



الجمهورية الجزائرية الديمقراطية الشعبية



République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Abbes Laghrou de Khenchela

Faculté des Sciences et Technologies

Département Maths et Informatique

Filière : Maths et Informatique

Spécialité : GLSD (Génie Logiciel et Système Distribué)

MEMOIRE

Présenté pour obtenir le Diplôme de Master

La Logique Floue pour La Sélection de service web à base QoS

Réalisé Par :

Manaa Said

Devant le jury :

PRESIDENT:	MCA	Abd El hadi Adel
RAPPORTEURE:	MCA	Chouhal Ouahiba
EXAMINATEUR:	MAA	Mebarki djmouai

Année universitaire : 2020 / 2021

Remerciement

Au terme de ce mémoire, je tiens en premier lieu à exprimer ma profonde gratitude à mon dieu qui m'a donné la force, la volonté, le courage, et surtout la patience pour terminer ce travail.

je remercie vivement Dr. Chouhal Ouahiba, qui n'ont pas hésité à me faire confiance et à avoir accepté de diriger ce travail. Elle m'a fait part de son expérience et de son savoir-faire. Je tiens aussi à le remercier énormément pour ses conseils ciblés et son soutien.

J'aimerais aussi exprimer ma profonde reconnaissance à tous ceux qui ont collaboré de près ou de loin dans la réalisation de ce travail.

Dédicace

je tiens à remercier profondément ma proche famille, mes frères et surtout mes parents dont la bienveillance et l'appui n'ont cessé de m'encourager depuis mon enfance dans chaque chose que j'entreprends. Ils m'ont permis de surmonter toutes les épreuves, même les plus difficiles. Je souhaite dédier ce travail à ma mère et mon père.

Résumé:

Les services Web sont des technologies émergentes qui répondent aux besoins des utilisateurs sur le Web. Choisir le bon service pour vos besoins est une tâche ardue. Il existe un grand nombre de Fournisseurs de services qui fournissent le même type de service, mais il existe des différences de qualité fonctionnelle et non fonctionnelle. Certains utilisateurs peuvent avoir besoin de plusieurs services en même temps. De plus, les consommateurs seront confus lors du choix des services, ce qui entraînera la nécessité d'une sélection automatique des services et d'un mécanisme basé sur les paramètres de qualité de service (QoS). Pour une meilleure classification des services web selon les critères qualité nous proposons l'utilisation de la technologie : la logique floue.

Donc, dans ce travail nous résoudrons le problème de la sélection de services en fonction des attributs de QoS. L'objectif est de sélectionner un ensemble de services pour répondre aux contraintes standard (minimiser le temps et le coût, et maximiser la disponibilité), et nous pouvons appliquer des contraintes définies par le client pour sélectionner les services les plus proches de ces exigences. La méthode proposée commence par le chargement des valeurs de QoS, puis les traite avec une logique floue.

Mots-clés : Services Web, QoS, logique floue

Table des matières

Introduction Générale.....	4
Problématique et objectifs.....	5
Plan du mémoire.....	6
Chapitre 1 : Les Services web et l'Architecture SOA.....	8
1.1 Introduction.....	8
1.2 Architecture Orientée Service (SOA).....	9
1.2.1 Définition de SOA.....	9
1.2.2 Avantages d'une architecture orientée service.....	9
1.2.3 Caractéristiques et fonctionnement de SOA.....	10
1.3 Service Web.....	11
1.3.1 Définition d'un service Web.....	11
1.3.2 Caractéristiques des services Web.....	12
1.3.2.1 Les caractéristiques fonctionnelles.....	12
1.3.2.2 Les caractéristiques non fonctionnelles.....	12
1.4 SOA à base de service web.....	13
1.4.1 Architecture étendue.....	14
1.5 Standards de base de services Web.....	15
1.5.1 Le protocole SOAP.....	15
1.5.1.1 Définition.....	15
1.5.1.2 Structure d'un message SOAP.....	16
1.5.2 WSDL (Web Services Description Language).....	17
1.5.3 UDDI (Universal Description Discovery and Integration).....	19
1.6 Conclusion.....	20
Chapitre 2 : Logique Floue et Système d'inférence Floue.....	22
2.1 Introduction.....	22
2.2 Historique De La Logique Floue.....	22
2.3 Le Concept De La Logique Floue.....	23
2.4 Théorie Des Sous-ensembles Flous	23
2.4.1 Définition	23
2.4.2 Comparaison : la logique floue et logique classique.....	24
2.4.3 Les Operations de La Logique Floue.....	24
2.4.4 Variable linguistique.....	26
2.4.5 Les Fonctions d'appartenance.....	27
2.5 Structure D'une Commande Floue.....	28
2.5.1 Bases De Règles Et Définitions.....	28
2.5.2 Système D'inférence Floue	30
2.6 Conclusion.....	37
Chapitre 3 : Modélisation et Implémentation.....	39
3.1 Introduction.....	39
3.2 Qualité de service.....	39
3.2.1 Paramètres de qualités de services.....	39
3.3 La sélection des services Web basée sur la logique floue.....	40
3.4 Sous-système flou.....	42
3.4.1 Fuzzification.....	42
3.4.2 Moteur d'inférence floue.....	43

3.4.2.1	Base de règles floues	44
3.4.3	Défuzzification.....	44
3.5	Détails d'implémentation.....	44
3.5.1	Algorithme de sélection floue.....	47
3.6	Implémentation et expérimentation.....	47
3.6.1	Scenario pour un seul service web	48
3.6.2	Présentation des interfaces graphiques.....	50
3.7	Conclusion.....	54
	Conclusion générale.....	55
	Bibliographie.....	56

