



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

N° d'ordre :IA9/M2/2019

Mémoire

présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Intelligence artificielle**

Une approche de sélection des services Web basée sur la logique floue

Par :
MIHI Europa

Soutenu le 06 juillet 2019, devant le jury composé de :

Zernadji Tarek	MCB	Président
Belouaar Houcine	MCB	Rapporteur
Fekraoui Farah	MAA	Examineur

Remerciements

Ce mémoire est le résultat d'un travail d'une année de recherche menée à l'Université de Biskra, Département d'Informatique.

Tout d'abord, je tiens à remercier Dieu, le tout puissant et miséricordieux, qui m'a donné la force et la patience pour accomplir ce travail.

Ce mémoire n'aurait pas pu avoir lieu sans un encadrement de grande qualité, celui de Mr. BELOUAAR Houcine pour avoir proposé et dirigé ce sujet de mémoire et pour son accueil bienveillant et ses conseils avisés, il a bien voulu répondre patiemment et aimablement à mes nombreuses questions. Ainsi je le remercie pour son aide précieuse et continue pendant la rédaction de ce mémoire.

Je remercie les honorables membres de jury d'avoir accepté d'être membre de mon jury de thèse, d'évaluer mon travail et pour nous avoir honorés de leur présence.

Je tenais également à remercier tous mes enseignants de la faculté d'informatique à l'université Mohammed kheider Biskra et plus précisément de la branche IA.

Enfin j'adresse mes gratitudes à tous mes collègues et mes amies.

Europa Mihi

Dédicace

Je remercie Dieu de m'avoir donné le courage pour accomplir
ce modeste travail que je dédie :

À mes très chers parents pour leurs interminables conseils,
et encouragement continu.

À mon frère et mes sœurs.

À mes chers amis.

À tous ceux qui me sont chers.

ملخص:

أصبح اختيار خدمة الويب من بين العديد من الخدمات المماثلة مهمة حساسة للغاية بالنسبة لمستهلك خدمات الويب لأنه يواجه صعوبة في الاختيار و لن يتمكن من تحديد الخدمة الأكثر ملاءمة. يمكن أن يكون الفرق الوحيد بين خدمات الويب المشابهة هو معايير جودة الخدمة (QoS) و تحديدا خصائصها غير الوظيفية التي لها تأثير كبير على اختيار أفضل خدمة.

من ناحية أخرى، فإن معايير الجودة غير الوظيفية المتعلقة بخدمات الويب للمستهلك غير دقيقة و غير مؤكدة أحيانا و غامضة لأن المعطيات الرقمية غير قادرة في بعض الأحيان على نموذجة مواقف حقيقية و بالتالي المنطق الضبابي يأتي لترجمة هذا التقييم مستعملا معايير جودة الخدمة بطريقة مناسبة للمستهلك.

في هذا العمل، نقترح نهجا يعتمد على المنطق الضبابي لاختيار خدمات الويب، و ذلك بأخذ بعين الاعتبار المعطيات اللغوية للمستهلك ثم تقديم خدمات الويب المطلوبة.

مهمتنا هي إضافة وحدة تعتمد على المنطق الضبابي تقوم بتحويل المعطيات من قيم حقيقية إلى قيم ضبابية بالإضافة إلى التصنيف متعدد المعايير لخدمات الويب المماثلة.

بالنسبة لتصنيف خدمات الويب المتشابهة، اخترنا حل استخدام أسلوب ترتيب التفضيل عن طريق التشابه مع الحل المثالي الذي يعتمد على طريقة (TOPSIS Flou).

الكلمات المفتاحية: خدمة ويب، اختيار، جودة الخدمة (QoS)، المنطق الضبابي، الترتيب، ضبابي TOPSIS.

Résumé

La sélection d'un service Web parmi plusieurs services similaires est devenue une tâche très délicate pour le consommateur de services car il est confronté à l'embarras du choix et il sera incapable de décider quel service est le plus approprié. La seule différence entre des services Web similaires peut être leur qualité de service (QoS) et plus précisément leurs propriétés non fonctionnelles qui ont une grande influence sur la sélection du meilleur service.

D'autre part, les critères de qualité des services liés aux services Web consommateurs sont imprécis et parfois incertains et ambigus car les données quantitatives sont insuffisantes pour modéliser des situations réelles donc la logique floue vient pour traduire cette évaluation en présentant les critères de qualités d'une manière adéquate pour le consommateur.

Dans ce travail, nous proposons une approche basée logique floue de sélection des services Web afin de prendre en compte les données floues linguistiques du consommateur et de le rendre ensuite les services web souhaités.

Notre travail consiste en l'ajout d'un sous-système qui consiste en la fuzzification des critères quantitatifs ainsi que le classement de services web similaires. Pour effectuer le classement des services Web similaires, nous avons opté pour une solution qui utilise la technique de l'ordre de préférence par similitude à la solution idéale (TOPSIS flou).

Mots clés : Service Web, sélection, qualité de service (QoS), logique floue, classement, TOPSIS flou.

Table des figures

1.1	Modèle fonctionnel de l'architecture SOA	7
1.2	Structure des services web	10
1.3	Architecture en Pile des services web	11
1.4	Traitement d'un message SOAP [24]	14
1.5	Structure de message SOAP	15
1.6	Structure d'un document WSDL	17
1.7	les trois types de l'annuaire UDDI	18
2.1	Fonction d'appartenance caractérisant un ensemble classique (a) et d'un ensemble floue (b)	25
2.2	Fonction d'appartenances	26
2.3	Intersection des fonctions d'appartenances	26
2.4	Union des fonctions d'appartenances	27
2.5	Variable linguistique de la qualité du service	28
2.6	Forme triangulaire	29
2.7	Forme trapézoïdale	29
2.8	Forme gaussienne	30
2.9	Système Inférence Flou	31
3.1	Architecture de l'approche proposée	37
3.2	Fonctionnement de l'approche	39
3.3	Comportement du sous-système flou	40
3.4	Variable linguistique Response-time	43

3.5	Variable linguistique Price	43
3.6	Variable linguistique Reliability	44
3.7	Variable linguistique Availability	44
3.8	Diagramme de séquence de publication des services	46
3.9	Diagramme de séquence de selection de services web.	47
4.1	Environnement logiciel utilisé.	50
4.2	Interface principale de eclipse.	51
4.3	Logo MySQL.	52
4.4	Logo PhpMyAdmin.	52
4.5	Logo XAMPP.	53
4.6	la table "Service-web".	54
4.7	Interface principale de l'application.	54
4.8	interface d'inscription du fournisseur.	55
4.9	interface d'inscription du client.	56
4.10	Interface Login du fournisseur.	56
4.11	Interface Login du client.	57
4.12	Interface d'espace client et affichage des services similaires.	58
4.13	Résultat calculé par TOPSIS Flou.	58
4.14	Etape 01 : TOPSIS Flou.	59
4.15	Etape 02 : TOPSIS Flou.	59
4.16	Etape 03 : TOPSIS Flou.	60
4.17	Etape 04 : TOPSIS Flou.	60
4.18	Etape 05 : TOPSIS Flou.	61
4.19	Interface Fournisseur ajout d'un nouveau service.	61

Liste des tableaux

- 3.1 Représentation du paramètre coût en flou 36
- 3.2 Paramètre Response time en représentation floue 42
- 3.3 Paramètre Price en présentation Floue 42
- 3.4 Paramètre Reliability en présentation Floue 42
- 3.5 Paramètre Availability en présentation Floue 43

Glossaire

* UDDI : Universal Description Discovery and Integration (Description et intégration de la description universelle)

* SW : service Web.

* TOPSIS : The Technique for Order of Preference by Similarity to Ideal Solution (La technique de l'ordre de préférence par similarité à la solution idéale).

* QoS : quality of service (qualité de service).

* WSDL : Web Services Description Language (Langage de description des services Web).

* FNIS : the fuzzy negative ideal solution (la solution idéale floue négative).

* FPIS : the fuzzy positive ideal solution (la solution idéale floue positive).

* MCDM : Multicriteria Decision Making (Prise de décision multicritères).

* UML : Unified Modeling Language (Le Langage de Modélisation Unifié).

* SOA : Service Oriented Architecture (Architecture Orientée Service).

Table des matières

Remerciements	i
Dédicace	ii
Résumé arabe	iii
Résumé	iv
Table des figures	vii
Liste des tableaux	viii
Glossaire	ix
Introduction générale	1
1 Service Web	4
1.1 Introduction	4
1.2 Architecture Orientée Service (SOA)	5
1.2.1 Définition de SOA	5
1.2.2 Objectifs	6
1.2.3 Avantages d'une architecture orientée service	6
1.3 Architecture des services Web	7
1.3.1 Définition d'un service Web	7
1.3.2 Caractéristiques des services Web	8
1.4 Différents types d'architecture des services web	9
1.4.1 Architecture de référence	9
1.4.2 Architecture étendue	11
1.5 Standards de base de services Web	13

1.5.1	Le protocole SOAP	13
1.5.2	WSDL (Web Services Description Language)	16
1.5.3	UDDI (Universal Description Discovery and Integration)	18
1.6	Conclusion	19
2	Logique Floue	20
2.1	Introduction	20
2.2	Historique de la logique floue	21
2.3	Logique floue vs logique classique	22
2.4	Y a-t-il un besoin de logique floue	22
2.5	Définition de la logique Floue	23
2.6	Domaines d'application :	23
2.7	Théorie de la logique floue	24
2.7.1	Les sous-ensembles flous	24
2.7.2	Les opérations de base sur les sous-ensembles flous	25
2.8	Variable linguistique	27
2.9	Système d'inférence Flou :	30
2.9.1	Fuzzification :	31
2.9.2	Inférence :	31
2.9.3	Déffuzification :	31
2.10	Conclusion	32
3	Modélisation	33
3.1	Introduction	33
3.2	Qualité de service	34
3.2.1	Paramètres de qualités de services	34
3.3	Une approche de sélection des services Web basée sur la logique floue	35
3.4	Architecture générale	36
3.5	Description des différents acteurs	37
3.6	Fonctionnement de l'approche	39
3.7	Diagramme de séquence de Publication des services web	46

3.8	Diagramme de séquence de Sélection de service web	47
3.9	Conclusion	48
4	Implémentation	49
4.1	Introduction	49
4.2	Environnement de développement	49
4.2.1	Environnement matériel et logiciel	49
4.2.2	Langage de programmation	50
4.2.3	Outils et technologies	51
4.3	La base de données	53
4.4	Présentation des interfaces graphiques	54
4.4.1	Interface d'accueil	54
4.4.2	Interfaces principales	55
4.4.3	Interface d'Espace Client	57
4.4.4	Interface d'Espace Fournisseur	61
4.5	Conclusion	62
	Conclusion générale	63
	Bibliographie	65

Introduction Générale

Un service Web peut être définie essentiellement comme une abstraction sémantiquement bien définie d'un ensemble d'activités informatiques ou physiques impliquant un certain nombre de ressources, destinées à répondre à un besoin client ou à un besoin métier.

Avec la popularité de l'internet, les services Web sont devenus indispensables dans presque tous les domaines. Ils constituent une catégorie de solutions Web utilisées dans les industries, telles que les services bancaires, shopping, et les communications. Les services Web peuvent être décrits simplement comme n'importe quel service offert sur le Web. Par exemple, dans une application Web météo, un utilisateur ouvre un site Web météorologique et saisit des données (par exemple : code postal), le serveur de site traite les données et envoie la réponse, dans ce cas, une prévision météorologique.

Les applications B2B (intégration de business-to-Business) et les applications B2C (Business-to-consumer) sont de plus en plus utilisées par les entreprises commerciales pour s'adapter au marché via le Web.

La technologie évolue et beaucoup de services sont disponibles sur le net, le problème de la recherche dans les services Web a émergé, il devient important d'être en mesure de localiser les services qui répondent à nos besoins avec plusieurs variétés. Plusieurs proposition ont été avancées pour résoudre ce problème et plusieurs normes ont été définies, mais aucun d'entre eux a été efficace ou est maintenant acceptée comme le moyen d'effectuer une recherche de service.

Problématique et objectifs

Le nombre de services Web offrant la même fonctionnalité sur le Web est en croissance sans cesse, Une simple requête d'un client peut avoir des centaines de réponses voir des milliers de ce que fait, le client tombe dans l'embarras de choix et se trouve face à un problème de choix entre les services similaires retournés par la requête. Tout fois, il est très rare de répondre à une requête client par un seul service.

La capacité d'accéder efficacement aux services Web est nécessaire. En outre, comme les services Web avec des fonctionnalités similaires sont censés être fournis par des fournisseurs concurrents, le défi majeur est de concevoir des stratégies d'optimisation pour trouver les "meilleurs" services Web.

En plus de cela, dans de nombreux cas, la valeur de la propriété QoS peut être difficile à définir avec précision car les qualités des services liés aux consommateurs de services Web sont imprécises et parfois incertaines et ambiguës en raison de l'état mental et du manque d'information des consommateurs sur le contenu des services Web et aussi sur leur QoS. Encore, les valeurs des QoS (notamment QoS qualitative) ne sont pas faciles à définir avec précision, donc, il est donc préférable que les préférences utilisateur utilisées comme propriétés QoS soient floues car mieux adaptées à l'interprétation des termes linguistiques. Par conséquent, la logique floue peut être appliquée pour soutenir la représentation de contraintes QoS imprécises.

L'objectif de ce travail consiste à proposer une approche de selection des Services Web basée sur la logique floue.

Pour cela nous proposons d'exploiter la méthode TOPSIS Flou. Cette approche doit d'une part permettre aux utilisateurs des services Web de lancer leurs requêtes en termes linguistiques et d'autre part, présenter les différents paramètres de qualité de service en valeurs floues permettant à la fin de retourner aux clients les services les plus adéquats.

Plan du mémoire

Après une Introduction générale décrivant le contexte du travail, la problématique et la solution proposée, le présent mémoire est organisé en comme suit :

- Le premier chapitre «Services Web» : Il présente la technologie des services Web et les

principaux standards qu'elle supporte, parmi les quels on cite les protocoles SOAP, WSDL, UDDI, il montre aussi leurs avantages et leurs inconvénient.

- Le deuxième chapitre «Logique Floue» : qui est une approche au raisonnement humain. Les fonctions d'appartenances sont le coeur de la logique floue, et le Systeme d'inférence flou ou le systeme prend le rôle de l'être humain et commence a raisonner, ses trois étape : Fuzzification, Inférence et Déffuzification.
- Le troisième chapitre «Modélisation» : Nous commencerons notre première contribution sous la forme d'une conception de l'architecture de notre système, nous présenterons aussi la descriptions détaillée de ses composants.
- Le quatrième chapitre «Implémentation et étude de cas» : Dans ce dernier chapitre, nous allons présenter l'environnement logiciel et matériel sur lequel le système sera réalisé et validé, les détails d'implémentation de notre application, propose une mise en oeuvre pratique et une présentation des résultats expérimentaux.

Ces quatre chapitres sont clôturés par une conclusion générale qui résume les résultats de notre travail, et présente les perspectives que nous souhaitons réaliser dans le futur.

Chapitre 1

Service Web

1.1 Introduction

De nos jours, les entreprises et les systèmes d'information expriment un grand besoin pour échanger des informations et des services. Ceci nécessite des langages communs de communication. Les efforts de standardisation de ces langages ont donné lieu à un nouveau domaine de recherche connu sous le nom de "protocolesB2B". Une technologie émergente dans ce domaine a permis de tracer quelques pistes intéressantes pour la communication entre entreprises. Cette technologie est celle de services Web. Les services Web sont un paradigme naissant qui vise à la transposition des architectures par composant dans le cadre du Web. Un service Web est un composant logiciel qui offre des services à travers une interface standardisé. La particularité des services Web est l'utilisation de la technologie Internet comme infrastructure pour la communication entre eux.

Les fonctionnalités offertes par les services Web sont relativement simples, alors que les besoins des utilisateurs sont de plus en plus complexes, à tel point qu'un seul service Web ne peut pas les satisfaire. L'implication de la composition des services Web, dans le but de créer un nouveau service Web dit composite ainsi que la sélection automatique de ces dernières offres des nouvelles perspectives qui répondent aux exigences complexes des utilisateurs [1].

1.2 Architecture Orientée Service (SOA)

Plusieurs paradigmes de développement de logiciel ont été proposés pour faciliter la création, l'exposition, l'interconnexion et la réutilisation d'applications à base de services. C'est pour cette raison que l'informatique distribuée a dû passer par le modèle client-serveur, l'architecture à plusieurs niveaux (l'orienté objet, l'orienté composant), les standards basés CORBA (Common Object Request Broker Architecture) de l'OMG (Object Management Groupe), Java RMI (Remote Method Invocation) de Sun, DCOM (Distributed Component Object Model) de Microsoft pour enfin aboutir à l'architecture orientée service (SOA) qui est née pour répondre aux inconvénients des technologies orientées composants tels que : la complexité, le manque de compatibilité, en plus ils sont difficilement interopérables entre eux et la modification dans un composant conduit généralement à une défaillance du système. Comme, ils sont souvent utilisés en Intranet [10].

1.2.1 Définition de SOA

Dans la littérature, il y a plusieurs définitions de l'architecture orientée Service. Voici quelques définitions de l'architecture SOA :

«Une architecture orientée services(SOA) est essentiellement une collection de services. Ces services communiquent entre eux. La communication peut impliquer soit un simple transfert de données, soit deux ou plusieurs services coordonnant une activité. Certains moyens de connecter les services les uns aux autres sont nécessaires.» [11].

Une architectures orientées services (SOA) est un paradigme qui permet l'intégration d'applications et de ressources de manière flexible en :

— Représentant chaque application ou ressource sous la forme d'un service exposant une interface standardisée;

— Permettant à un service d'échanger des informations structurées (messages, documents, objets métier);

— Coordonnant et organisant les services afin d'assurer qu'ils puissent être invoqués et utilisée de manière efficace [12].

« une architectures orientées services (SOA) est un paradigme récent pour la construction

d'applications logicielles à grande échelle à partir de services distribués [13]. L'un des principaux intérêts de SOA est fondamentalement la capacité sous-jacente d'une telle architecture à être intrinsèquement évolutive; car l'idée sous-jacente de SOA est que les services sont faiblement couplés et que la SOA peut être adaptée à son environnement. »

«L'architecture orientée services est un modèle utilisé pour transformer les composants d'un système d'information en services qui peuvent être intégrés pour construire des processus intermétiers. Les services sont fournis à d'autres composants via un protocole de communication, généralement sur un réseau. Les principes d'orientation de service sont indépendants de tout fournisseur, produit ou technologie.» [14].

