul Adawiyah Simple Additive Weighting Methods of Multi criteria Decision Making and Applications: A Decade Review.
- [32] : James G. Dolan, MD Multi-criteria clinical decision support: A primer on the use of multiple criteria decision making methods to promote evidence-based, patient-centered healthcare (dernière consultation 7 avril 2020)

- [33] : Mehri Saeid, Abdul Azim Abd Ghani, and Hasan Selamat Rank-Order Weighting of Web Attributes for Website Evaluation (dernière consultation 7 avril 2020)
- [34] : L. Taher, H. El Khatib, and R. Basha. A Framework and QoS Matchmaking Algorithm for Dynamic WebServices Selection (dernière consultation 1 avril 2020
- [35] : Hazar TURKI KOSSENTINI, Dr. Leila BACCOUCHE,Pr. Henda HAJJAMI BEN GHEZALA. Etude de cas pour la selection des services web basée sur les contraintes temporelles (dernière consultation 31 mars 2020)
- [36] : Michael C. Jaeger and Gero Muhl QoS-based Selection of Services: The Implementation of a Genetic Algorithm(dernière consultation 3 avril 2020)
- [37] : Cui, Kumara, and Lee: Scenario Analysis of Web Service Composition based on Multi-Criteria Mathematical Goal Programming Service Science 3(4), pp. 280-303, 2011 SSG & INFORMS (dernière consultation 4 avril 2020)
- [38] : T. Sen, M. Raiszadeh, and P. Dileepan. A branch and bound approach to the bicriterionscheduling problem involving total flowtime and range of lateness.(Dernière consultation 4 avril 2020)
- [39] : M.YAHLALI, WSQEP: Web Service Quality Evaluation Process" , International Conference on Pattern Analysis and Intelligent Systems (PAIS'2018), Tebessa, Algeria, 2018.
- [40] : JAMES G. Dolan, MD Multi-criteria clinical decision support: A primer on the use of multiple criteria decision making methods to promote evidence-based, patient-centered healthcare (dernière consultation 7 avril 2022).
- [41] : BENKHALED Hamid, REGRAGUI SidAhmed, Selection de Web services à base de la qualité de service Université Tahar Moulay de Saida, juin 2017 (dernière consultation 1 avril 2020)
- [42] : HARA Alem, YAHIAOUI Assia, Sélection des services Web Université A / Mira de Béjaia,juin2012(dernière consultation 1 avri l2020).
- [43] : 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.
- [44] : Zahid Javed, Ahsan Raza Sattar, Muhammad Shakeel Faridi, Unsolved Tricky Issues on COTS Selection and Evaluation, Global Journal of Computer Science and Technology, Vol 12 Issue 10 Version 1.0 , 2012.
- [46] : Kontio, J. and Tesoriero, R., "A COTS selection method and experiences of its use", In The Twentieth Annual Software Engineering Workshop, Greenbelt, Maryland, 1995.
- [47]:http://lpis.csd.auth.gr/systems/OWLSSLR/datasets.html?fbclid=IwAR3CJ6tEN9QWJkH1MjrXh-ND_HKiZIoieTill4Z0Q9_LFphDNgKORo8cuUE, 03/06/2022