Liste des figures

Figure (1.1) - Modèle fonctionnel de l'architecture SOA.....	10
Figure (1.2) - Architecture en Pile des services web.....	14
Figure (1.3) - Traitement d'un message SOAP.....	15
Figure (1.4) - Structure de message SOAP.....	16
Figure (1.5) - Structure d'un document WSDL.....	17
Figure (1.6) - les trois types de l'annuaire UDDI.....	19
Figure (2.1) - l'intersection « ET » de deux ensembles flous.....	24
Figure (2.2) - l'union « OU » de deux ensembles flous.....	25
Figure (2.3) - la complément « NON » d'ensembles flous.....	25
Figure (2.4) - Variable linguistique de la qualité du service.....	26
Figure (2.5) - Fonction d'appartenance floue triangulaire.....	27
Figure (2.6) - Fonction d'appartenance floue trapézoïdale.....	28
Figure (2.7) - la Structure d'une commande floue.....	29
Figure (2.8) - Schémas en blocs un système d'inférence floue.....	30
Figure (2.9) - Schémas d'inférence floue.....	32
Figure (2.10) - Schémas d'implication floue de Mamdani.....	33
Figure (2.11) - Schémas d'agrégation des conclusions.....	34
Figure (2.12) - Défuzzification par centre de gravité.....	35
Figure (2.13) - Défuzzification par valeur maximum.....	36
Figure (3.1) - Architecture de la sélection des services Web basée sur la logique floue.....	41
Figure (3.2) – Architecture de l'approche proposée	42
Figure (3.3) – Variable linguistique Disponibilité.....	45
Figure (3.4) – Variable linguistique Prix.....	46
Figure (3.5) – Variable linguistique Temps de Réponse.....	46
Figure (3.6) – la table " webService".....	47
Figure (3.7) – Méthode de Fuzzification pour VL Disponibilité VeryGood.....	48
Figure (3.8) – Méthode D'inférence de la règle 26.....	49
Figure (3.9) – Méthode de Défuzzification pour VL Disponibilité VeryGood.....	49
Figure (3.10) – Login du Fournisseur et Client	50
Figure (3.11) – Inscription du Fournisseur et Client.....	51
Figure (3.12) – Espace Fournisseur.....	52
Figure (3.13) – Espace Client	53

Introduction Générale

En substance, un service Web peut être défini comme une abstraction bien définie d'un ensemble d'activités informatiques ou physiques, impliquant de nombreuses ressources et conçu pour répondre aux besoins des clients ou aux besoins de l'entreprise.

Avec la popularisation d'Internet, les services Web sont devenus indispensables dans presque tous les domaines. Ils constituent une classe de solutions Web utilisées dans des secteurs tels que la banque, les achats et les communications. Le service Web peut être simplement décrit comme tout service fourni sur le Web. Par exemple, dans une application web météo, un utilisateur ouvre un site web météo et entre des données (par exemple : code postal), le serveur du site traite les données et envoie une réponse, en l'occurrence la météo.

Les entreprises utilisent de plus en plus les applications B2B (Business-to-Business Integration) et B2C (Business-to-Consumer) pour s'adapter au marché via Internet.

Avec le développement de la technologie et un grand nombre de services disponibles sur Internet, le problème de la recherche de services en réseau s'est posé, il est devenu très important de pouvoir trouver une variété de services qui répondent à nos besoins. Certaines suggestions ont été faites pour résoudre ce problème, et certaines normes ont été définies, mais aucune d'entre elles n'est efficace ou n'est maintenant acceptée comme moyen d'effectuer une recherche sur les services.

Problématique et objectifs

Le nombre de services Web qui assurent la même fonction sur le Web n'a cessé d'augmenter. Une seule demande d'un client peut avoir des centaines voire des milliers de réponses. Le client est gâté et confronté à la question de savoir comment choisir un service similaire à revenu. Cependant, il est très rare de répondre aux demandes des clients avec un seul service.

Besoin de pouvoir accéder efficacement aux services Web. De plus, étant donné que les fournisseurs concurrents sont censés fournir des services Web avec des fonctions similaires, le principal défi consiste à concevoir une stratégie d'optimisation pour trouver le « meilleur » service Web.

De plus, dans de nombreux cas, la valeur de l'attribut QoS peut être difficile à définir avec précision car la qualité de service liée au consommateur du service Web n'est pas exacte, parfois en raison de l'état mental et du manque d'information du consommateur, incertain et ambigu. De même, la valeur de la QoS (en particulier la QoS qualitative) n'est pas facile à définir avec précision, par conséquent, les préférences de l'utilisateur utilisées comme attributs de QoS doivent être ambiguës car elles sont plus adaptées à l'interprétation des termes linguistiques. Par conséquent, la logique floue peut être appliquée pour prendre en charge la représentation de contraintes de QoS inexactes.

Le but de ce travail est de fournir une méthode de sélection de services Web basée sur la logique floue.

À cette fin, nous recommandons d'utiliser un système d'inférence floue. D'une part, cette méthode doit permettre aux utilisateurs du service Web d'initier leurs requêtes sous forme de langage, et d'autre part, présenter différents paramètres de qualité de service à valeurs floues, afin que le service le plus adapté soit finalement renvoyé au client.

Plan du mémoire

Après une introduction générale au contexte de travail, aux problèmes et aux solutions suggérées, l'organisation de cet article est la suivante :

- Le premier chapitre «Les services web et l'architecture SOA» : on va présenter les services web et l'architecture SOA.
- Le deuxième chapitre «Logique Floue et Système d'inférence Floue» : C'est une méthode de raisonnement humain. La fonction d'appartenance est au cœur de la logique floue. Le système de raisonnement flou est constitué des trois étapes dans lesquelles le système joue le rôle d'humain et commence à raisonner : la fuzzification, le raisonnement et la défuzzification.
- Le troisième chapitre «Modélisation et Implémentation»: Ce chapitre sera consacré au l'architecture de notre système, nous présenterons aussi les descriptions détaillée de ses composants, aussi nous allons présenter l'application Qui en a réalisé.

Ces trois chapitres se terminent par une conclusion générale, résumant les résultats de nos travaux, et présentant les perspectives que nous espérons réaliser à l'avenir.

CHAPITRE 01 :

Les Services Web et l'architecture SOA

Chapitre 1 : Les Services web et l'Architecture SOA

1.1 Introduction

Aujourd'hui, les entreprises et les systèmes d'information expriment une forte demande d'échange d'informations et de services. Cela nécessite un langage de communication commun. Les efforts de standardisation de ces langages ont donné naissance à un nouveau domaine de recherche appelé le « protocole B2B ». Les technologies émergentes dans ce domaine permettent de suivre certains canaux de communication intéressants entre les entreprises. Cette technologie est une technologie de service Web. Les services Web sont un nouveau paradigme dont le but est de transformer l'architecture par composant au sein du Framework Web. Les services Web sont des composants logiciels qui fournissent des services via des interfaces standardisées. Les services Web se caractérisent par l'utilisation de la technologie Internet comme infrastructure de communication entre eux.

Les fonctions fournies par les services Web sont relativement simples, mais les besoins des utilisateurs deviennent de plus en plus complexes, de sorte qu'un seul service Web ne peut les satisfaire. La participation de la composition de services Web vise à créer de nouveaux services Web dits composites et la sélection automatique de ces derniers offre une nouvelle perspective pour répondre aux besoins complexes des utilisateurs.

1.2 Architecture Orientée Service (SOA)

L'architecture orientée services (SOA) est décrite comme une stratégie de création d'applications orientées services. Son but est de fournir des services qui peuvent être utilisés par d'autres services.