Le plus important dans l'Architecture SOA est la normalisation. Il définit les spécifications de tous les types d'application à découvrir et à communiquer les uns avec les autres. D'une part, cela permet également de faire usage de systèmes existants pour coopérer avec les nouvelles technologies modernes. D'autre part, la description de l'interface standard permet d'implémenter le service avec différents langages, plates-formes, modèles d'objets ou systèmes de messagerie [3].

1.2.2 Objectifs

L'architecture SOA vise trois objectifs importants :

- Identification des composants fonctionnels.
- Définition des relations entre ces composants.
- Etablissement d'un ensemble de contraintes sur chaque composant de manière à garantir les propriétés globales de l'architecture [1]

1.2.3 Avantages d'une architecture orientée service

SOA possède plusieurs avantages, parmi eux :

- SOA réside dans le fait que les applications réparties n'ont plus besoin d'un système de middleware réparti commun pour communiquer, mais seulement des protocoles de communication interopérables sur Internet.
- SOA peut donc être représentée par une interconnexion de multiples points d'accès.

- Le contrat de service du modèle des SOA s'inspire directement du modèle des contrats professionnels de service. Dans le cas des services web, ce contrat est un document WSDL.
- Une grande tolérance aux pannes et une maintenance facilitée.
- Une bonne modularité permettant de remplacer facilement un composant (service) par un autre puisque ça ne gêne pas le fonctionnement des autres services.
- La possibilité de l'évolution et la réutilisabilité des composants.
- La découverte des services disponibles, afin d'être sûr que les utilisateurs potentiels découvriront le service dont ils ont besoin.
- Une réduction du coût de développement et d'intégration d'application.
- La mise à l'échelle est rendue possible grâce à la découverte et à l'invocation de nouveaux services lors de l'exécution [15].

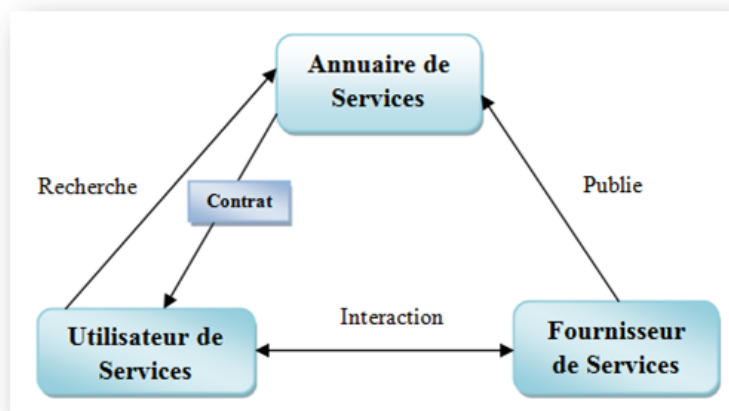


FIGURE 1.1 – Modèle fonctionnel de l'architecture SOA

1.3 Architecture des services Web

1.3.1 Définition d'un service Web

Un service web est un programme informatique permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués. Il s'agit donc d'un ensemble de fonctionnalités exposées sur internet ou sur un intranet, par et pour des applications ou machines, sans intervention humaine, et en temps réel [1].

Le groupe de World Wide Web Consortium (W3C) dans le cadre du service web a émis la définition suivante : « Un service web est un système logiciel conçu pour permettre l'interopérabilité de machine-à-machine sur un réseau. Dispose d'une interface décrite dans un format exploitable par machine (en particulier WSDL). D'autres systèmes exercent une interaction avec le service web d'une manière prescrite par sa description en utilisant des messages SOAP, typiquement transmis avec le protocole HTTP avec une sérialisation XML , en conjonction avec d'autres standards relatifs au web» [16].

Les différentes définitions proposées s'accordent au moins sur l'idée qu'un service Web est un nouveau type de composant logiciel ayant la capacité de publier ses fonctions sur Internet sous forme de services, de rendre ces services facilement invocables et de les mettre à disposition par l'intermédiaire de protocoles Internet standardisés (HTTP, XML) [1].

1.3.2 Caractéristiques des services Web

Les services Web possèdent des caractéristiques qui leur permettent une meilleure intégration dans les environnements hétérogènes :

Interopérabilité : Tout comme le couplage faible, l'interopérabilité est un critère primordial pour les services Web. L'interopérabilité permet à des applications clientes de communiquer avec des services programmés dans des langages de programmation différents et s'exécutant sur des plateformes (logicielles ou matérielles) différentes. Les services web se basent sur des standards XML pour simplifier la construction des systèmes distribués et la coopération entre ces derniers [17].

Réutilisable : Le principe de réutilisabilité permet de réduire les coûts de développement en favorisant la réutilisation de parties de codes qui ont déjà été réalisées. Une fonctionnalité, développée sous forme de service Web peut être réutilisée et combinée à d'autres fonctionnalités afin de composer de nouveaux services pour satisfaire une partie des besoins [18]. Ceci permet un gain de temps considérable, une réduction importante de la taille des applications par évitement de la duplication et facilite également la maintenance [19].

Simplicité d'utilisation : L'utilisation des standards tels que XML et HTTP a mis en valeur la simplicité de la manipulation des services Web, comme ils sont adaptés aux systèmes interconnectés d'une manière flexible [17].

Couplage faible : Le couplage est une métrique indiquant le degré d'interaction entre deux ou plusieurs composants logiciels. Le couplage faible est le concept qui permet de minimiser les dépendances et assurer l'évolution, la flexibilité et la tolérance. Cela signifie qu'il y a une séparation logique qui isole une application cliente et un service afin d'éviter une dépendance physique entre les deux, c'est à dire le demandeur du service ne connaît pas forcément le fournisseur. Pour ce faire, le client communique avec le service par échange de messages dans un format standard. Il est ainsi possible de modifier un service sans briser la compatibilité avec les applications clients existantes [19].

Modulaire : Les services Web fonctionnent de manière modulaire et non pas intégrée. Cela signifie qu'au lieu d'intégrer dans une seule application globale toutes les fonctionnalités, on crée (ou on récupère) plusieurs applications spécifiques qu'on les fait inter-opérer entre elles, et qui remplissent chacune, une de ces fonctionnalités [20].

Autonome et auto-descriptif : Le cadre des services Web contient en lui-même toutes les informations nécessaires à l'utilisation des applications, sous la forme de trois fonctions : trouver, décrire et exécuter dans une manière facilement reconnu par n'importe quelle application externe [18]. La capacité des services à se décrire par eux-mêmes permet d'envisager l'automatisation de l'intégration de services [15].

1.4 Différents types d'architecture des services web

Deux types d'architecture existent pour les services Web : La première dite architecture de référence, elle contient trois couches principales. La seconde architecture est plus complète, elle utilise les couches standards de la première architecture en ajoutant au-dessus d'autres couches plus spécifiques. Elle est appelé architecture étendue ou encore en Pile.

1.4.1 Architecture de référence

L'architecture classique de « référence » des services Web se fonde sur le modèle SOA (Service Oriented Architecture) qui fait intervenir trois acteurs : le client ou l'utilisateur du service, le fournisseur de service, ainsi qu'un intermédiaire jouant le rôle d'annuaire des services.

Le Fournisseur du service : Correspond à la personne ou à l'organisation propriétaire du

service qui réalisera effectivement le service demandé. Il publie son service en fournissant la description des services (les propriétés fonctionnel et non-fonctionnel) au format WSDL dans l'annuaire de services afin que les clients puissent découvrir et accéder au service. Alors, il représente l'environnement d'hébergement et d'exécution du service. Il est constitué de trois couches de bases [21] :

* **La couche de données** : contient les différentes bases de données utilisées par le service.

* **La couche applicative** : c'est la plateforme de développement qui assure l'exécution du service web.

* **La couche de description** : elle expose les fonctionnalités du service via un fichier WSDL.

L'annuaire : Standardisation des moyens d'accès à un service à partir d'une requête portant sur le contenu d'un service ou un fournisseur. Il correspond à un registre de descriptions des services Web qui offre aux fournisseurs le moyen de publier et d'indexer leurs services Web sur le réseau. Il permet également aux clients de rechercher, localiser et trouver ces services selon plusieurs critères répondant à leur besoins. En d'autres termes, l'annuaire joue le rôle d'intermédiaire entre les clients et les fournisseurs de services. Comme, il assure le regroupement, le stockage et la diffusion des descriptions des services web [22].

Le client : Il représente n'importe quel consommateur du service Web. D'un point de vue technique, le demandeur de service est constitué de l'application qui va rechercher et invoquer un service en envoyant une demande en XML (REST, XML-RPC, SOAP). Le client peut être une simple application Web, comme il peut être un autre service Web [22].

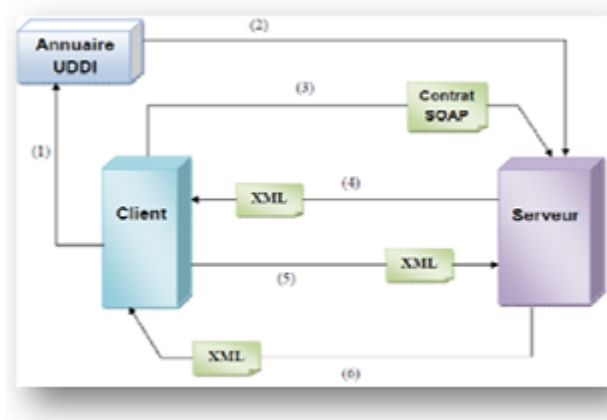


FIGURE 1.2 – Structure des services web

1.4.2 Architecture étendue

Cette architecture est aussi appelée « pile des services Web », du fait qu'elle est constituée de plusieurs couches se superposant les unes aux autres où chaque couche s'appuyant sur un standard particulier et répond à des préoccupations fonctionnelles différentes telles que la sécurité, la description, la messagerie fiable, le transport et les transactions.

On retrouve, au-dessus de la couche de transport, les trois couches formant l'infrastructure de base décrite précédemment. Nous apportons une explication des différents types de couches de l'architecture [10] :

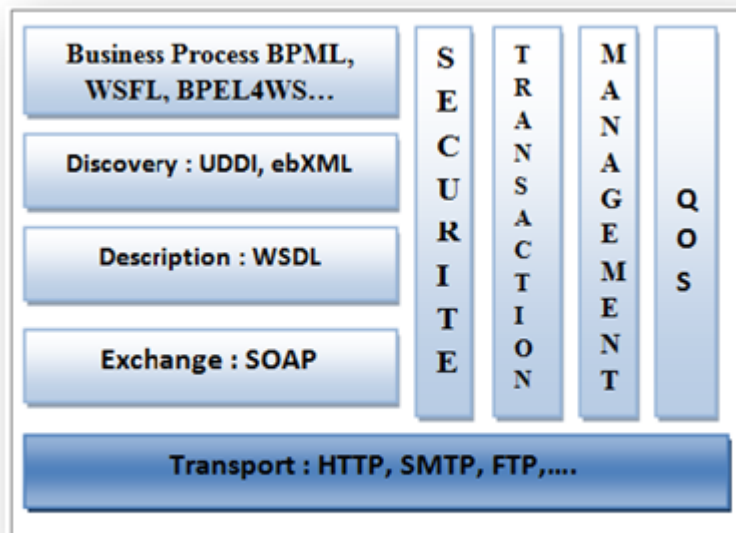


FIGURE 1.3 – Architecture en Pile des services web

- L'infrastructure de base (Discovery, Description, Exchange) : Définit les composants de l'architecture de base.
- Le transport : Est le mécanisme utilisé pour déplacer les requêtes de service du consommateur de services au fournisseur, et les réponses des services du fournisseur au consommateur de service pour pouvoir se découvrir et dialoguer entre eux. Le protocole le plus utilisé dans cette couche est HTTP. Cependant, d'autres protocoles peuvent être utilisés, tels que le SMTP, FTP. . . permettant ainsi aux services Web de rester indépendants du mode de transport utilisé.

- Le protocole de communication : Est un mécanisme utilisé par le fournisseur et le client de service pour communiquer le contenu des requêtes et des réponses. 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.
- Couches transversales (Sécurité, Transactions, Administration, QoS) : Ces couches rendent viable l'utilisation effective des services dans le monde industriel. Ce sont les aspects concernant la qualité de services [10] :
 - * **La sécurité** : La gestion de la sécurité est actuellement le frein le plus important à la mise en place d'architectures distribuées à base de services Web. Elle représente la normalisation des moyens permettant de couvrir les problématiques d'authentification et de gestion des droits d'accès. En plus, l'ensemble de règles qui peuvent être appliqués lors de l'authentification, de l'autorisation, et du contrôle d'accès des consommateurs qui invoquent les services. Les consommateurs peuvent aussi demander l'utilisation de règles de sécurité. Par exemple, dans le cas où il veut garantir l'intégrité des données transmises.
 - * **Transaction** : Normalisation des moyens permettant de garantir l'intégrité des transactions longues impliquant plusieurs services Web. Un environnement d'intégration de services peut contenir un élément chargé de garantir la cohérence des données utilisées et des résultats retournés par une collaboration (ou composition) de services.
 - * **Management** : Est le processus chargé de surveiller et de contrôler de manière proactive le comportement d'un service ou d'un ensemble de services.
- La couche Business Processus (Business Process) : Cette couche supérieure permet l'intégration de services Web, elle établit la représentation d'un «Business Process» comme un ensemble de service Web. De plus, la description de l'utilisation de différents services composant ce service est disponible par l'intermédiaire de cette couche

1.5 Standards de base de 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, nous présentons dans cette section les trois standards de base, à savoir, le protocole SOAP, le langage WSDL et l'annuaire UDDI. Ces trois standards sont basés sur le langage XML.

1.5.1 Le protocole SOAP

SOAP (Simple Object Access Protocol) : SOAP est un protocole léger bâti sur XML. Il permet la transmission des informations structurées (message) entre objet distant dans un environnement distribué et décentralisé (SOA). Le transfert se fait à l'aide des protocoles http et SMTP. Le protocole SOAP est particulièrement utile pour exécuter des dialogues requête-réponse RPC (Remote Procedure Call).

SOAP est un protocole adopté par le Consortium W3C. Le Consortium W3C crée des standards pour le Web : son but est donc de créer des standards pour favoriser l'échange d'information.

SOAP veut dire Simple Object Access Protocol. Si l'on voulait traduire cette définition en français cela donnerait Protocole Simple d'Accès aux Objets. En effet, le protocole SOAP consiste à faire circuler du XML généralement, via du http sur le port 80 (ou en utilisant un autre protocole). Cela facilite grandement les communications, car le XML est un langage standard et le port utilisé est le port 80, qui ne pose donc pas de problème pour les firewalls (pare-feu) de l'entreprise, contrairement à d'autres protocoles [23].

les caractéristiques du protocole

- SOAP permet une normalisation des échanges de données. Les données sont encodées en XML et échangées par des appels de procédures à distance en utilisant HTTP/SMTP/POP comme protocole de communication.
- Simple, extensible et permet le diagnostic des erreurs.
- Message unidirectionnel (Requête-Réponse).
- Fonctionne de manière synchrone et asynchrone.

- Indépendant de la plate-forme et du langage.
- N'est pas perturbé par un pare-feu [1].

Fonctionnement

La requête SOAP transmise par le client à l'aide des protocoles de transmissions (exemple HTTP) passe par un Listener() qui permet de adresser (cas de java) aux bibliothèques, ces bibliothèques donnent au client l'accès aux méthodes des services. Si l'exécution des méthodes a besoin de informations de la base de données alors il doit connecter avec la base de données.

Tout d'abord un message SOAP est un document XML qui doit avoir la forme suivante [24] :

- La structure de l'enveloppe SOAP qui définit une structure générale décrivant le contenu, le destinataire, et la nature du message.
- Les règles d'encodage SOAP qui définissent le mécanisme de sérialisation utilisé pour échanger des objets.
- SOAP RPC qui définit, pour les utilisations synchrones, une convention de représentation des appels et des réponses des procédures distantes.

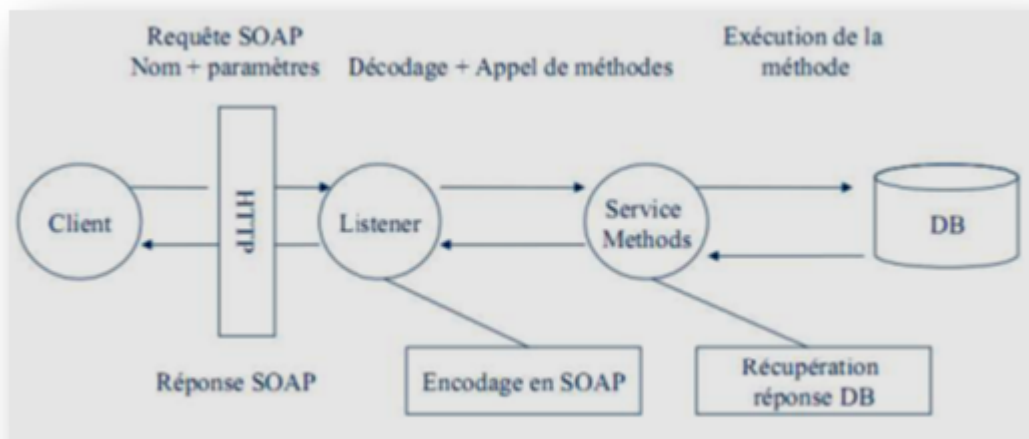


FIGURE 1.4 – Traitement d'un message SOAP [24]

Les composants d'un message SOAP

Un message SOAP est composé des parties suivantes comme la montre la figure suivante :



FIGURE 1.5 – Structure de message SOAP

- Le HTTP Header : Le protocole http envoie une requête POST. L'entête http se trouve juste avant le message SOAP, et définit le destinataire du message, les règles d'encodage HTTP, etc. Le champ SOAP Action peut être utilisé pour indiquer l'intention de la requête SOAP. Cette information peut être utilisée par un firewall pour filtrer les messages. Ce champ est obligatoire mais peut être vide si on n'indique pas l'intention de la requête.
- L'enveloppe SOAP : L'enveloppe contient l'espace de nommage définissant la version de SOAP utilisée, et les règles de sérialisation, et d'encodage.
- Le header SOAP : Est optionnelle, extensible. Les balises XML qui permettent de symboliser cette partie sont : `<env:Header>` et `</env:Header>` qui peuvent être complétées par des attributs permettant de définir le domaine de noms du service Web. En fait, l'entête permet principalement d'ajouter des informations sur le comportement des différents nœuds intermédiaires, lors de traitement du message.
- Le Body SOAP : L'élément corps de message SOAP contient des données spécifiques à l'application. Les balises symbolisant cette partie sont : `<env:Body>` et `</env:Body>`. Les données doivent donc être sérialisées selon l'encodage XML. En plus, des données de cette partie peuvent transporter un type spécial comme : les messages d'erreurs [1].