1.2.1 Définition de SOA

Voici Une Définition parmi plusieurs 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.». [2]

La chose la plus importante dans l'architecture SOA est la Normalisation. Il définit les spécifications de tous les types d'applications à découvrir et à communiquer entre elles. D'une part, il permet également d'utiliser les systèmes existants pour coopérer avec les nouvelles technologies modernes. D'autre part, la description des interfaces standard permet l'utilisation de différents langages, plates-formes, modèles d'objets ou systèmes de messagerie pour mettre en œuvre des services.

1.2.2 Avantages d'une architecture orientée service

Une architecture orientée services permet d'obtenir tous les avantages d'une architecture client-serveur et notamment [5]:

- Une modularité permettant de remplacer facilement un composant (service) par un autre
- Une réutilisabilité possible des composants (par opposition à une système tout-en-un fait sur mesure pour une organisation).
- De meilleures possibilités d'évolution (il suffit de faire évoluer un service ou d'ajouter un nouveau service)
- Une plus grande tolérance aux pannes
- Une maintenance facilitée.

1.2.3 Caractéristiques et fonctionnement de SOA

Il n'existe pas à proprement parler de spécifications officielles d'une architecture SOA, néanmoins les principales notions fédératrices que l'on retrouve dans une telle architecture sont les suivantes [5]:

- La notion de service, c'est-à-dire une fonction encapsulée dans un composant que l'on peut interroger à l'aide d'une requête composée d'un ou plusieurs paramètres et fournissant une ou plusieurs réponses. Idéalement chaque service doit être indépendant des autres afin de garantir sa réutilisabilité et son interopérabilité.
- La description du service, consistant à décrire les paramètres d'entrée du service et le format et le type des données retournées. Le principal format de description de services est WSDL (Web Services Description Language), normalisé par le W3C.
- La publication (en anglais advertising) et la découverte (discovery) des services. La publication consiste à publier dans un registre (en anglais registry ou repository) les services disponibles aux utilisateurs, tandis que la notion de découverte recouvre la possibilité de rechercher un service parmi ceux qui ont été publiés. Le principal standard utilisé est UDDI (Universal Description Discovery and Integration), normalisé par l'OASIS.
- L'invocation, représentant la connexion et l'interaction du client avec le service. Le principal protocole utilisé pour l'invocation de services est SOAP (Simple Object Access Protocol).

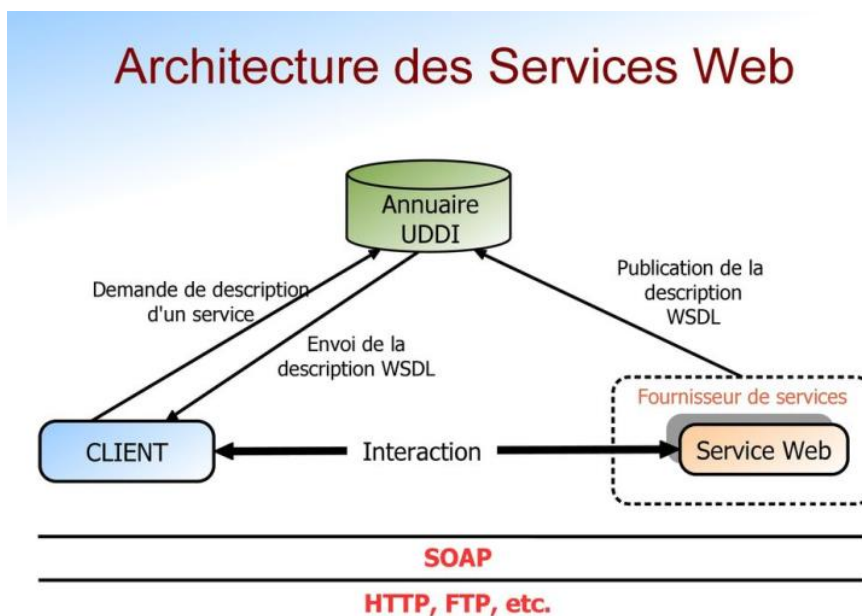


Figure (1.1) - Modèle fonctionnel de l'architecture SOA

1.3 Service Web

1.3.1 Définition d'un service Web

Plusieurs définitions des services web ont été proposées. Nous citons celles de W3C, d'IBM et de Wikipédia. [1]

Le groupe de W3C qui travaille sur les services Web a défini le service Web comme « un système logiciel destiné à supporter l'interaction ordinateur-ordinateur sur le réseau. Il a une interface décrite en un format traitable par l'ordinateur (e.g. WSDL). Autres systèmes réagissent réciproquement avec le service web d'une façon prescrite par sa description en utilisant des messages SOAP, typiquement transmis avec le protocole http et une sérialisation XML, en conjonction avec d'autres standards relatifs au web ». [6]

Selon [7] : « Un service web est une agrégation de fonctionnalités publiées pour être utilisées. Il utilise internet comme conduit pour réaliser une tâche. Il est semblable à un processus métier virtuel qui définit des interactions au niveau application »

Selon IBM [8] : « Les services web sont la nouvelle vague des applications Web. Ce sont des applications modulaires, auto-contenues et auto descriptive qui peuvent être publiées, localisées et invoquées depuis le web. Les services web effectuent des actions allant de simples requêtes à des processus complexes. Une fois qu'un service Web est déployé, d'autres applications peuvent le découvrir et l'invoquer ».

1.3.2 Caractéristiques des services Web

Chaque service web a deux types de caractéristiques [3]:

1.3.2.1 Les caractéristiques fonctionnelles

Les caractéristiques fonctionnelles d'un service web désignent les opérations qu'il peut fournir. Elles sont décrites dans la description de service en termes d'opérations et reflètent le fonctionnement du service.

1.3.2.2 Les caractéristiques non fonctionnelles

Les caractéristiques non fonctionnelles d'un service web appelées aussi : qualités de service QoS (Quality Of Service) définissent les capacités de service à fonctionner dans de bonnes conditions en termes de disponibilité, performance, etc. QoS est un ensemble de propriétés opérationnelles du service que l'on doit constater dans la réalisation de service. Formalisées dans le contrat, ces propriétés représentent un ensemble d'exigences concernant la mise en œuvre du service et constituent donc un engagement de niveau de service.

Nous donnons ici une liste des propriétés classées par en suivant les paramètres définis par le W3C :

- **Le coût (ou le prix):** désigne le prix que le client doit payer pour chaque invocation du service.
- **La latence (aussi connue comme le temps de réponse):** désigne le temps pris par un service pour répondre à une requête.
- **Le débit:** le débit d'un service représente le nombre de demandes que le service est en mesure de traiter dans un intervalle de temps donné.
- **La fiabilité (aussi connue comme le taux d'exécution avec succès):** désigne la capacité du service à accomplir sa fonction correctement pendant une période de temps spécifiée. Il peut être mesuré par le temps moyen entre pannes.
- **La disponibilité:** ce paramètre désigne la probabilité que le service soit actif. Il peut être exprimé par le ratio : périodeactive/périodetotale. La période active représente le temps où le service est opérationnel en répondant aux demandes des clients. La période totale (périodeactive + périodeinactive), elle désigne la durée durant laquelle le client souhaite que le service soit actif.

- **La sécurité:** la sécurité d'un service web comporte différents aspects assurant que les échanges de messages entre le client et le service soient sécurisés. Plus précisément c'est un regroupement d'un ensemble de qualités à savoir : la confidentialité, le cryptage des messages et le contrôle d'accès.
- **La réputation:** c'est une mesure de la crédibilité du service. Elle dépend principalement des expériences d'utilisateurs finaux.

Ces paramètres peuvent être classés, selon leur mesure, en trois catégories [3]:

- Métriques annoncées par le fournisseur : métriques dont les valeurs ont été annoncées par le fournisseur. Exemple : le coût d'un service.
- Métriques évaluées par le consommateur : métriques dont les valeurs ont été données par l'utilisateur. Exemple la réputation.
- Métriques observables : métriques dont les valeurs ont été obtenues par la surveillance ou le test. Comme exemple : le temps de réponse ou la disponibilité.

1.4 SOA à base de service web

Il existe deux types d'architectures pour les services Web : la première est appelée architecture de référence, qui contient trois couches principales. La seconde architecture est plus complète, elle reprend les couches standards de la première architecture et ajoute d'autres couches plus spécifiques par dessus. C'est ce qu'on appelle une architecture étendue, ou encore Pile.

1.4.1 Architecture étendue

Une architecture étendue est constituée de plusieurs couches se superposant les unes sur les autres, d'où le nom de pile des services web. La Figure 1.2 décrit un exemple d'une telle pile. La pile est constituée de plusieurs couches. Chaque couche s'appuie sur un standard particulier. On retrouve, au-dessus de la couche de transport, les trois couches formant l'infrastructure de base décrite précédemment. [10]

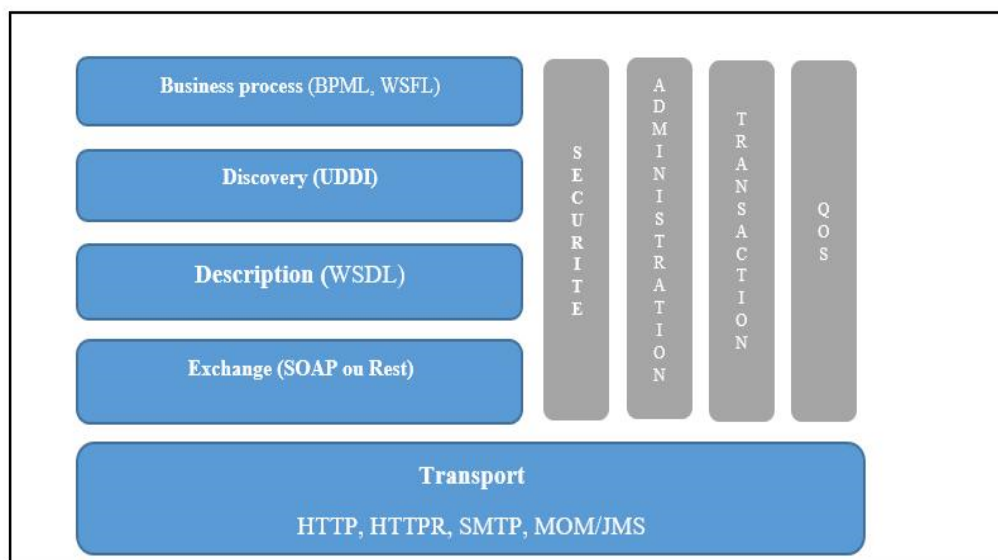


Figure (1.2) - Architecture en Pile des services web

Nous apportons une explication de la mise en relief des trois types de couches (Voir la Figure 1.2) [10]:

L'infrastructure de base (Discovery, Description, Exchange) : Ce sont les fondements techniques établis par l'architecture de référence. Nous distinguons les échanges des messages établis par SOAP, la description de service par WSDL et la recherche de services Web que les organisations souhaitent utiliser via le registre UDDI ;

Couches transversales (Security, Transactions, Administration, QoS) : Ce sont ces couches qui rendent viable l'utilisation effective des services Web dans le monde industriel ;

La couche Business Processus (Business Process) : Cette couche supérieure permet l'intégration de services web. Elle établit la représentation d'un « BusinessProcess » comme une composition de services Web. Grâce à cette couche, nous décrivons le flux de contrôle et le flux de données de ces compositions.

1.5 Standards de base de services Web

Une caractéristique du grand succès de la technologie des services Web est qu'elle repose sur des technologies standards. Dans cette section, nous présenterons trois standards de base, à savoir le protocole SOAP, le langage WSDL et l'annuaire UDDI. Ces trois normes sont toutes basées sur le langage XML.

1.5.1 Le protocole SOAP

1.5.1.1 Définition : SOAP est un standard proposé par W3C [6]. Il représente un protocole à la fois simple et léger destiné à l'échange d'informations. Ce dernier est de type RPC (Remote Procedure Call) d'échange de messages entre les Web services, standard ouvert et entièrement basé sur le langage XML pour assurer l'interopérabilité entre composants et les appels de procédures distantes (RPC). [11]

Ce protocole ne requiert aucune plate-forme spécifique, ni d'exigence particulière en ce qui concerne l'implémentation de l'application. SOAP peut cependant reposer sur d'autres protocoles de transport comme par exemple SMTP ou JMS (Java Message Service) mais HTTP est le plus populaire. Un message SOAP est un document XML encapsulé dans une enveloppe permettant d'organiser les informations d'une manière à être échangées entre partenaires. Il existe deux types de messages SOAP [11]:

- **Request :** il inclut le nom de la méthode à invoquer et la liste de ses paramètres.
- **Response:** il inclut le résultat de la méthode invoquée dans le message Request.