SOAP RPC

SOAP RPC définit les conventions permettant d'utiliser SOAP comme un RPC, et le format des messages pour effectuer une requête ou envoyer une réponse. SOAP RPC se base sur les spécifications de XML-RPC.

1.5.2 WSDL (Web Services Description Language)

Le WSDL est un langage qui permet de décrire les services Web, et en particulier, les interfaces et protocoles relatifs aux services Web dans le format XML, et cache tout le détail de l'implémentation du service Web, cela permettra au requérant d'un service de comprendre les données qui seront échangées lors de la communication entre les applications et découvrir comment un service Web peut être invoqué indépendamment de toute plate-forme ou langage de programmation [25].

Le WSDL permettant de fournir les spécifications nécessaires à l'utilisation d'un service Web en décrivant [26] :

- Les méthodes et les paramètres correspondants
- Les réponses fournies en retour.
- Les points d'accès aux services (Les protocoles de communication et de transport utilisés).
- Les formats de données d'entrée et de sortie pouvant être traités.
- L'ensemble des opérations et des messages qui peuvent être transmis vers et depuis un Web service donnée.
- La localisation du service où il peut être invoqué (URI).

La structure générale d'un document WSDL d'un service Web est représentée par la figure suivante :

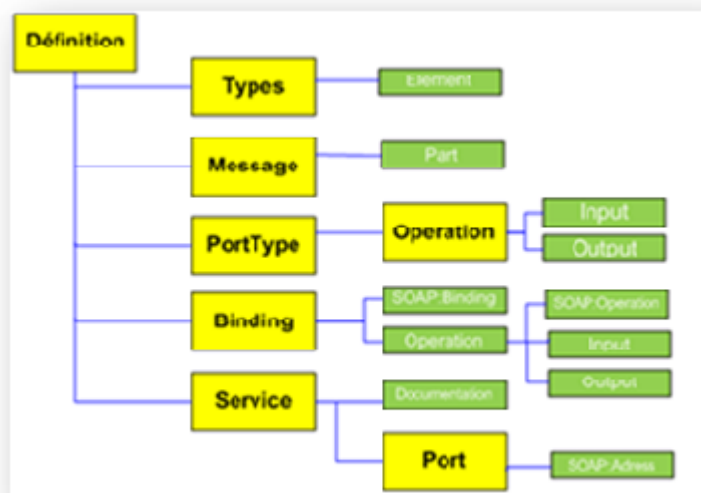


FIGURE 1.6 – Structure d'un document WSDL

Une description WSDL d'un service Web est faite sur deux niveaux, niveau abstrait et niveau concret [27] :

— La description abstraite : Elle définit d'une façon abstraite l'interface du service, c'est à dire les informations qui sont indépendantes d'un contexte de mise en œuvre.

* **Définitions** Cet élément contient la définition du service. C'est la racine de tout document WSDL. Il définit le nom du service Web et contient tous les éléments qui décrivent le service web.

* **Types** Définition des types de données utilisés lors de l'échange sous la forme d'un schéma XML.

* **Message** Permet la définition des structures de données et leur contenu qui vont être échangées dans l'utilisation du service (données à transmettre, valeur de retour).

* **Porttype** Il définit l'interface du service qui permet la définition abstraite d'un ensemble d'opérations offertes par un service Web. Chaque opération fait référence à un message d'entrée et à des messages en sortie. Il y a quatre types d'opération : one-way, request-response, solicit-response et notification.

* **Opérations** Est une description abstraite d'une fonction supportée par le service Web. C'est à dire la description d'une action exposée dans le port.

— La description concrète : Elle désigne la définition de l'implémentation d'un service donné, c'est-à-dire les informations liées à un usage contextuel du service.

* **Binding** : Spécifie une liaison d'un <porttype> à un protocole concretb (SOAP, HTTP, MIME, . . .). Un <porttype> peut avoir plusieurs liaisons.

* **Service** : Cet élément précise les informations complémentaires nécessaires pour invoquer le service. Il contient plusieurs éléments 'ports', contenant chacun un nom, une URL de point d'accès et une référence à une liaison donnée.

* **Port** : Adresse réseau d'une liaison définissant le point d'accès associé à un service.

Inconvénients de WSDL

- Il est difficile d'apprentissage, car c'est un langage de balises.
- Il n'est pas extensible et il est peu expressif.
- Il ne définit pas la logique des composants du service, donc ne supporte pas la comparaison de service.

1.5.3 UDDI (Universal Description Discovery and Integration)

UDDI, a été conçu en 2000 à l'initiative d'un ensemble d'industriels (Ariba, IBM, Microsoft), en vue de devenir le registre standard de la technologie des services Web. Pour convenir à la technologie des services Web, les services référencés dans UDDI sont accessibles par l'intermédiaire du protocole de communication SOAP, et la publication des informations concernant les fournisseurs et les services doit être spécifiée en XML afin que la recherche et l'utilisation soient faites de manière dynamique et automatique. UDDI est une spécification définissant la manière de publier et de découvrir les services Web sur un réseau.

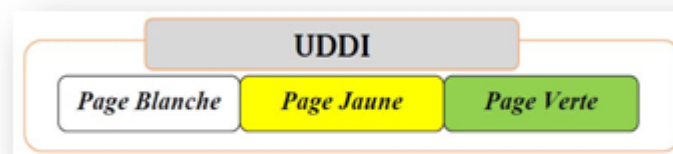


FIGURE 1.7 – les trois types de l'annuaire UDDI

Les informations qu'il contient peuvent être séparées en trois types :

Les pages blanches incluant l'adresse, le contact et les identifiants relatifs aux services Web.

Les pages jaunes identifiant les secteurs d'affaires relatifs aux services Web.

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

1.6 Conclusion

L'architecture Orientée Services à imposer un nouveau style de construction des applications informatiques distribués. Ce style est caractérisé par plusieurs avantages dont l'interopérabilité est l'avantage primordiale visé. Dans ce chapitre, on a présenté les éléments de cette architecture, les technologies sous-jacentes des services web telles que SOAP, WSDL, et UDDI. On a vu également, les différents types de composition de services.

Chapitre 2

Logique Floue

2.1 Introduction

La logique classique joue un rôle immense dans différents domaines, sa structure peut être exprimée des faits qu'avec "vrai" ou "faux" qui limite son champ d'action dans des techniques et des applications comme l'Intelligence Artificielle, la prise de décision des données incertaines, etc. qui veulent imiter le raisonnement et l'esprit humain. C'est à dire des techniques qui s'appuient sur l'incertitude pour leur bon fonctionnement. La logique floue est justement conçue pour régler ce problème, pour permettre la caractérisation des éléments de façon "graduelle". Notre perception du monde réel est imprégnée de concepts qui n'ont pas de limites clairement définies, par exemple, beaucoup, grands, beaucoup plus grands que les jeunes, etc. ne sont vrais que dans une certaine mesure et ils sont faux à un certain degré. Ces concepts peuvent être appelés concepts flous ou vagues, un cerveau humain travaille avec eux, tandis que les ordinateurs ne peuvent pas le faire. Les langages naturels, dont le niveau est beaucoup plus élevé que les langages de programmation, sont flous alors que les langages de programmation ne le sont pas [28]. L'utilité des ensembles flous réside dans leur capacité à modéliser des données incertaines ou ambiguës, si souvent rencontrées dans la vie réelle.

Le terme de logique floue a été introduit avec la proposition de 1965 de la théorie des ensembles flous par Lotfi Zadeh en 1965 à l'université de Berkeley comme extension de la logique booléenne en se basant sur sa théorie mathématique des ensembles flous, qui est une généralisation de la théorie des ensembles classiques. En introduisant la notion de degré dans la vérifi-

cation d'une condition, permettant ainsi à une condition d'être dans un autre état que vrai ou faux, la logique floue confère une flexibilité très appréciable aux raisonnements qui l'utilisent, ce qui rend possible la prise en compte des imprécisions et des incertitudes [29].

La logique floue n'est pas une logique imprécise, mais une logique qui s'adapte à l'humain en laissant une place entre la certitude de la vérité et la certitude du faux. Il ressemble au raisonnement humain dans l'utilisation d'informations approximatives et incertaines pour prendre des décisions [30].

Cette logique ne consiste pas à être précis dans les affirmations, mais au contraire répondre à des propositions vagues, nécessitant une certaine incertitude. Par exemple, en logique classique, à la question : Est-ce que cette personne est grande? On ne peut répondre que par vrai, si c'est le cas, ou faux dans le cas contraire. Avec la logique floue, on peut représenter les cas où la personne est très petite, moyennement petite, normale, pas très grande, grande, etc [31].

Un des intérêts de la logique floue pour formaliser le raisonnement humain est que les règles sont énoncées en langage naturel. Elle est devenue un outil important pour le nombre d'applications différentes allant du contrôle du système d'ingénierie à l'intelligence artificielle [32].

2.2 Historique de la logique floue

Depuis longtemps l'homme cherche à maîtriser les incertitudes et les imperfections inhérentes à sa nature. La première réelle manifestation de la volonté de formaliser la prise en compte des connaissances incertaines fut le développement de la théorie des probabilités à partir du XVII^e siècle. Mais les probabilités ne peuvent maîtriser les incertitudes psychologiques et linguistiques. On a donc assisté aux développements des théories de probabilité subjective (dans les années 50) puis de l'évidence (dans les années 60).

Puis la Logique Floue est apparue en 1965 à Berkeley dans le laboratoire de Lotfi Zadeh avec la théorie des sous-ensembles flous puis en 1978 avec la théorie des possibilités. Ces deux théories constituent aujourd'hui ce que l'on appelle Logique Floue.

La Logique Floue permet la formalisation des imprécisions dues à une connaissance globale d'un système très complexe et l'expression du comportement d'un système par des mots. Elle permet donc la standardisation de la description d'un système et du traitement de données

aussi bien numériques qu'exprimées symboliquement par des qualifications linguistiques [33].

2.3 Logique floue vs logique classique

Dans la logique classique, les variables gérées sont Booléennes. C'est à dire qu'elles ne prennent que deux valeurs 0 ou 1. La logique floue a pour but de raisonner à partir de connaissances imparfaites qui opposent résistance à la logique classique. Pour cela la logique floue se propose de remplacer les variables booléennes par des variables flous [33].

2.4 Y a-t-il un besoin de logique floue

La logique floue a un niveau de généralité beaucoup plus élevé que la logique bivalente. C'est la généralité de la logique floue qui sous-tend une grande partie de ce que la logique floue a à offrir. Parmi les contributions importantes de la logique floue sont les suivants [34] :

— FL-généralisation. Toute théorie basée sur la logique bivalente, T, peut être généralisée FL, et donc améliorée, en ajoutant à T des concepts et des techniques tirés de la logique floue. Exemples : contrôle flou, programmation linéaire floue, théorie des probabilités floues et topologie floue.

— Variables linguistiques et règles if-then floues. Le formalisme des variables linguistiques et des règles if-then floues est, en effet, un puissant langage de modélisation largement utilisé dans les applications de la logique floue. Fondamentalement, le formalisme sert de moyen de synthèse et de compression de l'information par l'utilisation de la granulation.

— Précision co-intensive. La logique floue a un grand pouvoir de précision. Ce pouvoir est nécessaire pour la formulation de définitions communes de concepts scientifiques et la formalisation conjointe de domaines centrés sur l'homme tels que l'économie, la linguistique, le droit, la résolution de conflits, la psychologie et la médecine

— NL-Calcul (computing with words). La logique floue sert de base au calcul NL, c'est-à-dire au calcul avec des informations décrites en langage naturel. NL-Calcul est directement lié à la mécanisation de la compréhension du langage naturel et au calcul avec des probabilités imprécises. Plus généralement, le calcul NL est nécessaire pour traiter l'incertitude de second

ordre, c'est-à-dire l'incertitude sur l'incertitude ou l'incertitude pour le court terme.

2.5 Définition de la logique Floue

Plusieurs définitions ont été abordées pour le mot logique floue. dans la suite, nous allons donner quelques définitions :

Définition de Zadeh : La logique floue s'intéresse aux principes formels du raisonnement approximatif, avec un raisonnement précis considéré comme limitant. il vise à modéliser le mode de raisonnement imprécis qui joue un rôle essentiel dans la capacité humaine remarquable à prendre des décisions dans un environnement d'incertitudes et d'imprécisions [34].

Définition de Wikipédia : La logique floue est une forme de logique de grande valeur dans laquelle les valeurs de vérité des variables peuvent être n'importe quel nombre réel entre 0 et 1. En revanche, dans la logique booléenne, les valeurs de vérité des variables ne peuvent être que la valeur entière 0 ou 1. La logique floue a été utilisée pour gérer le concept de vérité partielle, où la valeur de la vérité peut varier entre complètement vrai et complètement faux. En outre, lorsque des variables linguistiques sont utilisées, ces degrés peuvent être gérés par des fonctions spécifiques (appartenance).

une autre définition : La logique floue est une méthodologie de résolution de problèmes qui fournit un moyen simple de tirer des conclusions définitives à partir d'informations vagues et imprécises [36].

2.6 Domaines d'application :

La majorité des premières applications réussies de la logique floue ont été mises en oeuvre au Japon. La première application notable était sur le train à grande vitesse à Sendai, dans lequel la logique floue pouvait améliorer l'économie, le confort et la précision du trajet. Il a également été utilisé en reconnaissance des symboles écrits à la main dans les ordinateurs de poche de Sony, L'aide au vol pour les hélicoptères, le contrôle des systèmes de métro afin d'améliorer le confort de conduite, la précision de l'arrêt et l'économie d'énergie, la consommation de carburant améliorée pour les automobiles, le contrôle à un bouton pour les machines à laver, le

contrôle automatique du moteur pour les aspirateurs avec reconnaissance de l'état de la surface et le degré de souillure et les systèmes de prédiction pour la reconnaissance précoce des tremblements de terre à travers l'Institut de sismologie Bureau de météorologie au Japon.

La logique floue a potentiellement de nombreuses applications dans tous les domaines, les principaux domaines d'applications de logique floue sont [35] :

- Contrôle automatique des vannes de barrage pour centrales hydroélectriques (Tokyo Electric Power.)
- Contrôle simplifié des robots (Hirota, Fuji Electric, Toshiba, Omron)
- Caméra visant la télédiffusion d'événements sportifs (Omron)
- Contrôle efficace et stable des moteurs de voitures (Nissan)
- Régulateur de vitesse pour automobiles (Nissan, Subaru)
- Substitution d'un expert pour l'évaluation des activités boursières (Yamaichi, Hitachi)
- Planification optimisée des horaires d'autobus (Toshiba, NipponSystem, Keihan-Express)
- Système d'archivage pour documents (Mitsubishi Elec.)
- Système de prévision pour la reconnaissance précoce des tremblements de terre (Bureau de sismologie de métrologie, Japon)
- Technologie de la médecine : diagnostic du cancer (École de médecine de Kawasaki)
- Reconnaissance des motifs en images avec des caméras vidéo (Canon, Minolta)
- Commande automatique des moteurs pour aspirateurs avec reconnaissance de l'état de surface et du degré de salissure (Matsushita) Commande de rétro-éclairage pour caméscopes (Sanyo)

2.7 Théorie de la logique floue

La logique floue repose sur plusieurs théories :

2.7.1 Les sous-ensembles flous

La logique floue repose sur la théorie des ensembles flous, qui sont une généralisation de la théorie des ensembles classiques. Les ensembles classiques sont également appelés ensembles

nets, par opposition à flou, et de même la logique classique est également appelée logique booléenne ou binaire [33]. Un sous ensemble flou est caractérisé par une fonction d'appartenance u dans l'intervalle des nombres réels $[0, 1]$ (degré d'appartenance qui est l'extension de la fonction caractéristique d'un sous-ensemble classique).