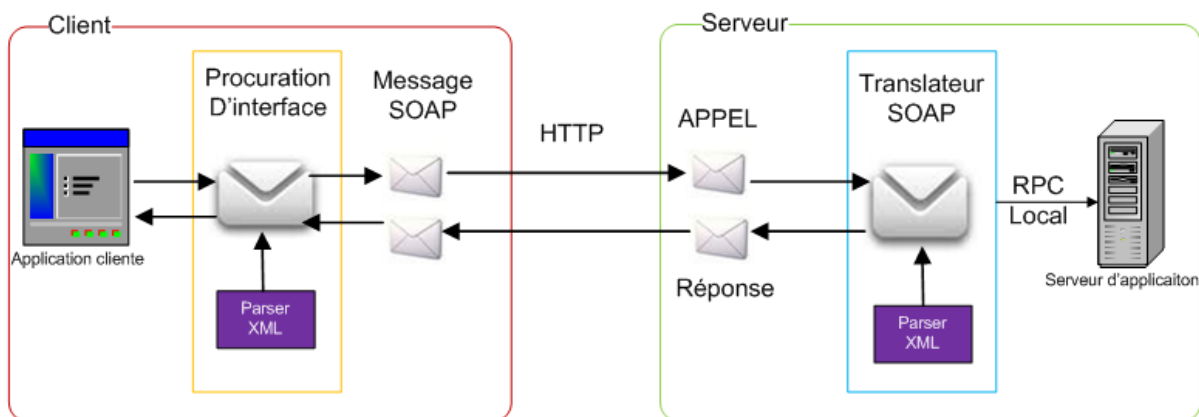


Figure (1.3) - Traitement d'un message SOAP

1.5.1.2 Structure d'un message SOAP

Un message SOAP présente une structure normalisée. Il est toujours constitué d'un élément racine (SOAP-ENV : envelope), qui contient un élément en-tête (SOAP-ENV : header) optionnel et un élément corps (SOAP-ENV : body) obligatoire. [9]

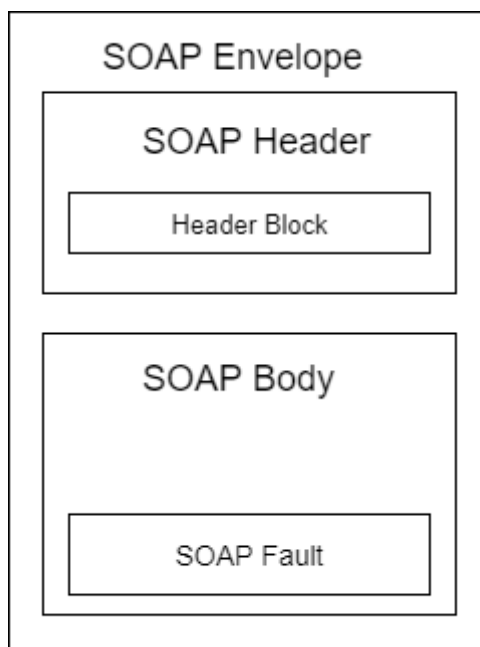


Figure (1.4) - Structure de message SOAP

- **Envelope:** C'est lui qui contient le message et ses différents sous-blocs. Il s'agit du bloc racine XML. Il peut contenir un attribut encoding Style dont la valeur est un URL vers un fichier de typage XML qui décrira les types applicables au message SOAP. [11]
- **Header :** C'est un bloc optionnel qui contient des informations de l'en-tête sur le message. S'il est présent, ce bloc doit toujours se trouver avant le bloc Body à l'intérieur du bloc Envelope. [11]
- **Body :** C'est le bloc qui contient le corps du message. Il doit absolument être présent de manière unique dans chaque message et être contenu dans le bloc Envelope. SOAP ne définit pas comment est structuré le contenu de ce bloc. Cependant, il définit le bloc qui peut s'y trouver. [11]
- **Fault:** Ce bloc est la seule structure définie par SOAP dans le bloc Body. Il sert à reporter des erreurs lors du traitement du message, ou lors de son transport. Il ne peut apparaître qu'une seule fois par message. Sa présence n'est pas obligatoire. [11]

SOAP RPC

SOAP RPC définit la convention d'utilisation de SOAP comme RPC et le format de message utilisé pour envoyer des requêtes ou envoyer des réponses. SOAP RPC est basé sur la spécification XML-RPC.

1.5.2 WSDL (Web Services Description Language)

WSDL (Web Services Description Language) est un langage de description fondé sur XML (Extensible Markup Language). Il a été soumis au W3C (World Wide Web Consortium) comme standard industriel pour la description des services Web. WSDL tire sa puissance de deux principes architecturaux principaux : la possibilité de décrire un ensemble d'opérations métier et la possibilité de séparer la description entre deux unités de base. Ces unités fournissent une description des opérations et les détails du mode d'intégration de l'opération et des informations qui lui sont associées. [8]

Un document WSDL définit des services en tant que collections de points de contact de réseau ou de ports. Dans WSDL, les définitions abstraites des points de contact (end points) et des messages sont séparées de leur déploiement réseau concret ou des liaisons de format de données. Ainsi, la réutilisation des définitions abstraites est possible : messages, qui sont des descriptions abstraites des données échangées et types de ports, qui sont des collections abstraites d'opérations. Les spécifications de format de données et de protocole concret pour un type de port particulier constituent une liaison réutilisable. Un port est défini en associant une adresse réseau à une liaison réutilisable et une collection de ports définit un service. C'est pourquoi, un document WSDL est composé de plusieurs éléments. [8]

La structure d'informations dans un fichier WSDL est la suivante :

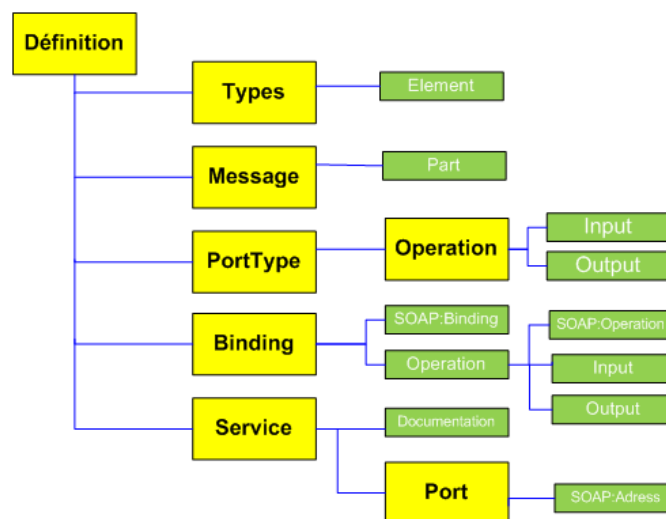


Figure (1.5) - Structure d'un document WSDL

Une description WSDL d'un service Web est faite sur deux niveaux, niveau abstrait et niveau concret [12]:

- 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 à un protocole concret (SOAP, HTTP, MIME, . . .). Un 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 de service, il ne prend donc pas en charge la comparaison de services.

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 [12].

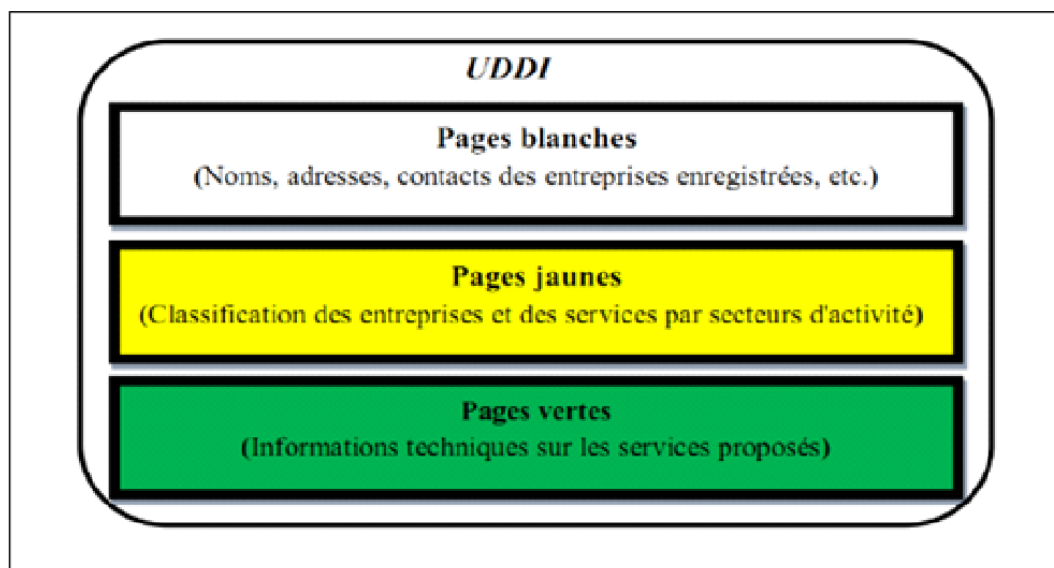


Figure (1.6) - les trois types de l'annuaire UDDI

Les informations qu'il contient peuvent être divisées en trois catégories :

Les Pages blanches comprennent des adresses, des informations de contact et des identifiants liés aux services Web.

Les Pages jaunes qui identifient les domaines d'activité liés aux services Web.

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

1.6 Conclusion

L'architecture orientée services impose un nouveau style de construction d'applications informatiques distribuées. Ce style se caractérise par plusieurs avantages, parmi lesquels l'interopérabilité est le principal avantage recherché. Dans ce chapitre, nous avons présenté les éléments de l'architecture et les technologies sous-jacentes des services Web tels que SOAP, WSDL et UDDI.

CHAPITRE 2 :

Logique Floue et Système d'inférence floue

Chapitre 2 : Logique Floue et Système d'inférence Floue

2.1 Introduction

Les mathématiques floues sont de plus en plus classées comme le terme général de la théorie floue, combinant plusieurs théories, qui sont la généralisation ou l'extension de leurs homologues classiques : la théorie des sous-ensembles flous étend la théorie des ensembles et la logique floue étend la théorie des ensembles flous. Logiquement, la théorie des quantités floues étend la théorie des nombres et des intervalles, la théorie des possibilités étend la théorie des probabilités, et plus généralement, la théorie de la mesure floue étend la théorie de la mesure. Ils visent tous à fournir des concepts, des techniques et des méthodes formelles et rigoureuses pour la collecte, la représentation et le traitement de connaissances et de données vagues.

Dans ce chapitre, nous présenterons brièvement les concepts de base de la logique floue d'un point de vue industriel, et les différentes applications de cette dernière, notamment dans le diagnostic industriel.

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. [13]

2.3 Le Concept De La Logique Floue

Assouplissant l'algèbre de Boole, la logique floue remplace la valeur de vérité d'une proposition à choisir dans {vrai,faux} par un degré de vérité, à choisir par exemple dans [0, 1]. En logique floue, il y a donc des degrés dans la satisfaction d'une condition.

Un exemple simple et ancien est la docimologie, c'est-à-dire l'évaluation des élèves d'une classe : dans chaque discipline, celui-ci est évalué par une note entre 0 (mauvais) et 20 (exceptionnel) et on estime le niveau de l'élève pour telle ou telle orientation par une pondération de ces notes avec divers coefficients selon l'orientation considérée [14].

2.4 Théorie Des Sous-ensembles Flous

2.4.1 Définition

Dans la logique booléenne, un élément appartient ou n'appartient pas à un ensemble donné: c'est le «tout ou rien». Zadeh (1965) constate que le plus souvent, les objets rencontrés dans le monde réel n'ont pas de critères précis d'appartenance. Il va tenter alors de sortir de cette logique booléenne en introduisant la notion d'appartenance pondérée. Il définit l'ensemble flou comme une classe d'objets avec un continuum de degrés d'appartenance à cette classe. Un tel ensemble est caractérisé par une fonction d'appartenance qui associe à chaque objet un degré d'appartenance compris entre zéro et un. [15]

Soit X un ensemble de référence et soit x un élément quelconque de X . Un ensemble flou A de X est défini comme l'ensemble des couples [15]:

$$A = \{(x, \mu_A(x)), x \in X\}$$

Avec

$$\mu_A : X \rightarrow [0, 1]$$

Ainsi, un ensemble flou A de X est caractérisé par une fonction d'appartenance qui associe, à chaque élément x de X un réel dans l'intervalle $[0, 1]$; $\mu_A(x)$ représente le degré d'appartenance de x à A . Ainsi, plus la valeur de $\mu_A(x)$ est proche de l'unité et plus le degré d'appartenance de x à A est élevé. [15]

Si on a [15]:

$\mu_A : X \rightarrow \{0, 1\}$ on retrouve le cas booléen, soit x appartient à A ($\mu_A = 1$) soit il n'appartient pas à A ($\mu_A = 0$).

2.4.2 Comparaison : la logique floue et 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 [16].

2.4.3 Les Operations de La Logique Floue

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 $\mu(A)$ et $\mu(B)$ leur fonction d'appartenance, on définit [13]:

A- L'OPERATEUR « ET »

L'opérateur logique correspondant à l'intersection d'ensembles est le « ET ». Le degré de vérité de la proposition « A ET B » est le minimum des degrés de vérité de A et de B;

$$\mu_{A \cap B}(x) = \min [\mu_A(x), \mu_B(x)] \quad \forall x \in U$$

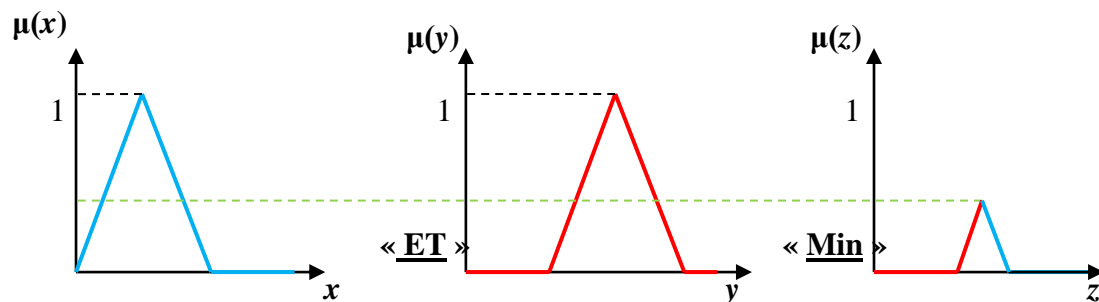


Figure (2.1) - l'intersection « ET » de deux ensembles flous

B- L'OPERATEUR « OU »

L'opérateur logique correspondant à l'union d'ensembles est le OU. Le degré de vérité de la proposition « A OU B » est le maximum des degrés de vérité de A et de B.