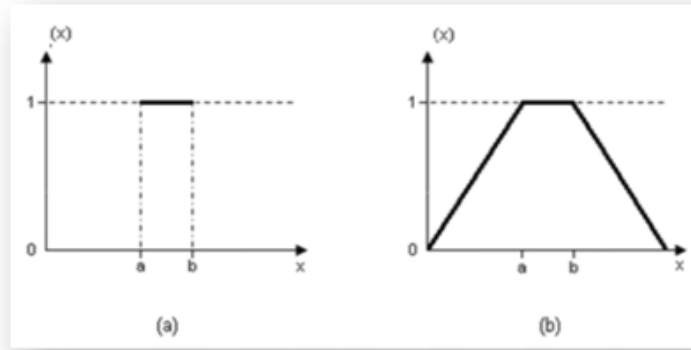


FIGURE 2.1 – Fonction d'appartenance caractérisant un ensemble classique (a) et d'un ensemble flou (b)

2.7.2 Les opérations de base sur les sous-ensembles flous

La théorie mathématique sur les sous-ensembles flous définit de nombreuses opérations sur ces sous-ensembles et sur les fonctions d'appartenances qui rendent ces notions utilisables. Nous ne présentons ici que les opérations de base de cette théorie. Si A et B sont deux sous-ensembles flous et $u(A)$ et $u(B)$ leur fonction d'appartenance, on définit [33] :

Dans ce mémoire, nous citons seulement quelques opérations :

- Le complémentaire de A , par la fonction d'appartenance : $u(A) = 1 - u(A)$

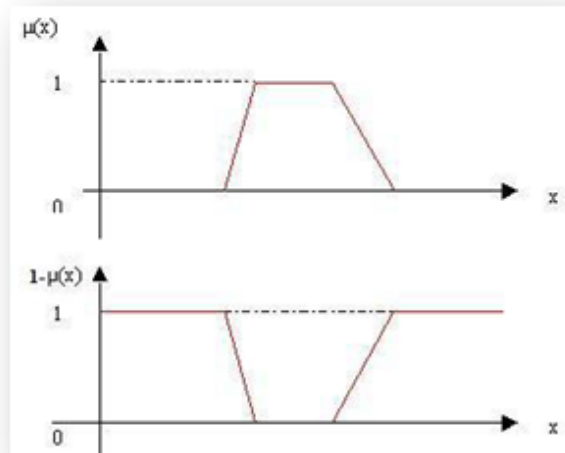


FIGURE 2.2 – Fonction d'appartenances

— L'intersection des sous-ensemble A et B, est donnée par la fonction d'appartenance :

$$(A \cap B) = \min(u(A), u(B))$$

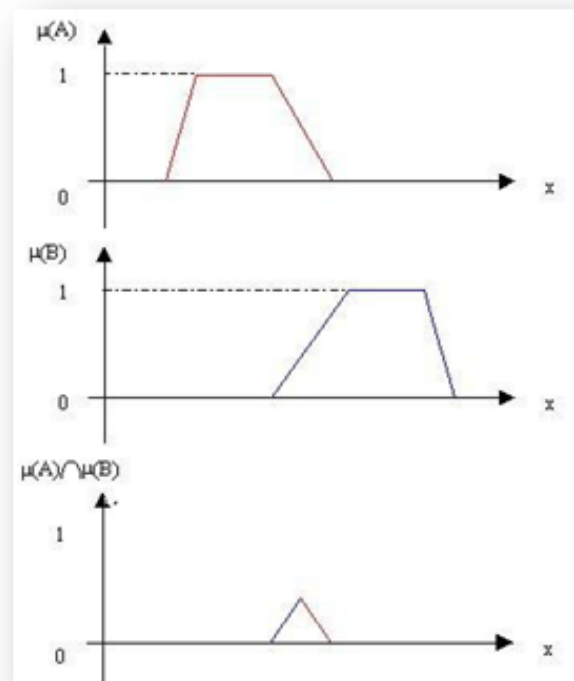


FIGURE 2.3 – Intersection des fonctions d'appartenances

— L'union des sous-ensemble A et B, est donnée par la fonction d'appartenance :

$$(A \cup B) = \max(u(A), u(B))$$

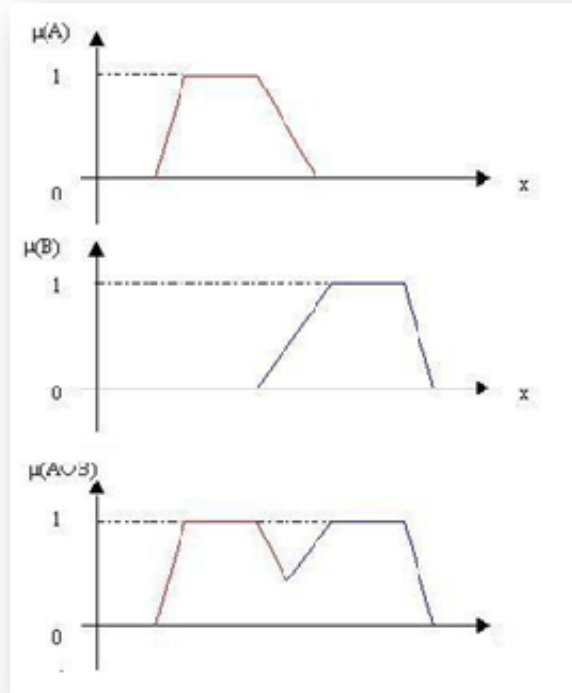


FIGURE 2.4 – Union des fonctions d'appartenances

Ces définitions sont celles qui sont les plus communément utilisées mais parfois, pour certains cas, d'autres sont plus appropriées. Par exemple, l'intersection peut être définie par le produit des fonctions d'appartenance et l'union par la moyenne arithmétique des fonctions d'appartenance. Ces différentes techniques de calcul engendrent une énorme capacité d'adaptation des raisonnements flous.

2.8 Variable linguistique

Le concept des variables linguistiques joue un rôle important dans le domaine de la logique floue. Une variable linguistique comme son nom le suggère, est une variable définie à base de mots ou des phrases au lieu des nombres. En effet, la description d'une certaine situation, d'un phénomène ou d'un procédé contient en général des expressions floues comme quelque, beau-

coup, souvent, chaud, froid, rapide, lent, grand, petit ... etc.. Ce genre d'expressions forme ce qu'on appelle des variables linguistiques de la logique floue [33].

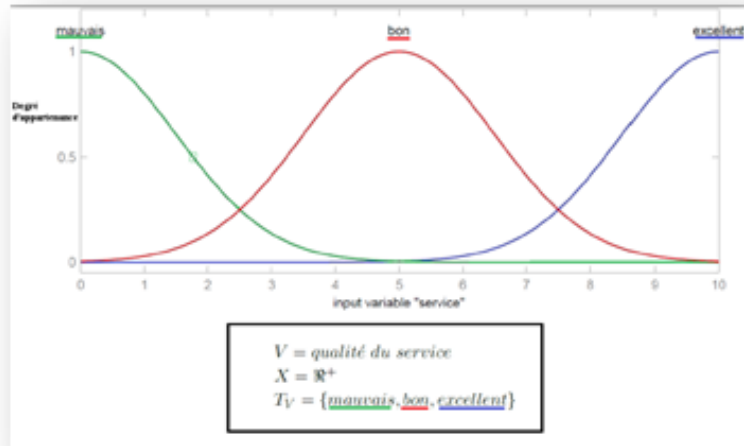


FIGURE 2.5 – Variable linguistique de la qualité du service

Définition des fonctions d'appartenance

Un ensemble flou est défini par sa fonction d'appartenance qui correspond à la notion de fonction caractéristique en logique classique, elle permet de mesurer le degré d'appartenance d'un élément à l'ensemble flou. En toute généralité, une fonction d'appartenance d'un ensemble flou est désignée par $\mu_A(x)$. L'argument x se rapporte à la variable caractérisée, alors que l'indice A indique l'ensemble concerné.

Les fonctions d'appartenance peuvent avoir différentes formes [33] :

a) Fonction d'appartenance triangulaire :

$$\tilde{N} = \begin{cases} 0 & \text{si } x < a \\ \frac{x-a}{b-a} & \text{si } a \leq x \leq b \\ \frac{c-x}{c-b} & \text{si } b \leq x \leq c \\ 0 & \text{si } x > c. \end{cases}$$

ou a, b, c sont des nombres réels et $(a < b < c)$

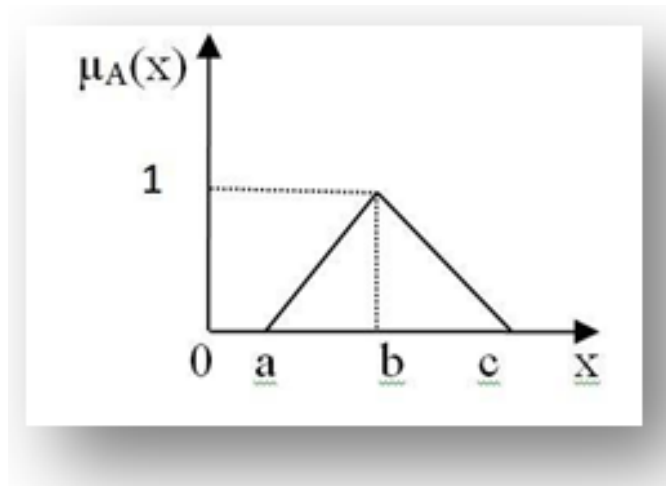


FIGURE 2.6 – Forme triangulaire

b) Fonction d'appartenance trapézoïdale :

$$\tilde{N} = \begin{cases} \frac{x-b}{b-a} & \text{si } a < x \leq b \\ 1 & \text{si } b \leq x \leq c \\ \frac{d-x}{d-b} & \text{si } c \leq x \leq d \\ 0 & \text{autrement.} \end{cases}$$

Où a, b, c et d sont des nombres réels et $(a \leq b \leq c \leq d)$

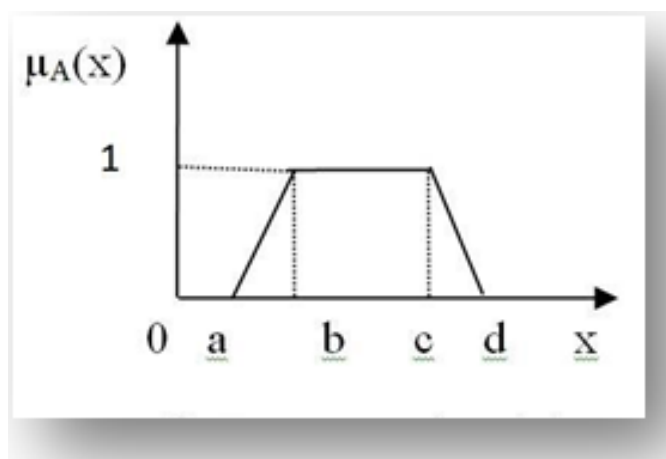


FIGURE 2.7 – Forme trapézoïdale

c) Fonction d'appartenance gaussienne : $u_A(x) = \exp\left(-\frac{1}{2}\left(\frac{x-m}{\sigma}\right)^2\right)$

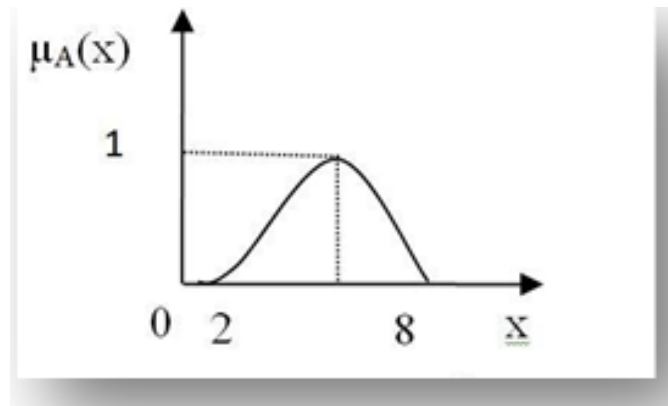


FIGURE 2.8 – Forme gaussienne

B. variable linguistique

Le concept de fonction d'appartenance vu précédemment nous permettra de définir des systèmes flous en langage naturel, la fonction d'appartenance faisant le lien entre logique floue et variable linguistique.

2.9 Système d'inférence Flou :

Un Système d'Inférence Floue (SIF) a comme but de transformer les données d'entrée en données de sortie à partir de l'évaluation d'un ensemble des règles. Les entrées sont issues du processus de fuzzification et l'ensemble de règles normalement sont définies par le savoir faire de l'expert. Un SIF est constitué de trois étapes, Comme le montre la figure suivante :

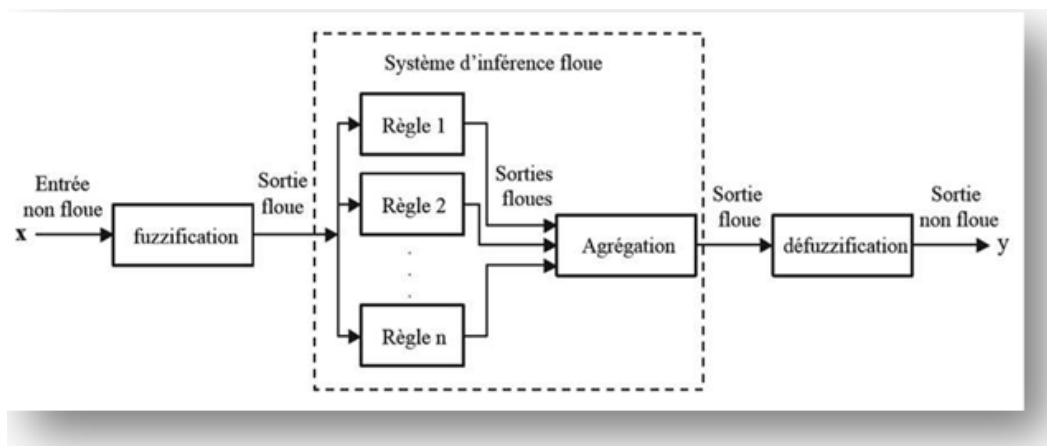


FIGURE 2.9 – Système Inférence Flou

2.9.1 Fuzzification :

qui consiste à caractériser les variables linguistiques utilisées dans le système. Il s'agit donc d'une transformation des entrées réelles en une partie floue définie sur un espace de représentation lié à l'entrée. Cet espace de représentation est normalement un sous-ensemble flou. Durant l'étape de la fuzzification, chaque variable d'entrée et de sortie est associée à des sous-ensembles flous.

2.9.2 Inférence :

consiste à utiliser le moteur d'inférence, qui est un mécanisme permettant de condenser l'information d'un système à travers d'un ensemble de règles définies pour la représentation d'un problème quelconque. Chaque règle délivre une conclusion partielle qui est ensuite agrégée aux autres règles pour fournir une conclusion (agrégation). Les règles constituent le système d'inférence floue, dans la suite de ce chapitre nous donnons une description de règles floues dans un cadre plus formel.

2.9.3 Défuzzification :

qui consiste à caractériser les variables linguistiques utilisées dans le système. Il s'agit donc d'une transformation des entrées réelles en une partie floue définie sur un espace de repré-

sentation lié à l'entrée. Cet espace de représentation est normalement un sous-ensemble flou. Durant l'étape de la fuzzification, chaque variable d'entrée et de sortie est associée à des sous-ensembles flous.

2.10 Conclusion

La logique floue est une approche au raisonnement humain. Dans ce chapitre nous avons vu les étapes et les méthodes les plus répandus pour la réalisation d'un système basé sur la logique floue. Les fonctions d'appartenances sont le coeur de la logique floue, et ces eux qui mettent la différence entre la logique classique (booléen) et la logique floue. L'inférence est ou l'expérience humaine détermine la manière de raisonnement du système. Et l'agrégation c'est ou le système prend le rôle de l'être humain et commence à raisonner et de défuzzifier en suite pour donner des résultats

Chapitre 3

Modélisation

3.1 Introduction

Avec le nombre croissant de services mis à disposition sur Internet et dont la fonctionnalité est similaire, " comment distinguer et sélectionner le meilleur service Web parmi une liste de services web similaires?" devient un problème majeur à résoudre.

La plupart des approches existant dans la littérature sont basées sur la sélection des services Web selon la description fonctionnelle, mais ces services Web ne peuvent pas être en mesure de fournir la même qualité de service.

La sélection des services Web selon les aspects non fonctionnels est basée essentiellement sur les critères de qualité de service (QoS), qui se mesurent à l'aide de plusieurs métriques (Coût, le temps d'exécution, la fiabilité, la disponibilité, et la réputation,...etc) Cependant, ces critères sont imprécis et parfois incertains et ambigus dûs à l'état mental de l'utilisateur et aussi sur le manque d'informations de l'utilisateur sur les données des services Web. La logique floue vient traduire cette évaluation en présentant les critères de qualité de manière adéquate pour l'utilisateur.

L'utilisation de la logique floue permet de supporter une représentation imprécise des contraintes QoS et présenter les préférences de l'utilisateur en tant que propriétés QoS avec une présentation floue car elles sont mieux adaptées à l'interprétation des termes linguistiques.

Le but de notre travail d'une part est de représenter les critères de qualité de services en valeurs linguistiques et d'autre part pouvoir sélectionner d'une façon optimale les services Web en

se basant sur la notion de la logique floue Web qui répondent aux exigences non fonctionnelles des clients.

Nous commençons dans un premier temps par présenter en détail notre solution qui résout le problème de la prise des données linguistiques de l'utilisateur d'une part et d'autre part de sélectionner le service web approprié à retourner au client.

Cette solution consiste en la proposition d'un nouveau modèle de sélection des services Web basé sur la logique floue.

3.2 Qualité de service

La qualité de service peut être définie comme une comparaison entre les attentes du client et ce qui est réellement offert. Elle est définie comme une combinaison de plusieurs critères qui peuvent alors être considérés comme un critère de choix lorsque l'on sélectionne parmi plusieurs services web découverts ceux qui respectent les contraintes imposées.

3.2.1 Paramètres de qualités de services

La sélection des services web à base de Qualités de Services (QoS) consiste à choisir parmi les services Web découverts, ceux qui répondent mieux aux exigences de l'utilisateur sur la base des besoins non fonctionnels QoS.

Il existe des mesures pour quantifier un service Web. Ces dernières sont présentés par la qualité de service. De plus, les QoS représentent l'aspect non-fonctionnel d'un service. Ces valeurs permettent la sélection des services pertinents aux demandes des utilisateurs.

Plusieurs critères de qualité de service sont décrits dans différents travaux et les propriétés de QoS les plus représentatives sont présentées comme suit [6] :

— **Disponibilité** : la probabilité que les ressources, les services sont disponibles pour les parties autorisées en tout temps pour l'utilisation ou le pourcentage de temps que le service fonctionne.

— **Fiabilité** : La capacité d'un service à exécuter les fonctions requises dans des conditions spécifiées pendant une période donnée.

— **Débit (Throughput)** : Le nombre de demandes de service terminées sur une période donnée.

— **Temps de réponse** : la durée entre l'envoi d'une demande par un utilisateur de service et la réception d'une réponse.

— **Sécurité** : L'aspect qualité du service Web pour assurer la confidentialité et la non répudiation en authentifiant les parties impliquées, en chiffrant les messages et en assurant le contrôle d'accès. Le fournisseur de services peut avoir différentes approches de sécurité en fonction du demandeur de service.

— **Accessibilité** : c'est l'aspect qualité d'un service qui représente le degré de capacité à servir une requête de service Web. Il peut être exprimé comme une mesure de probabilité indiquant le taux de réussite ou la probabilité d'une instanciation de service réussie à un moment donné.

— **Coût d'exécution** : Le coût à dépenser pour exécuter un service Web, ce coût peut être fourni par le fournisseur de services.

— **Réputation** : Le taux moyen du service rapporté par le client, il est basé sur le service client; cela dépend principalement de l'expérience de l'utilisateur final de l'utilisation du service. Différents utilisateurs peuvent avoir des opinions divergentes sur le même service, généralement; le client évalue le service en le classant avec une valeur.

Dans notre proposition, nous allons opter pour l'utilisation de ces critères non fonctionnels de services Web dans le processus de sélection.

3.3 Une approche de sélection des services Web basée sur la logique floue

Cette solution consiste en un nouveau modèle de sélection de services Web basé sur la logique floue. L'utilisation de la logique floue dans notre approche est primordiale, avec cette logique, nous présentons les valeurs de critères de qualité de service relatives aux services Web en valeurs linguistiques : par exemple au lieu de donner une seule valeur numérique au paramètre C_1 , on lui attribue par exemple : mauvais, moyen, bon et excellent.