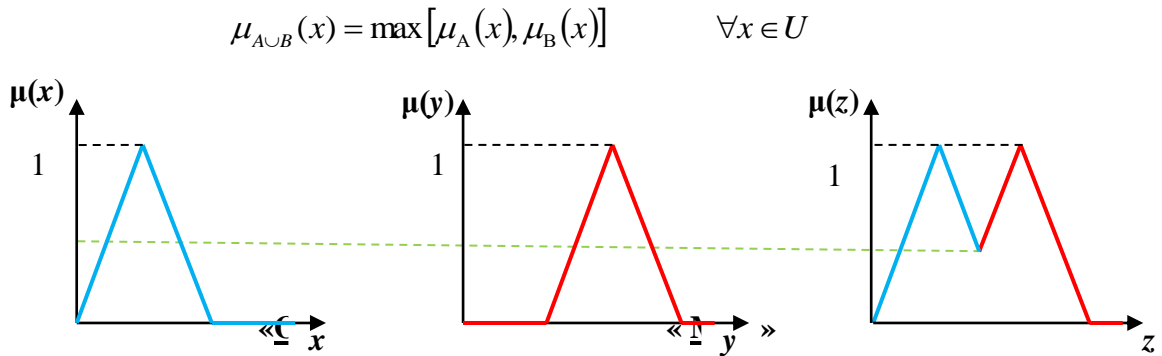


Figure (2.2) - l'union « OU » de deux ensembles flous

C- L'OPERATEUR « NON »

L'opérateur logique correspondant au complément d'un ensemble est la négation.

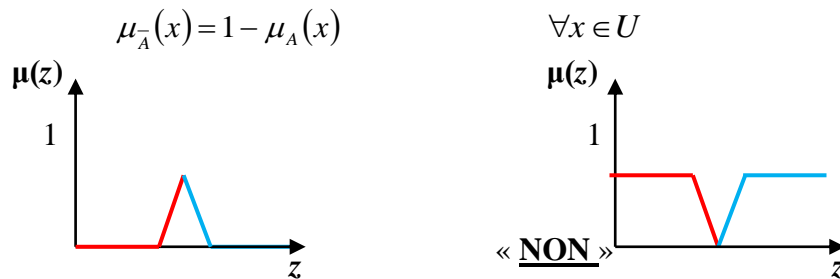


Figure (2.3) - la complément « NON » d'ensembles flous

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. [13]

2.4.4 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, beaucoup, souvent, chaud, froid, rapide, lent, grand, petit ...etc.". Ce genre d'expressions forme ce qu'on appelle des variables linguistiques de la logique floue [13].

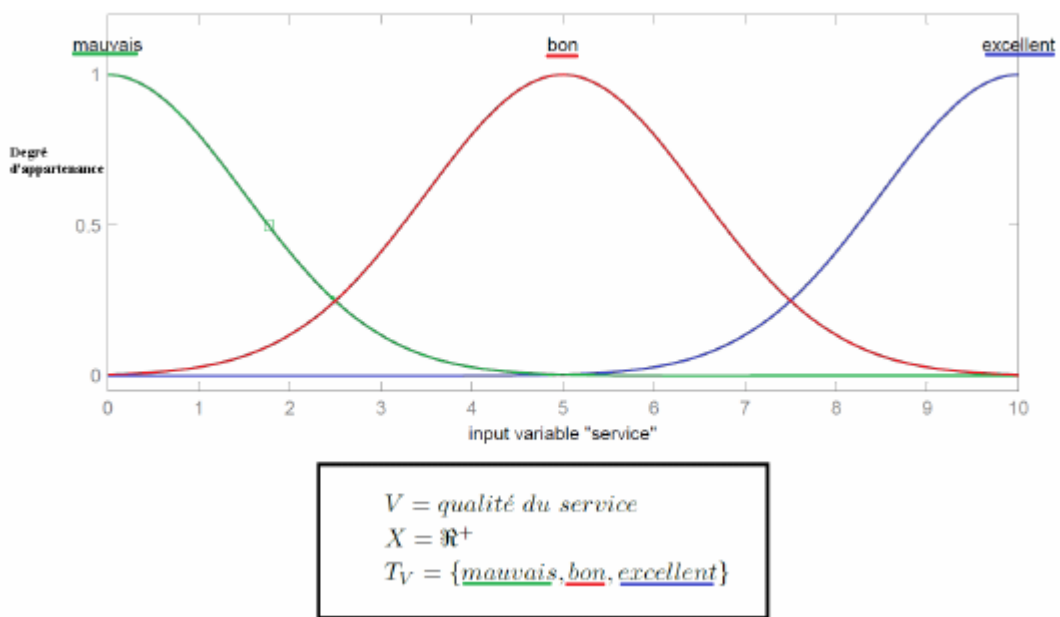


Figure (2.4) - Variable linguistique de la qualité du service

2.4.5 Les Fonctions d'appartenance

Chaque valeur linguistique est caractérisée par une fonction d'appartenance. Il y a deux façons pour définir la fonction d'appartenance des ensembles flous : numérique et fonctionnelle. Une définition numérique exprime le degré de fonction d'appartenance d'un ensemble flou comme un vecteur de nombres dont la dimension dépend du nombre d'éléments discrets dans l'univers du discours. Une définition fonctionnelle définit la fonction d'appartenance d'un ensemble flou dans une expression analytique où le degré d'appartenance pour chaque élément est calculé.

Certaines formes standard de fonctions d'appartenance sont couramment utilisées pour représenter les ensembles flous basés sur l'univers du discours. Les fonctions d'appartenance les plus couramment utilisées sont : (a) fonction S, (b) fonction π , (c) forme triangulaire, (d) forme trapézoïdale, (e) forme exponentielle et la forme gaussienne . La forme triangulaire et la forme trapézoïdale sont les plus utilisées pour déterminer le degré d'appartenance [17].