Exemple : si on a le critère coût, cette dernière peut être présentée de cette manière :

Coût	
Terme linguistique	Nombre flou triangulaire
Moins cher	[0 1 5]
Acceptable	[4 7 10]
Cher	[9 12 20]

TABLE 3.1 – Représentation du paramètre coût en flou

3.4 Architecture générale

L'architecture de notre approche, qui exprime l'extension de l'architecture de base SOA, en ajoutant une nouvelle composante **Découverte** entre le client et l'annuaire **UDDI** de telle façon la phase de la recherche des services similaires serait donc simple et facile.

Initialement, le fournisseur publie les services web avec leurs QoS dans le registre UDDI qui dispose d'une base de données destinée aux QoS.

L'extraction de QoS est effectuée pour chaque service Web afin de produire une matrice de décision comportant les SWs et leurs QoS correspondant, puis ces QoS sont transformés en valeurs floues en utilisant les fonctions d'appartenance par exemple : fonction triangulaire.

Cette matrice de décision floue est ensuite passée au module de classement flou qui applique la méthode TOPSIS floue sur ces données pour classer les services web et donner à la fin des scores qualifiant chacun d'eux.

Le meilleur service qui sera servi au demandeur correspondant à la plus grande valeur du score.

La figure suivante (Figure 3.1) représente l'architecture proposée :

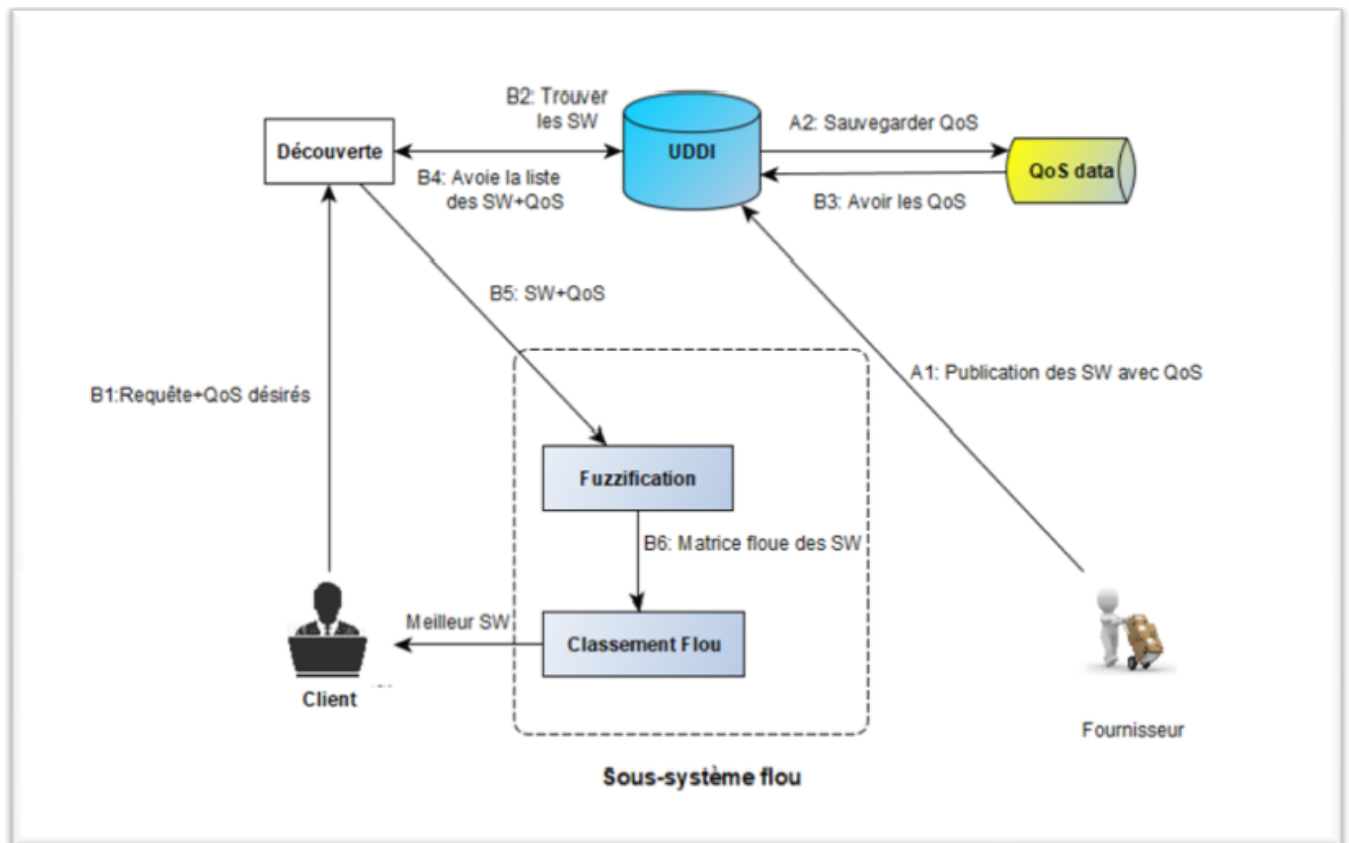


FIGURE 3.1 – Architecture de l'approche proposée

3.5 Description des différents acteurs

Nous allons décrire en détail l'aspect structurel et fonctionnel des différents acteurs présents dans cette approche.

* **Client** : qui souhaite avoir un service en lançant une requête au registre UDDI via le module Découverte.

* **Découverte** : ce module s'occupe de la requête émise par le client en l'envoyant vers le registre UDDI en récupérant les SW similaires.

Les tâches effectuées par ce module sont :

1. Obtenir la requête de l'utilisateur.
2. Trouver la liste de la recherche des services similaires à partir du registre UDDI.

3. Extraire les propriétés non fonctionnelles (QoS) du service.
4. Passer chaque QoS au sous-système flou.

*** Le registre UDDI : Universal Description Discovery and Integration**

Est un standard défini par OASIS qui est devenu le registre standard de la technologie des services Web. Il a pour but de permettre d'automatiser les communications entre fournisseurs et clients et de gérer les méta-données des services.

UDDI est une spécification qui définit les mécanismes qui permettent aux entreprises de publier leurs services et de découvrir les détails techniques d'un service Web (le WSDL) ainsi que les informations orientées métier, et d'interagir avec d'autres services via le Web. Ainsi, lorsque l'on veut mettre à disposition un nouveau service, on crée un fichier appelé Business Registry qui décrit le service en utilisant un langage dérivé d'XML suivant les spécifications UDDI. Il se comporte lui-même comme un service Web dont les méthodes sont accessibles par l'intermédiaire du protocole de communication SOAP.

Les opérations pouvant être effectuées par UDDI sont :

– La recherche : Qui se fait par le nom et/ou par des mots clés qui fournit le service désiré ou par la catégorie d'entreprise ou par la localisation de l'entreprise pour une éventuelle interaction.

– La navigation.

– L'ajout et la suppression de services.

*** Le fournisseur :**

Le Fournisseur du service Correspond à la personne ou à l'organisation propriétaire du service qui réalisera effectivement le service demandé. Il publie son service en fournissant la description des services (les propriétés fonctionnel et non-fonctionnel) au format WSDL dans l'annuaire de services afin que les clients puissent découvrir et accéder au service. Alors, il représente l'environnement d'hébergement et d'exécution du service. Il est constitué de trois couches de bases :

-**La couche de données** : contient les différentes bases de données utilisées par le service.

-**La couche applicative** : c'est la plateforme de développement qui assure l'exécution du service web.

-**La couche de description** : elle expose les fonctionnalités du service via un fichier WSDL [4].

3.6 Fonctionnement de l'approche

Le fonctionnement détaillé de notre approche peut être décrit par la figure (3.2) suivante :

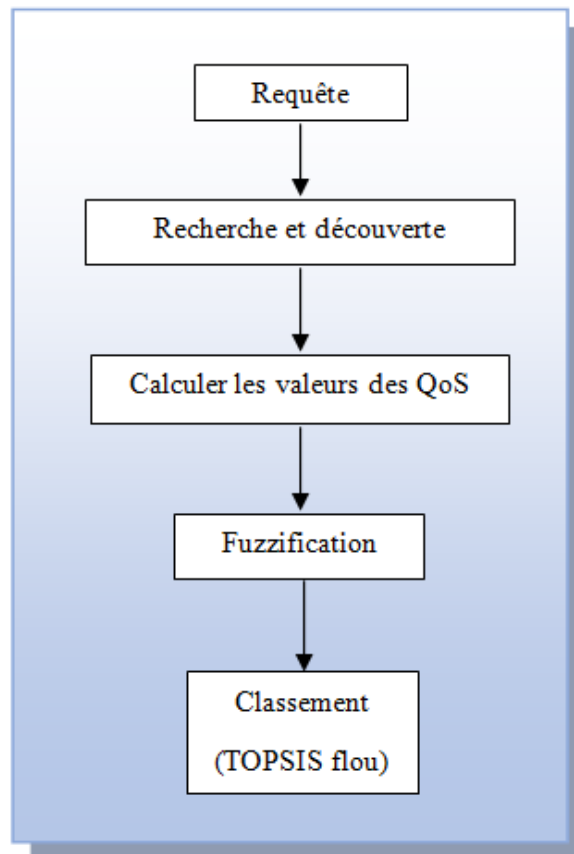


FIGURE 3.2 – Fonctionnement de l'approche

Sous-système flou

Le sous-système flou est le cœur de notre solution. Il se constitue principalement de deux modules : Le module Fuzzification et le module Classement flou. Il a comme entrée les services web avec leur critères de qualités de services et en sortie le meilleur service à fournir au client. Ce sous module peut être décortiqué comme suit (Figure 3.3) :

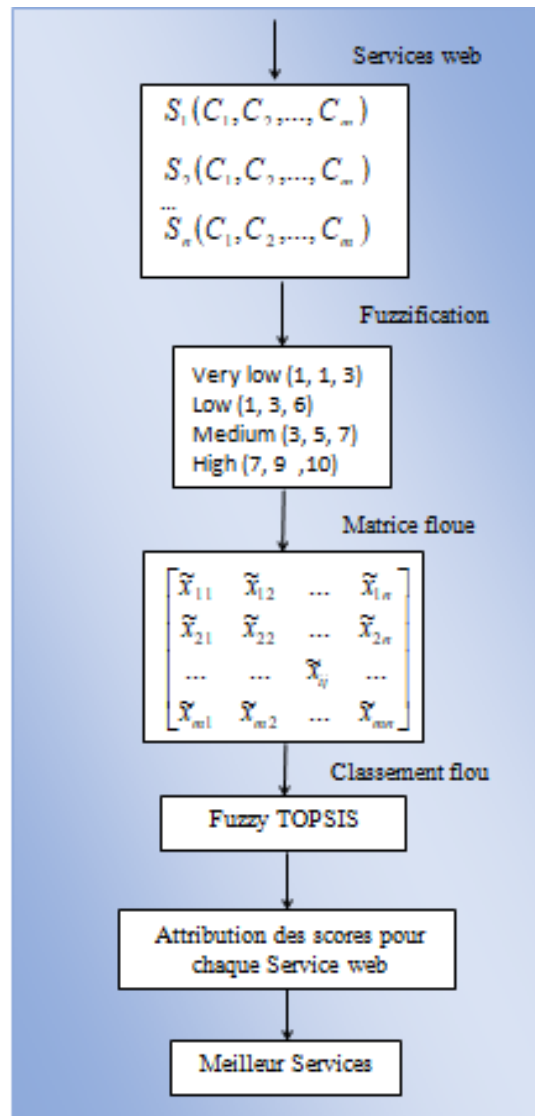


FIGURE 3.3 – Comportement du sous-système flou

* **Module Fuzzification** : Le but de la fuzzification est de transformer les données numériques en variables linguistiques. En choisissant un type de fonction d'appartenance (par exemple, méthode d'appartenance triangulaire), traduire les valeurs croisées des paramètres QoS en valeurs floues. Le résultat est une matrice floue des services web composé de (S_1, S_2, \dots, S_m) services web et de (C_1, C_2, \dots, C_n) qualité de service qui peut être représentée par la figure suivante :

$$\tilde{D} = \begin{matrix} & c_1 & c_2 & \dots & c_n \\ \begin{matrix} s_1 \\ s_2 \\ \dots \\ s_m \end{matrix} & \begin{pmatrix} \tilde{x}_{11} & \tilde{x}_{12} & \dots & \tilde{x}_{1n} \\ \tilde{x}_{21} & \tilde{x}_{22} & \dots & \tilde{x}_{2n} \\ \dots & \dots & \dots & \dots \\ \tilde{x}_{m1} & \tilde{x}_{m2} & \dots & \tilde{x}_{mn} \end{pmatrix} \end{matrix}$$

Fuzzification

Le but de l'étape de fuzzification est de transformer une donnée numérique en une variable linguistique. Pour cela, nous devons créer des fonctions d'appartenance qui définissent le degré d'appartenance d'une donnée numérique à une variable linguistique. Les entrées de la fuzzification sont les paramètres de qualité de service (QoS) dont les valeurs sont quantitatives.

Pour chaque critère, nous pouvons créer plusieurs fonctions d'appartenance à savoir : Fonction triangulaire, trapézoïdale ou gaussienne. Si les services Web que nous voulons tester sont publiés avec les critères : fiabilité, disponibilité et temps de réponse : chacune de ces donnée saura plusieurs fonctions d'appartenance. Par exemple : Si nous voulons transformer la fiabilité en variable linguistique, nous pouvons trouver plusieurs valeurs linguistiques qualifiant ces données numériques : faible, moyenne et élevée.

Même principe pour les deux autres critères : la disponibilité peut être présentée par les valeurs linguistiques suivantes : faible, moyenne et élevée et le critère de temps de réponse peut être présenté par les valeurs linguistiques : petit, moyen et long.

En bref, cette première étape consiste à transformer les variables d'entrée en variables linguistiques :

- Pour chaque variable, on définit dans un premier temps l'univers du discours (i.e. la plage de valeurs que peut prendre la variable) ;
- La variable est ensuite découpée en catégories appelées valeurs linguistiques ;
- Une fonction (allant de 0% à 100%) permettant de définir pour chaque valeur son pourcentage de véracité à l'affirmation : « l'observation est dans telle catégorie » est affectée à chaque catégorie.

Dans notre solution, nous avons opté d'utiliser la représentation Triangulaire floue pour représenter les données linguistiques de QoS car la traduction de l'expertise humaine vers ce type de nombre flou est facile. Les tableaux suivants est un exemple qui montre la présentation des

critères en valeurs linguistiques :

Terme linguistique	Nombre flou triangulaire
Response time	
Low	[20 20 450]
Medium	[300 600 900]
High	[700 900 1200]
Very high	[1000 1300 1500]

TABLE 3.2 – Paramètre Response time en représentation floue

Terme linguistique	Nombre flou triangulaire
Price	
cheap	[2 2 8]
Medium	[6 10.5 15]
expensive	[12.8 16 20]

TABLE 3.3 – Paramètre Price en présentation Floue

Terme linguistique	Nombre flou triangulaire
Reliability	
Low	[0 0 25]
Fair	[15 30 50]
Good	[35 50 70]
Very good	[60 80 100]

TABLE 3.4 – Paramètre Reliability en présentation Floue

Terme linguistique	Nombre flou triangulaire
Availability	
Low	[0 0 25]
Fair	[15 30 50]
Good	[35 50 70]
Very good	[60 80 100]

TABLE 3.5 – Paramètre Availability en présentation Floue

Ces paramètres de QoS peuvent être représentés graphiquement selon la fonction d'appartenance triangulaire par les figures suivantes :

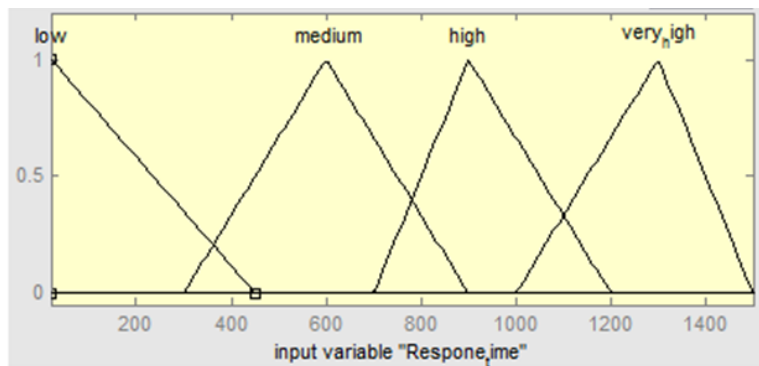


FIGURE 3.4 – Variable linguistique Response-time

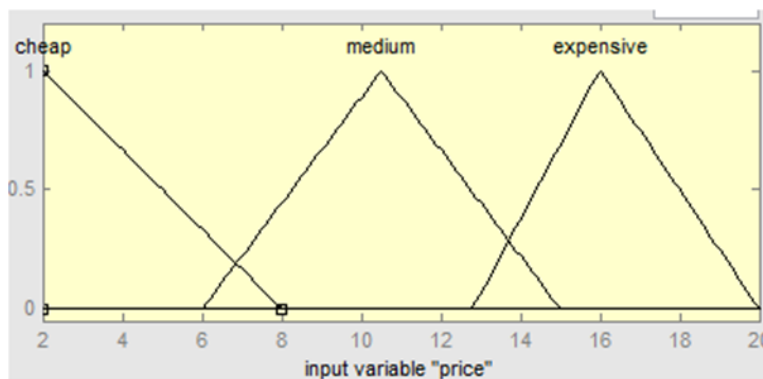


FIGURE 3.5 – Variable linguistique Price

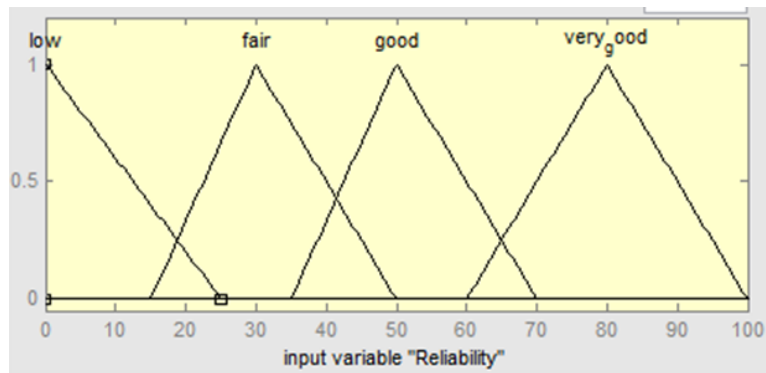


FIGURE 3.6 – Variable linguistique Reliability

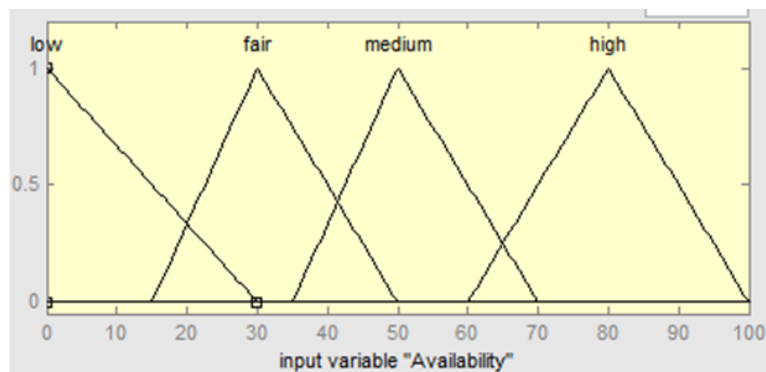


FIGURE 3.7 – Variable linguistique Availability

* **Module Classement flou** : il s'occupe du classement des services similaires. Il possède un module de classement qui est chargé de sélectionner le meilleur service en classant les services similaires en utilisant l'algorithme « TOPSIS flou ».

Méthode TOPSIS floue : TOPSIS (The Technique for Order of Preference by Similarity to Ideal Solution) est une méthode d'analyse décisionnelle multicritères, développée à l'origine par Ching-Lai Hwang et Yoon en 1981, puis développée par Yoon en 1987 et par Hwang, Lai et Liu en 1993. TOPSIS basé sur le concept que l'alternative choisie devrait avoir la distance géométrique la plus courte par rapport à la solution idéale positive (PIS) et la plus grande distance géométrique par rapport à la solution idéale négative (NIS) [8]. Elle est la plus connue pour résoudre les problèmes de MCDM.

Chen [40] a étendu TOPSIS aux Nombres Triangulaires flous. Il a introduit une méthode de sommet pour calculer la distance entre deux Nombre Triangulaires flous, comme suit :

$$d(\tilde{x}, \tilde{y}) = \sqrt{\frac{(a_1 - a_2)^2 + (b_1 - b_2)^2 + (c_1 - c_2)^2}{3}}$$

La méthode de TOPSIS floue est la suivante [7][8] :

1. Affectation des notations aux critères et aux services et calcul de leurs notations floues agrégées. Toutes les évaluations des experts sont exprimés en nombres triangulaire flous. Nous supposons que nous avons un groupe de décision de k membres et de n services (S_1, S_2, \dots, S_n) qui sont évalué selon le critère $C_j (j = 1, 2, \dots, m)$. l'évaluation de l'expert K_{th} pour le service S_i selon le critère C_j est noté : $\tilde{X}_{ij}^k = (a_{ij}^k, b_{ij}^k, c_{ij}^k)$ le poids du critère C_j est noté $W_{ij}^k = (w_{j1}^k, w_{j2}^k, w_{j3}^k)$ Sur la base de la notation floue, les valeurs floues agrégées x_{ij} des alternatives pour chaque critère sont données par

$$X_{ij} = (a_{ij}, b_{ij}, c_{ij}) \text{ où } a_{ij} = \min_k \{a_{ij}^k\}, b_{ij} = \frac{1}{k} \sum_{k=1}^k b_{ij}^k \text{ et } c_{ij} = \max_k \{c_{ij}^k\}$$

Les poids flous agrégés \tilde{w}_{ij} pour chaque critère est calculé par :

$$\tilde{w}_j = (w_{j1}, w_{j2}, w_{j3}). w_{j1} = \min_k \{w_{j1}^k\}, w_{j2} = \frac{1}{k} \sum_{k=1}^k w_{j2}^k \text{ et } w_{j3} = \max_k \{w_{j3}^k\}$$

2. Calculer la matrice de décision floue normalisée. La matrice floue normalisée \tilde{R} est donnée par :

$$\text{Pour les critères positives : } \tilde{r}_{ij} = \left(\frac{a_{ij}}{c_j^*}, \frac{b_{ij}}{c_j^*}, \frac{c_{ij}}{c_j^*} \right) \text{ and } c_j^* = \max_i \{c_{ij}\}$$

$$\text{pour les critères négatives : } \tilde{r}_{ij} = \left(\frac{a_j^-}{c_{ij}}, \frac{a_j^-}{b_{ij}}, \frac{a_j^-}{c_{ij}} \right) \text{ and } c_j^* = \min_i \{a_{ij}\}$$

3. Calculer la matrice de decision normalisée pondérées

$$\tilde{V} = [\tilde{v}_{ij}]_{m \times n} \text{ where } \tilde{v}_{ij} = \tilde{r}_{ij} \cdot \tilde{w}_{ij}$$

4. Calculer la solution idéale floue positive (FPIS) and la solution idéale floue négative (FNIS) :

$$\text{Le FPIS est calculé par : } S^* = (\tilde{v}_1^*, \tilde{v}_2^*, \dots, \tilde{v}_3^*) \text{ where } \tilde{v}_j^* = \max_i \{v_{ij3}\}$$

$$\text{Le FNIS est calculé par : } S^- = (\tilde{v}_1^-, \tilde{v}_2^-, \dots, \tilde{v}_3^-) \text{ where } \tilde{v}_j^- = \min_i \{v_{ij1}\}$$

5. Calculer la distance entre chaque alternative au FPIS et FNIS :

$$d_i^* = \sum_{j=1}^n d(\tilde{v}_{ij}, \tilde{v}_j^*) \text{ where } i = 1, 2, 3, \dots, m$$

$$d_i^- = \sum_{j=1}^n d(\tilde{v}_{ij}, \tilde{v}_j^-) \text{ où } i = 1, 2, 3, \dots, m$$

$d(\tilde{x}, \tilde{y})$ est la mesure de distance entre les nombres flous \tilde{x} et \tilde{y} .

6. Calculer le coefficient pour chaque service web S_i ; Ce coefficient représente la distance

de la solution idéale positive S^* et la solution idéale négative S^- :

$$CC_i = \frac{d_i^*}{d_i^- + d_i^*}$$

7. Classer les services.

Fonctionnement général du système :

Nous allons montrer le fonctionnement du système à travers des diagrammes de séquence.

Diagrammes de séquence :

Un diagramme de séquence permet de représenter des collaborations entre objets selon un point de vue temporel. Une séquence est une scénarisation théorique d'un cas d'utilisation précis en impliquant toutes les possibilités auxquelles ce dernier peut être confronté. Les objets communiquent en échangeant des messages représentés sous forme de flèches. Aussi, ce diagramme spécifie l'ordre des envois et des réceptions de messages[9]. Dans cette partie, on va présenter différents diagrammes de séquence de notre approche comme suit :

3.7 Diagramme de séquence de Publication des services web

Au début le fournisseur doit publier les services web disponibles avec l'évaluation de leurs critères de QoS au niveau de l'annuaire.

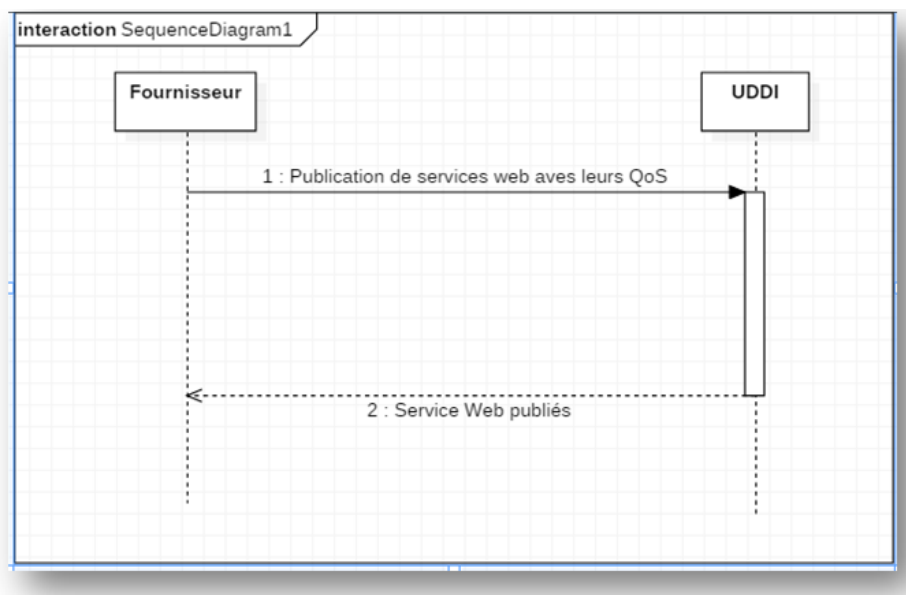


FIGURE 3.8 – Diagramme de séquence de publication des services

3.8 Diagramme de séquence de Sélection de service web

Une fois que nous disposons d'un registre UDDI dont lequel il y a eu des services web publiés, le client pourra ainsi lancer sa requête de recherche de services web. Cette requête est d'abord envoyée au module Découverte qui joue un rôle intermédiaire entre le client et le registre UDDI. Après la recherche dans ce dernier registre, il y a eu une liste de SW similaire qui est retournée à ce module qui fait l'extraction des différents critères de QoS et envoie ensuite cette liste (Service web + Critères) au module de Fuzzification qui traduit les valeurs des QoS en termes linguistiques. Ensuite une matrice floue est envoyée au module Classement flou qui applique la méthode Topsis flou sur la matrice d'entrée pour donner au résultat le meilleur service qui sera remis au client qui l'invoque auprès du fournisseur.

L'ensemble des interactions entre les différents acteurs pour aboutir à la sélection d'un meilleur service peut être décrit par le diagramme de séquence suivant :

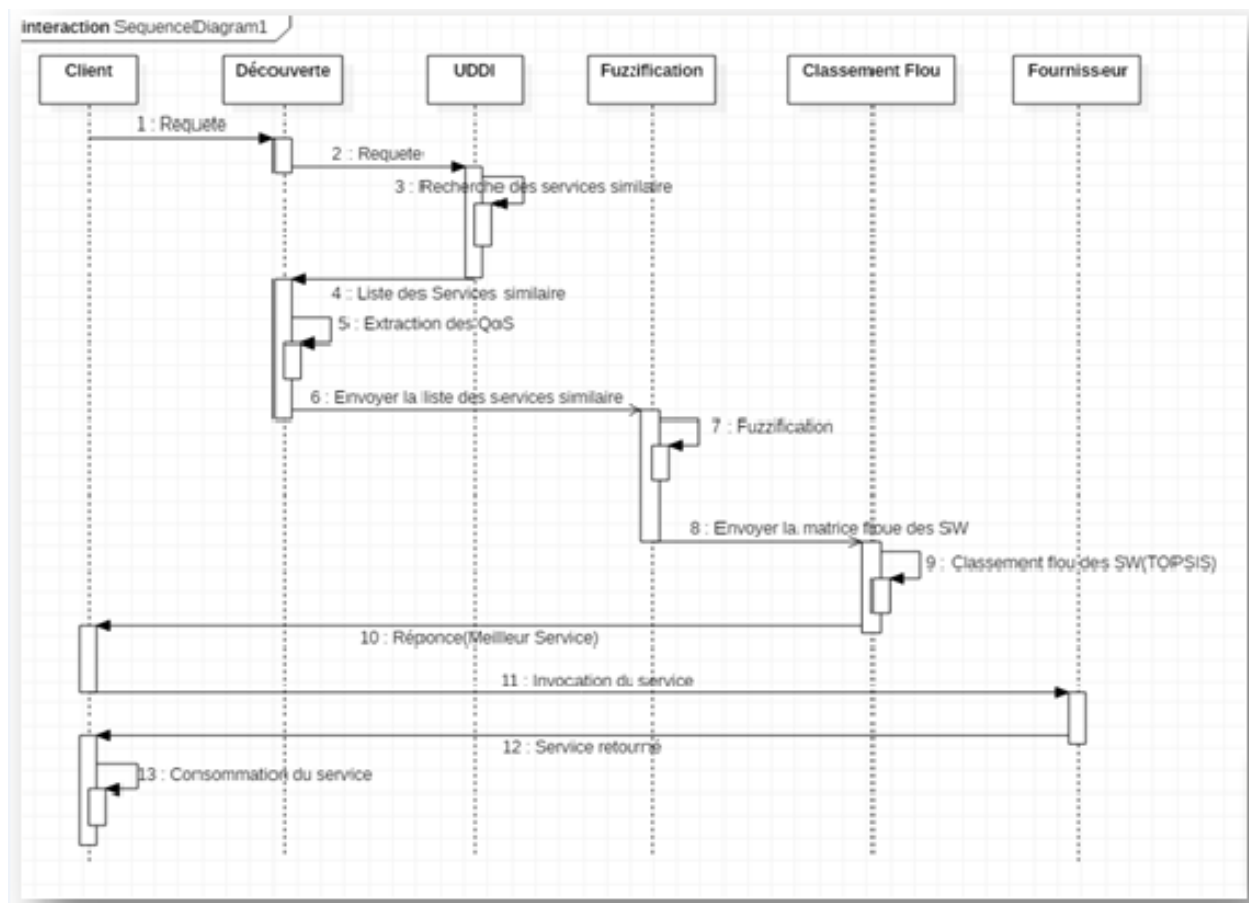


FIGURE 3.9 – Diagramme de séquence de sélection de services web.

3.9 Conclusion

Dans ce chapitre nous avons présenté notre approche proposée pour la sélection des services web basée sur la logique floue. Pour le classement des services web similaires, nous avons intégré une méthode de décision multicritères(MCDM) TOPSIS Floue. Elle sert à classer l'ensemble des services Web par ordre de choix sur la base d'un ensemble de critères flous. Nous avons exprimé également le fonctionnement de notre proposition en se basant principalement sur les différents diagrammes UML et sur les scénarios de description pour chaque phase. Cette étude conceptuelle présente l'architecture générale de notre approche.

Dans le prochain chapitre, nous allons présenter l'implémentation d'un prototype pour montrer la faisabilité de notre approche.

Chapitre 4

Implémentation

4.1 Introduction

Après avoir présenté en détails notre approche de sélection des services Web en prenant en compte les critères de qualité de service avec l'utilisation de l'algorithme qui aide à la décision multicritère TOPSIS Flou dans le chapitre précédent. Ce présent chapitre sera consacré à la phase d'implémentation, nous allons montrer comment nous avons réalisé et implémenté notre système. Nous commençons par la présentation de l'environnement logiciel utilisé, à travers la présentation des outils et du langage de programmation. Pour terminer, nous allons présenter les interfaces graphiques en décrivant les différentes fonctionnalités de notre application.

4.2 Environnement de développement

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

4.2.1 Environnement matériel et logiciel

Le matériel utilisé est un ordinateur portable décrit dans la figure suivante :

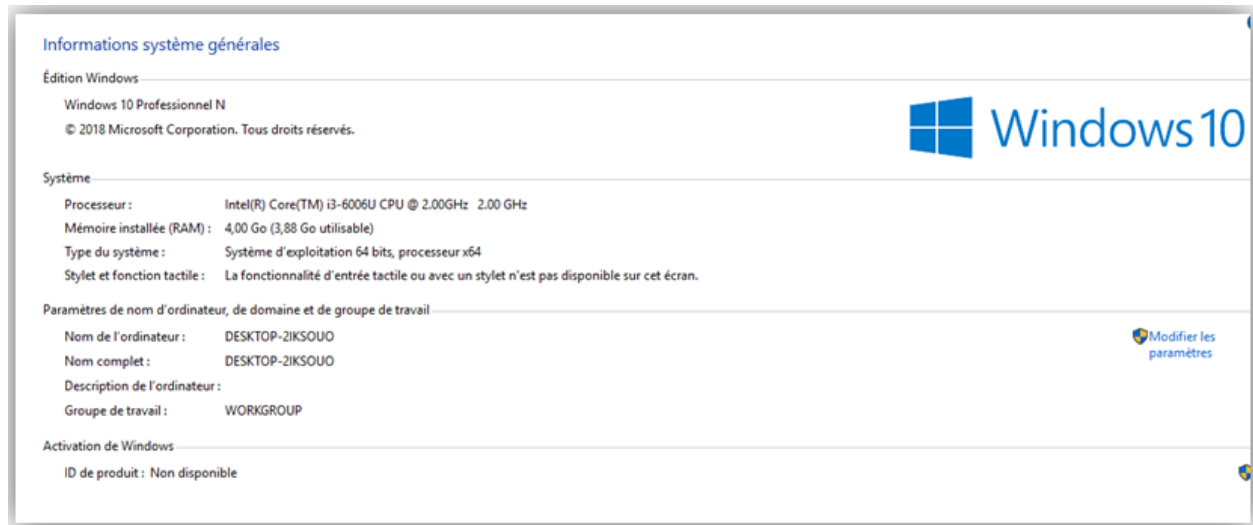


FIGURE 4.1 – Environnement logiciel utilisé.

4.2.2 Langage de programmation

De nos jours il existe de nombreux langages de programmation, plus au moins dédiés à tel ou tel type d'applications particulières. Parmi eux, notre choix s'est focalisé sur le langage JAVA.

Le langage Java

Apparu fin 1995 début 1996 et développé par Sun Microsystems Java s'est très rapidement taillé une place importante en particulier dans le domaine de l'internet et des applications client-serveur. Les objectifs de Java sont d'être multiplateformes et d'assurer la sécurité aussi bien pendant le développement que pendant l'utilisation d'un programme Java. Il est en passe de détrôner le langage C++ dont il hérite partiellement la syntaxe mais non ses défauts. Comme C++ et Delphi, Java est algorithmique et orienté objet à ce titre il peut effectuer comme ses compagnons, tous les tâches d'un tel langage (bureautiques, graphiques, multimédias, base de données, environnement de développement, etc...) . Son point de fort qui le démarque des autres est sa portabilité due à ses bibliothèques de classes indépendantes de la plate-forme ,ce qui est le point essentiel de la programmation sur internet ou plusieurs machines dissemblables sont interconnectées [38]

4.2.3 Outils et technologies

Eclipse

Eclipse est un projet, décliné et organisé en un ensemble de sous- projets de développements logiciels, de la fondation Eclipse visant à développer un environnement de production de logiciels libre qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java [38].

Son objectif est de produire et fournir des outils pour la réalisation de logiciels, englobant les activités de programmation (notamment environnement de développement intégré et frameworks) mais aussi d'AGL recouvrant modélisation, conception, test, gestion de configuration, reporting. . . Son EDI, partie intégrante du projet, vise notamment à supporter tout langage de programmation à l'instar de Microsoft Visual Studio.

Figurant parmi les grandes réussites de l'open source, Eclipse est devenu un standard du marché des logiciels de développement, intégré par de grands éditeurs logiciels et sociétés de services. Les logiciels commerciaux Lotus Notes 8, IBM Lotus Symphony ou WebSphere Studio Application Developer sont notamment basés sur Eclipse.



FIGURE 4.2 – Interface principale de eclipse.

Gestion de base de données

MySQL est un système de gestion de données relationnelles (SGBDR), rapide, robuste et facile d'utilisation. Il est adapté à la gestion de données dans un environnement réseau. Il est fourni avec des nombreux outils et qui est compatible avec de nombreux langages de programmation. Il est célèbre SGBDR du monde open-source, MySQL est surtout installé pour les applications Web, avec le serveur Xampp et le langage de page web dynamique JSP [39].



FIGURE 4.3 – Logo MySQL.

PhpMyAdmin

phpMyAdmin est une interface d'administration pour le SGBD MySQL. Il est écrit en langage PHP et s'appuie sur le serveur HTTP Apache. Il permet d'administrer les éléments suivants [36] :

- les bases de données.
- les tables et leurs champs (ajout, suppression, définition du type).
- les index, les clés primaires et étrangères.
- les utilisateurs de la base et leurs permissions.



FIGURE 4.4 – Logo PhpMyAdmin.

XAMPP : XAMPP est un ensemble de logiciels permettant de mettre en place facilement un serveur Web et un serveur FTP. Il s'agit d'une distribution de logiciels libres (XApache MySQL Perl PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de

connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus [37].



FIGURE 4.5 – Logo XAMPP.

4.3 La base de données

Pour implémenter

Le prototype, es c avorit créer une base de données de èns res denousnot qualité de service utilisées dans notre prototype sont :

Prix d'un service (Price) : C'est le coût à payer pour consommer le service, ce coût peut être fourni par le fournisseur du service.

Temps de réponse (Response Time) : Il est une performance qui représente la vitesse avec laquelle un service Web répond à une requête. Il est mesuré en milliseconde.

Fiabilité (Reliability) : Elle est la capacité d'un service à remplir ses fonctions requises dans les conditions indiquées pour une période de temps déterminée. Elle est mesurée en pourcentage.

Disponibilité (Availability) : La disponibilité d'un service est la probabilité que le service est accessible. La disponibilité d'un service Web pour les utilisateurs sur une période donnée, représenté comme un ratio du temps disponible divisé par le temps total, les fournisseurs calculent le ratio en incréments mensuels ou annuels.

Pour implémenter notre prototype, nous avons créer une base de données de trois tables :

- Client
- Fournisseur
- Service-web

La principale table est la table "Service-web". Elle contient les services web et leurs qualités de services.


#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra
<input type="checkbox"/> 1	Id_serv 	int(10)			Non	Aucune	AUTO_INCREMENT
<input type="checkbox"/> 2	Type_Service	varchar(20)			Non	Aucune	
<input type="checkbox"/> 3	Nom_Service	varchar(20)			Non	Aucune	
<input type="checkbox"/> 4	Cout	float			Non	Aucune	
<input type="checkbox"/> 5	Temp_reponce	float			Non	Aucune	
<input type="checkbox"/> 6	Fiabilite	float			Non	Aucune	
<input type="checkbox"/> 7	disponibilite	float			Non	Aucune	

FIGURE 4.6 – la table "Service-web".

4.4 Présentation des interfaces graphiques

Nous allons limité à quelques interfaces :

4.4.1 Interface d'accueil

La qualité de l'interface est l'une des caractéristiques qui attire l'utilisateur . De cela nous avons essayé de la représenter dans une bonne forme tout en respectant l'aspect de simplicité.

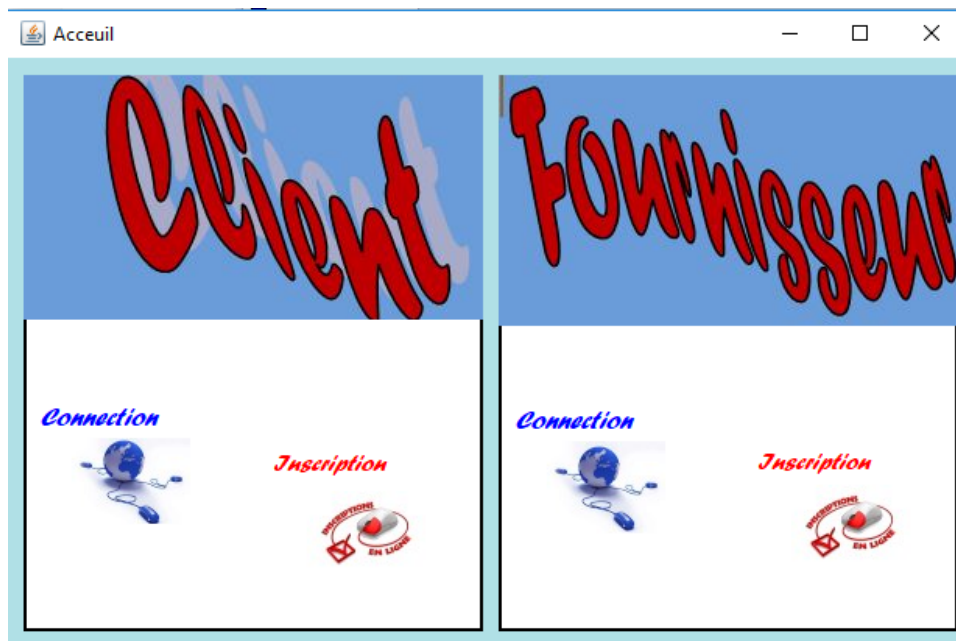


FIGURE 4.7 – Interface principale de l'application.

4.4.2 Interfaces principales

Notre système contient plusieurs interfaces qui traitent les différents cas d'utilisation des services Web comme suit :

Interface d'Inscription du Fournisseur

L'utilisateur doit préciser son statut avant de s'inscrire. Nous montrons l'interface de formulaire d'inscription du fournisseur comme le montre la figure 4.8. Dans la partie formulaire d'inscription de fournisseur, on trouve une partie qui concerne les informations personnelles du fournisseur.




The image shows a web browser window titled "Inscription fournisseur". The page has a light blue background. At the top left, there is a heading "Complétez le formulaire du fournisseur" in blue italicized text. Below this heading is a large illustration of a document with horizontal lines and a blue pen resting on it. To the right of the illustration is a registration form with four input fields. The labels for these fields are "Nom", "Prénom", "Nom D'utilisateur", and "Mot de passe", all in blue italicized text. At the bottom right of the form area, there are two buttons: "inscrire" and "Retour", both in blue italicized text.

FIGURE 4.8 – interface d'inscription du fournisseur.

Interface d'Inscription du Client

Un client doit aussi s'inscrire afin de pouvoir réaliser ses tâches. La figure 4.9 montre l'interface d'inscription du client.



Inscription Client

Complétez le formulaire du client

Nom

Prénom

Nom D'utilisateur

Mot de passe

inscrire *Retour*

FIGURE 4.9 – interface d'inscription du client.

Interface Login du fournisseur

Dans le cas d'ajout d'un nouveau service Web à notre base de données par un fournisseur des services, il faut d'abord authentifier l'identité du fournisseur.



Session Fournisseur

Nom d'utilisateur

Mot de passe

Ok *Annuler*

FIGURE 4.10 – Interface Login du fournisseur.

Interface Login du client

Un client qui demande un service web il faut d'abord s'authentifier l'identité.

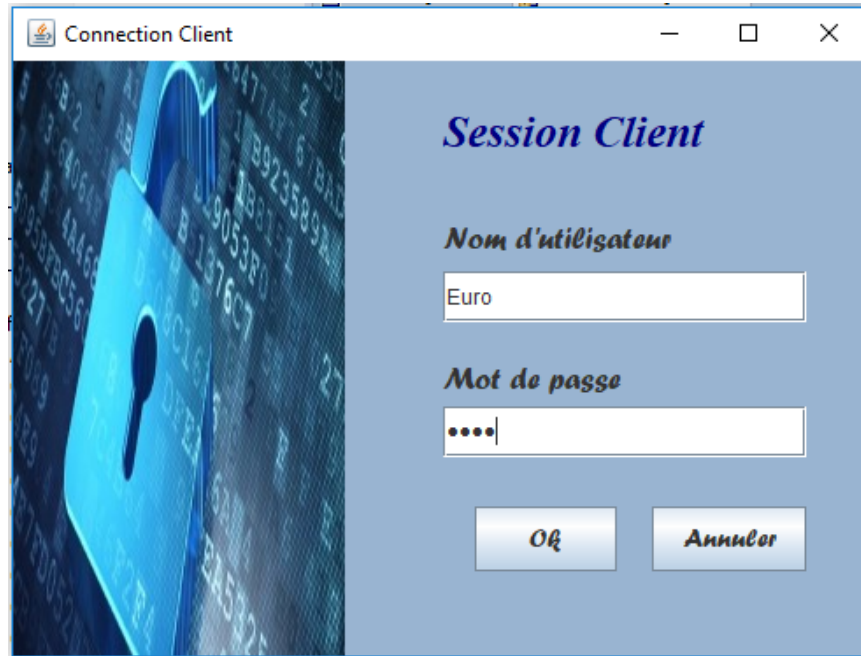


FIGURE 4.11 – Interface Login du client.

4.4.3 Interface d'Espace Client

Pour le client, une fois qu'il est inscrit il sera redirigé vers «l'espace client» qui lui permet de sélectionner les services satisfaisant ses besoins à travers l'interface montré dans la figure 4.12.

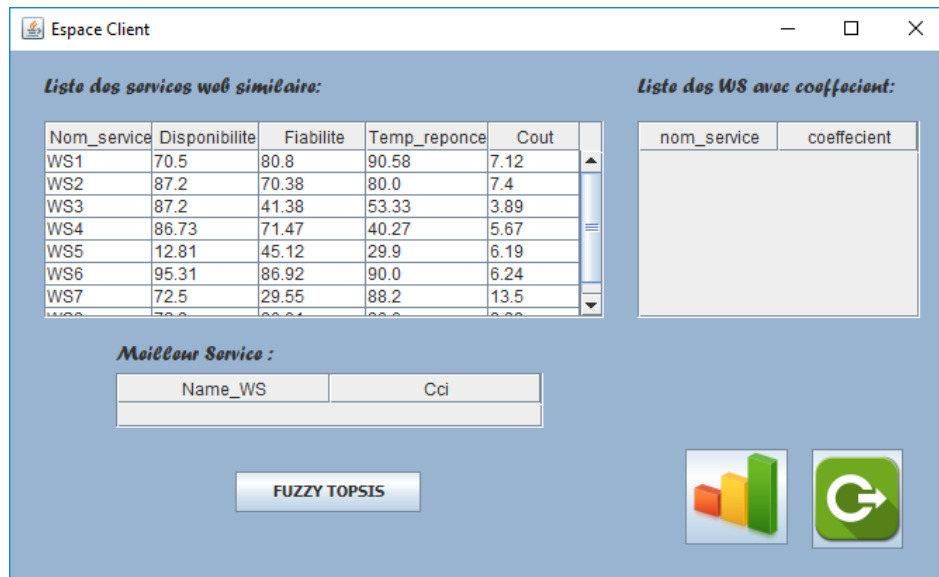


FIGURE 4.12 – Interface d’espace client et affichage des services similaires.

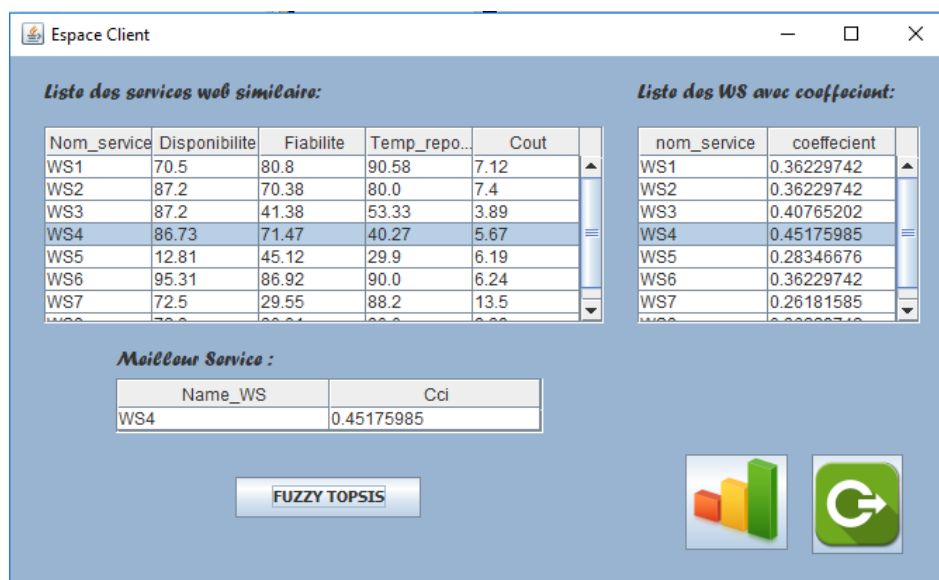


FIGURE 4.13 – Résultat calculé par TOPSIS Floue.

Dans la fenêtre «Espace client» un clic sur le bouton «FUZZY TOPSIS» fait apparaître une fenêtre, qui explique les étapes de la méthode TOPSIS Flou.

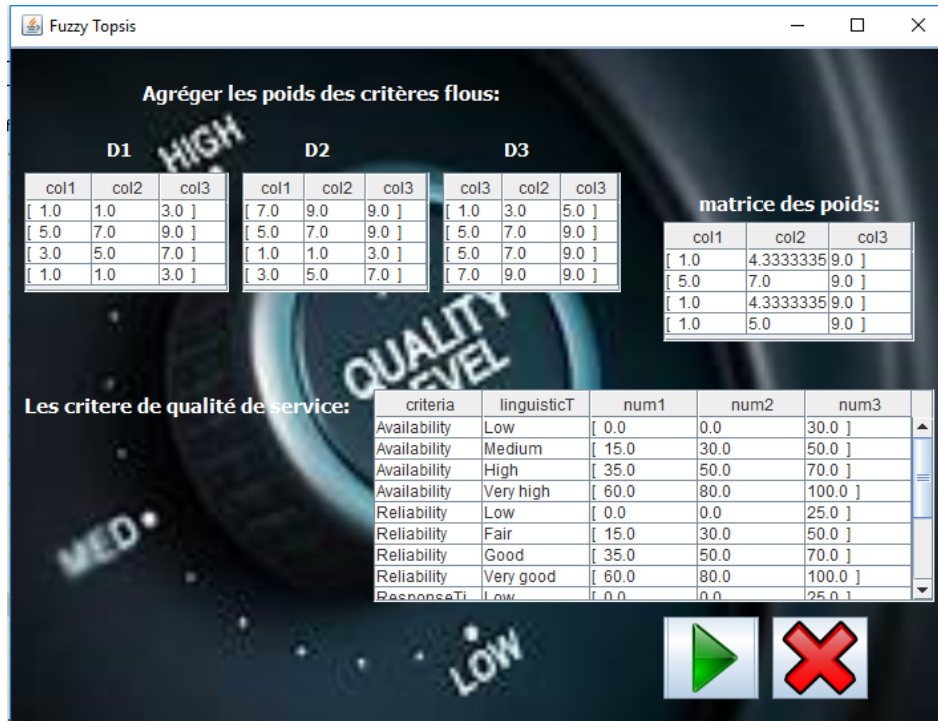


FIGURE 4.14 – Etape 01 : TOPSIS Flou.

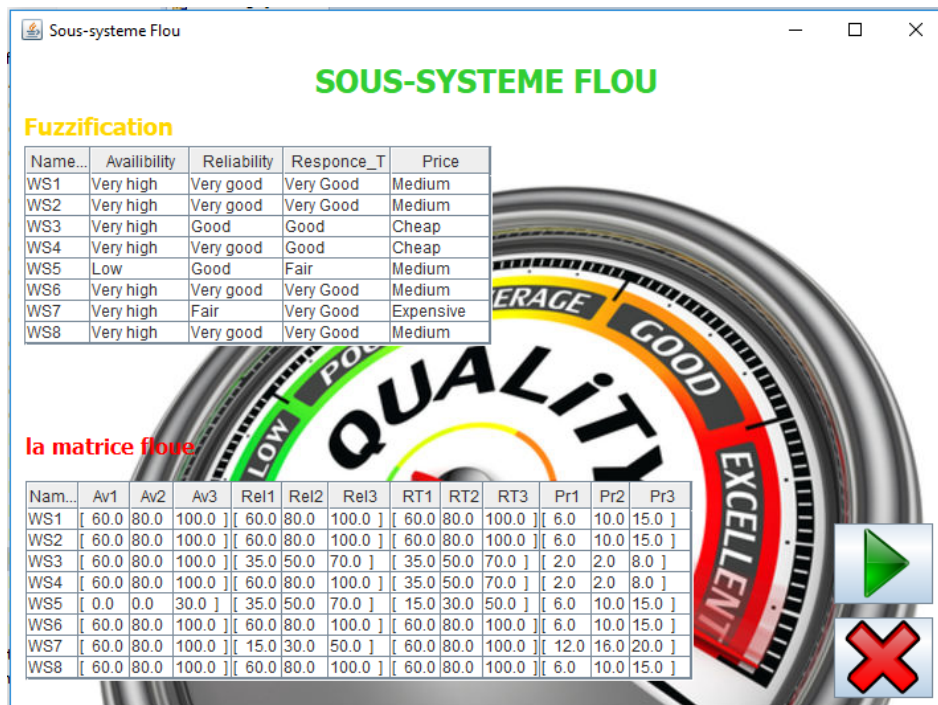


FIGURE 4.15 – Etape 02 : TOPSIS Flou.

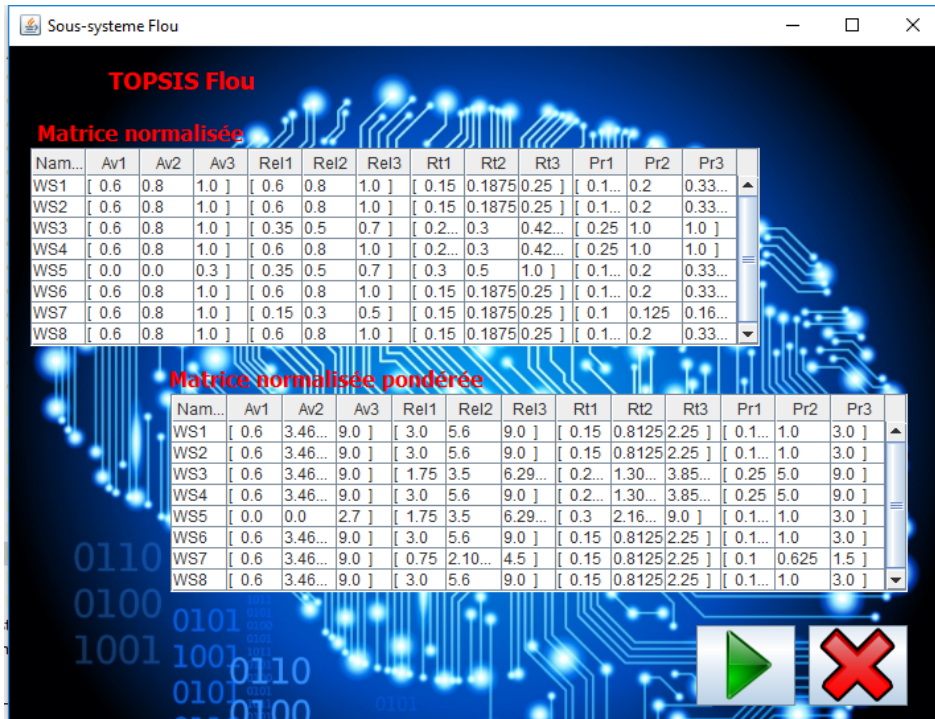


FIGURE 4.16 – Etape 03 : Topsis Flou.

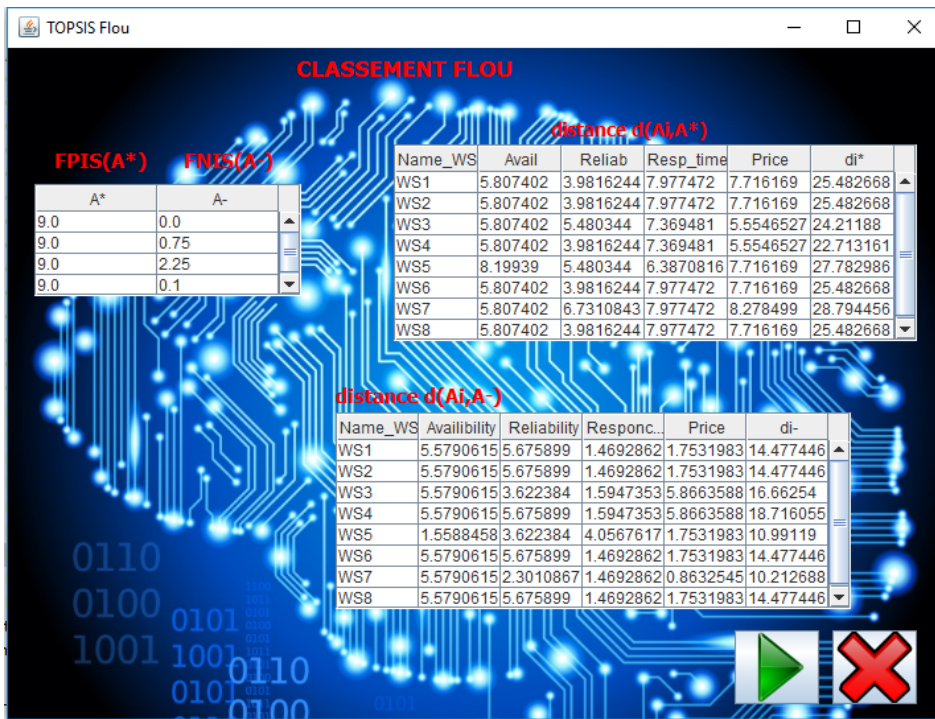


FIGURE 4.17 – Etape 04 : Topsis Flou.

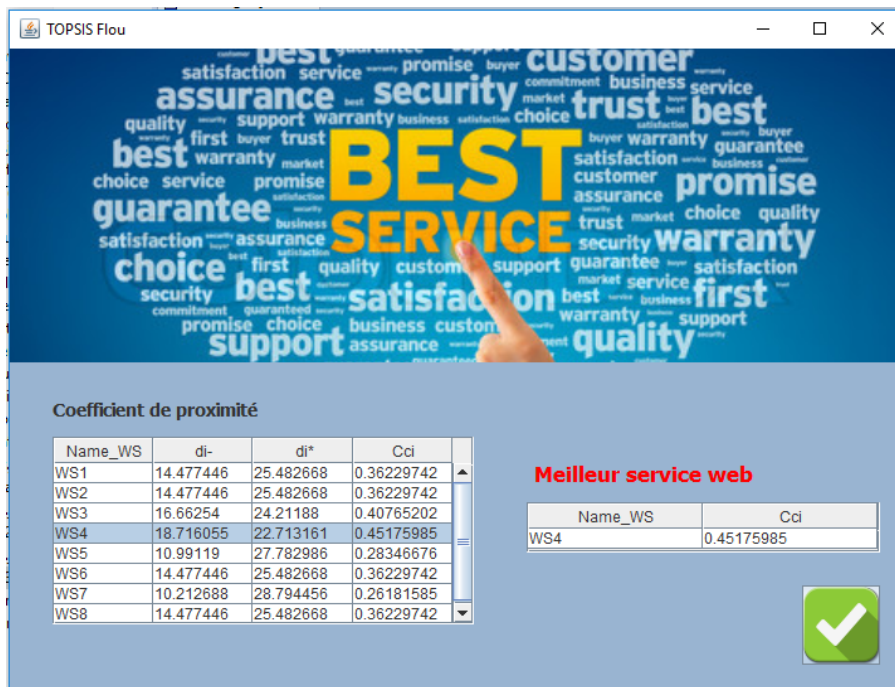


FIGURE 4.18 – Etape 05 : TOPSIS Flu.

4.4.4 Interface d'Espace Fournisseur

Une fois un fournisseur s'inscrit il sera rediriger vers son espace personnel qui est l'«Espace Fournisseur» qu'à travers il pourra publier les descriptions de son service. La figure 4.17 montre l'interface de l'espace du fournisseur et les différentes fonctionnalités d'un fournisseur.

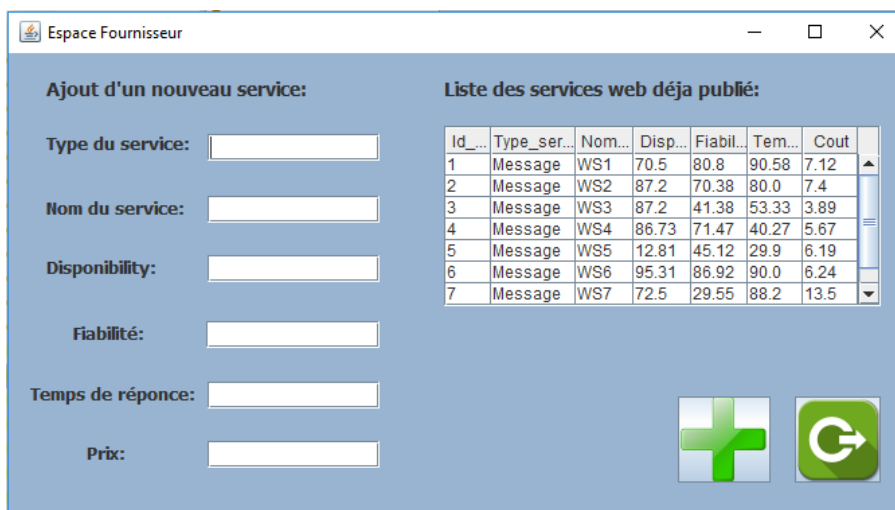


FIGURE 4.19 – Interface Fournisseur ajout d'un nouveau service.

4.5 Conclusion

Pour valider la conception d'un système, il faut l'implémenter en utilisant les outils adéquats. Dans ce chapitre nous avons présenté quelques détails concernant la réalisation de notre application de sélection des services Web à base de qualité de service (QoS) avec une intégration de l'algorithme TOPSIS floue, qui s'occupe du classement des services avec leurs qualités à travers des étapes bien définies et bien spécifiques. Nous avons testé notre proposition avec une liste des services web similaires.

D'après les résultats obtenus, nous pouvons confirmer que les résultats sont favorables et que la méthode TOPSIS floue est très efficace dans le domaine de sélection de services Web.

Nous avons présenté quelques détails concernant la réalisation de notre application, en choisissant le langage de programmation java pour sa compatibilité avec les concepts orienté objet, et également pour ces avantages multiples de simplicité, portabilité et sécurité. Nous avons aussi présenté un exemple illustrant les différents services offerts par notre application.

Conclusion générale

Ce projet de fin d'étude présente le fruit de notre travail, il nous a permis d'acquérir beaucoup de nouvelles connaissances.

Pour répondre à notre objectif, nous avons débuté notre travail par une présentation détaillée des concepts de base des services Web, qui sont des technologies émergentes et prometteuses pour le développement, le déploiement et l'intégration d'applications sur l'Internet. Ils constituent la technologie de base pour le développement d'architectures orientées services. Ces architectures sont de plus en plus répandues sur le Web.

Actuellement, de nombreux services Web, avec des fonctionnalités similaires sont fournis par des fournisseurs concurrents, et de ce fait les utilisateurs finaux ont besoins d'approches efficaces pour la sélection des services.

Nous avons proposé une nouvelle approche qui sert à la découverte des services Web, en s'appuyant sur l'algorithme TOPSIS flou pour le classement des services similaires qui facilitera la sélection des services pertinents selon le besoin des clients. avec exemple service web type message.

Dans le premier chapitre, nous avons présenté les technologies des services Web et les principaux standards qu'elle supporte. Le second chapitre est consacré pour la logique floue, qui est une approche au raisonnement humain. Les fonctions d'appartenances sont le coeur de la logique floue, ils mettent la différence entre la logique classique et la logique floue.

Le troisième chapitre a été consacré à la modélisation de notre approche.

Pour implémenter notre approche proposée, nous avons présenté dans le chapitre 4 le déroulement d'un prototype qui montre la faisabilité de la méthode TOPSIS Flou dans le classement des services web.

comme perspectives, nous envisagerons d'améliorer notre proposition par les points sui-

vants :

- intégré le paradigme agent. qui représente une grande potentialité (autonomie, intelligence, coopération, coordination ... etc) dans le web.
- Créer une application mobile universelle (Android, iOS,) qui permet d'effectuer toutes les besoins qu'un client puisse invoquer.
- Permettre aux clients de choisir les critères.
- Ajouter d'autres critères pour donner plus de choix aux clients.
- Tester notre prototype avec un DataSet réelles des services web.

Bibliographie

- [1] MEGHERBI SAMMY, Une approche basée agent pour la sélection des Services Web , 2018, mémoire de master, Université Mohamed khider Biskra.
- [2] Radhouane MIMOUNE, proposition d'un modèle pour la prise en compte de la qualité de sevice dans la selection des SW, 2017, mémoire de master, Université Mohamed khider Biskra.
- [3] BELOUAAR, Houcine. Modélisation d'une approche basée agent et logique floue pour la qualité des services Web. 2018. Thèse de doctorat. Université Mohamed Khider Biskra.
- [4] Système interactif de banque à base de la technologie SOAP des services Web, par : MI-LOUDI Lina , Rapporteur : BELOUAAR Houcine
- [5] Application de la technique de la logique floue pour la prédiction de l'amorçage des intervalles d'air pointes-plans, Présenté par :BOUZID Houssam Eddine BENMERIEM Soufiane.
- [6] THESE DE DOCTORAT : Développement d'une approche floue multicritère d'aide à la coordination des décideurs pour la résolution des problèmes de sélection dans les chaines logistiques, Idris IGOULALENE.
- [7] Nădăban, Sorin, Simona Dzitac, and Ioan Dzitac. "Fuzzy TOPSIS : a general view." *Procedia Computer Science* 91 (2016) : 823-831.]
- [8] "A Simplified Description of FUZZY TOPSIS Method for Multi Criteria Decision Making" Miss. Neelima B. Kore¹, Prof. Dr. K. Ravi², Asst. Prof. Mr. S. B. Patil³
- [9] Demba COULIBALY. Un langage et un environnement de conception et de développement de services web complexes. Thèse du doctorat, 2006.

- [10] BACHTARZI, Fayçal. Une Approche de Composition des Services Web Basée Transformation de graphes. université costantine, 2014, vol. 2.
- [11] Site web, URL : <http://www.service-architecture.com/>. Consulté le 01/06/2019.
- [12] DODANI, Mahesh H. From Objects to Services : A Journey in Search of Component Reuse Nirvana. *Journal of Object Technology*, 2004, vol. 3, no 8, p. 49-54.
- [13] VERJUS, Hervé et POURRAZ, Frédéric. A formal framework for building, checking and evolving service oriented architectures. In : Fifth European Conference on Web Services (ECOWS'07). IEEE, 2007. p. 245-254.
- [14] ZUO, Wei. Managing and modeling web service evolution in SOA architecture. 2016. Thèse de doctorat. Université de Lyon.
- [15] DEHANE, Aicha Djihad et KEBIR, Zohra. Evaluation des techniques de codage d'ontologies sur les performances de la composition de services Web. 2011.
- [16] Hamida, Soraya. Une approche basée agent mobile pour le m-service web sémantique. Diss. Université Mohamed Khider Biskra, 2014.
- [17] Bouchahma Karim. Conception et développement d'un service web pour l'échange d'information dans le domaine humanitaire. TRAVAIL DE BACHELOR "Suisse".
- [18] BOUDJELABA Hakim. Sélection des web services sémantiques. Mémoire de Magister, 2012.
- [19] Djelmoudi Abdelhamid et Belaredj Mohammed Amin. La sélection des services web à base des systèmes immunitaire artificiels (clonalg). Mémoire Master, 2012.
- [20] KHELLAF Radhia. Vérification de la des services compatibilité web pour une composition. Thèse de Magister, 2014.
- [21] Sofiane Chemaâ. Une approche de composition de services web à l'aide des réseaux de petri orientés objet. Thèse de Doctorat, 2014.
- [22] SOUKKARIEH Bouchra. Technique de l'internet et ses langages : vers un système d'information web restituant des services web sensibles au contexte. THÈSE de DOCTORAT, 2010.

- [23] Site web, URL : <https://www.w3.org/2002/07/soap-translation/soap12-part1.html>. Consulté le : 24/05/2019.
- [24] Site web, URL : " www.w3.org/2002/07/soap-translation/soap12-part0.html." Consulté le : 24/04/2019.
- [25] MOULAY KHATIR Oussama Mohy eddine et HENAOUI Redouane. Sélection de service web à base d'algorithme mimétique. Mémoire de Master, 2013.
- [26] " Jean-Pierre BORG EFORT. Les services web. Cours d'informatique, 2011.
- [27] Sofiane Chemaï. Une approche de composition de services web à l'aide des réseaux de petri orientés objet. Thèse de Doctorat, 2014.
- [28] Sivanandam, S. N., Sai Sumathi, and S. N. Deepa. Introduction to fuzzy logic using MATLAB. Vol. 1. Berlin : Springer, 2007.
- [29] DERNONCOURT, Franck, and ELISABETH MÉTAIS. "La Logique Floue : le raisonnement humain au coeur du système décisionnel?." Memory NFE211 engineering decision systems Paris (February 2011) <http://www.academia.edu/1053161> (2011).
- [30] Hema Priya, N., and S. Chandramathi. "QoS Based Optimal Selection of Web Services Using Fuzzy Logic." Journal of Emerging Technologies in Web Intelligence 6.3 (2014).
- [31] Malik, Mohamed Mahdi. "Le rôle de la logique floue dans le web sémantique." Secrétariat du DEA d'Informatique—Université de la Méditerranée Faculté des Sciences de Luminy—Case 901 163, Avenue de Luminy—13288 MARSEILLE Cedex 9 Tel : 0 491 829 316—Télécopie : 0 491 829 275 secrétariat : secretariat-dea@dil.univ-mrs.fr (2002) : 263.
- [32] Omar, Amina S., Mwangi Waweru, and Richard Rimiru. "A Literature Survey : Fuzzy Logic and Qualitative Performance Evaluation of Supply Chain Management." The International Journal Of Engineering And Science (IJES) Volume 4 : 56-63.
- [33] BOUZID Houssam Eddine, BENMერიEM Soufiane. Application de la technique de la logique floue pour la prédiction de l'amorçage des intervalles d'air pointes-plans. Mémoire Master académique.2013.

- [34] Zadeh, Lotfi A. "Is there a need for fuzzy logic?." *Information sciences* 178.13 (2008) : 2751-2779.
- [35] Leonid, Reznik. "Fuzzy controllers." *Newnes* 1 (1997) : 1-9.
- [36] Omar, Amina S., Mwangi Waweru, and Richard Rimiru. "A Literature Survey : Fuzzy Logic and Qualitative Performance Evaluation of Supply Chain Management." *The International Journal Of Engineering And Science (IJES)* Volume 4 : 56-63.
- [37] Site web, URL : <https://desgeeksetdeslettres.com/programmation-java/xampp-plateformepour-heberger-son-propre-site-web>. Consulté le 15/05/2019.
- [38] Site web, URL : [https://fr.wikipedia.org/wiki/Java_\(langage\)](https://fr.wikipedia.org/wiki/Java_(langage)). Consulté le : 02/04/2019.
- [39] Site web, URL : <https://dev.mysql.com/downloads/>. Consulté le : 05/04/2019.
- [40] Chen, Vivien YC, et al. "Fuzzy MCDM approach for selecting the best environment watershed plan." *Applied soft computing* 11.1 (2011) : 265-275