- **Fonction appartenance triangulaire:**

Les fonctions d'appartenance ont une forme triangulaire dont l'apparence précise est déterminée par les valeurs a, b, c comme le montre la figure(2.5). 'a' et 'c' sont les limites inférieure et supérieure des ensembles flous et 'b' est la moyenne de 'a' et 'c'. Pour un élément x, la fonction d'appartenance est définie comme suit :

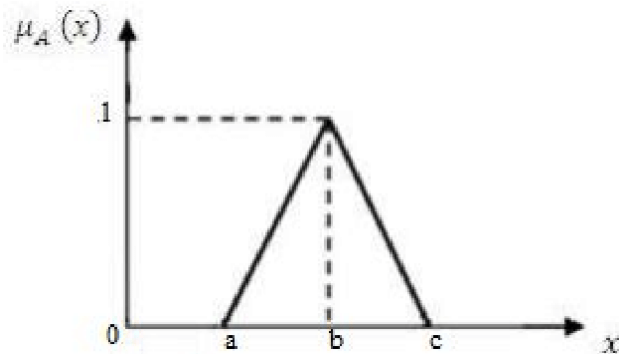


Figure (2.5) - Fonction d'appartenance floue triangulaire.

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

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

• **Fonction d'appartenance trapézoïdale:**

Une fonction d'appartenance floue trapézoïdale est illustrée à la figure (2.6). 'a' et 'd' sont les limites inférieure et supérieure des ensembles flous et la région comprise entre 'b' et 'c' a toujours une valeur de membre égale à un. La fonction d'adhésion pour un nombre flou trapézoïdal est donnée comme suit :

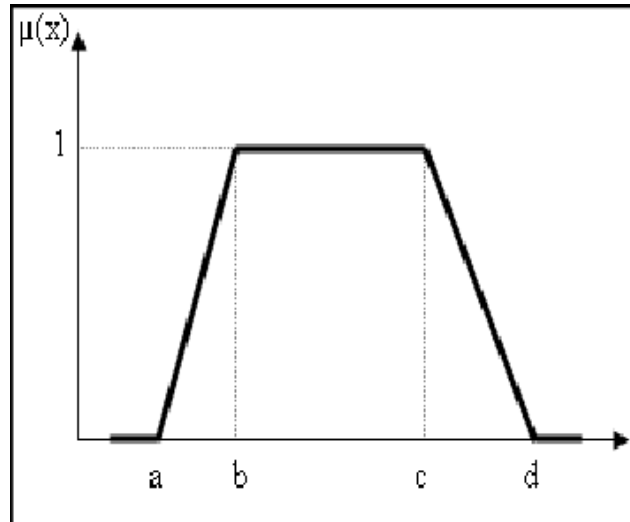


Figure (2.6) - Fonction d'appartenance floue trapézoïdale.

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

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

2.5 Structure D'une Commande Floue

2.5.1 Bases De Règles Et Définitions

Le schéma de la [Figure (2.7)] présente la Structure conventionnelle d'une commande floue.

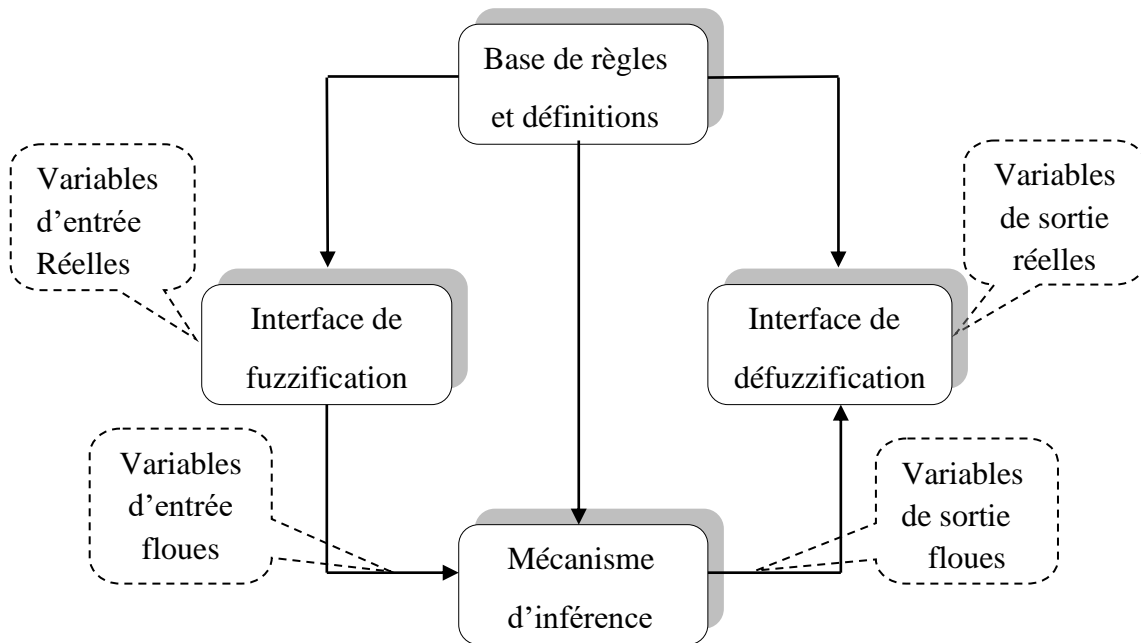


Figure (2.7) - la Structure d'une commande floue

A- Bases De Règles

Le but de la logique floue est de formaliser et de mettre en œuvre le raisonnement humain. À ce stade, il peut être classé dans le domaine de l'intelligence artificielle. L'outil le plus largement utilisé dans les applications de logique floue est la base de règles floues. La base de règles floues se compose de règles qui sont généralement utilisées en parallèle, mais peuvent également être liées dans certaines applications.

Une règle est du type : **SI** « prédicat » **ALORS** « conclusion ».

Par exemple : « Si vitesse est élevée et le feu est rouge **ALORS** freine fort ».

Les bases de règles floues sont les mêmes que les systèmes experts traditionnels, qui fonctionnent sur la base de connaissances issues de l'expertise humaine.

Une règle floue comporte trois étapes fonctionnelles « **fuzzification, d'inférence, défuzzification** ».

B- Quand Peut-On Utiliser Les Bases De Règles Floues ?

Lorsque les conditions suivantes sont remplies, la base de règles floues peut être sélectionnée pour résoudre le problème d'application :

- possibilité d'agir sur le procédé (contrôlabilité),
- existence d'une expertise ou d'un savoir-faire,
- possibilité de mesurer ou d'observer les grandeurs importantes (entrées et sorties),
- expertise qualitative (si elle est mathématique, l'automatique classique est à favoriser),
- expertise graduelle (si elle est booléenne, les systèmes experts sont plus adaptés).

2.5.2 Système D'inférence Floue

Le système d'inférence floue (SIF) est formé de trois blocs comme l'indique la [Figure (2.8)]. Le premier, bloc de fuzzification transforme les valeurs numériques en degrés d'appartenance aux différents ensembles flous de la partition.

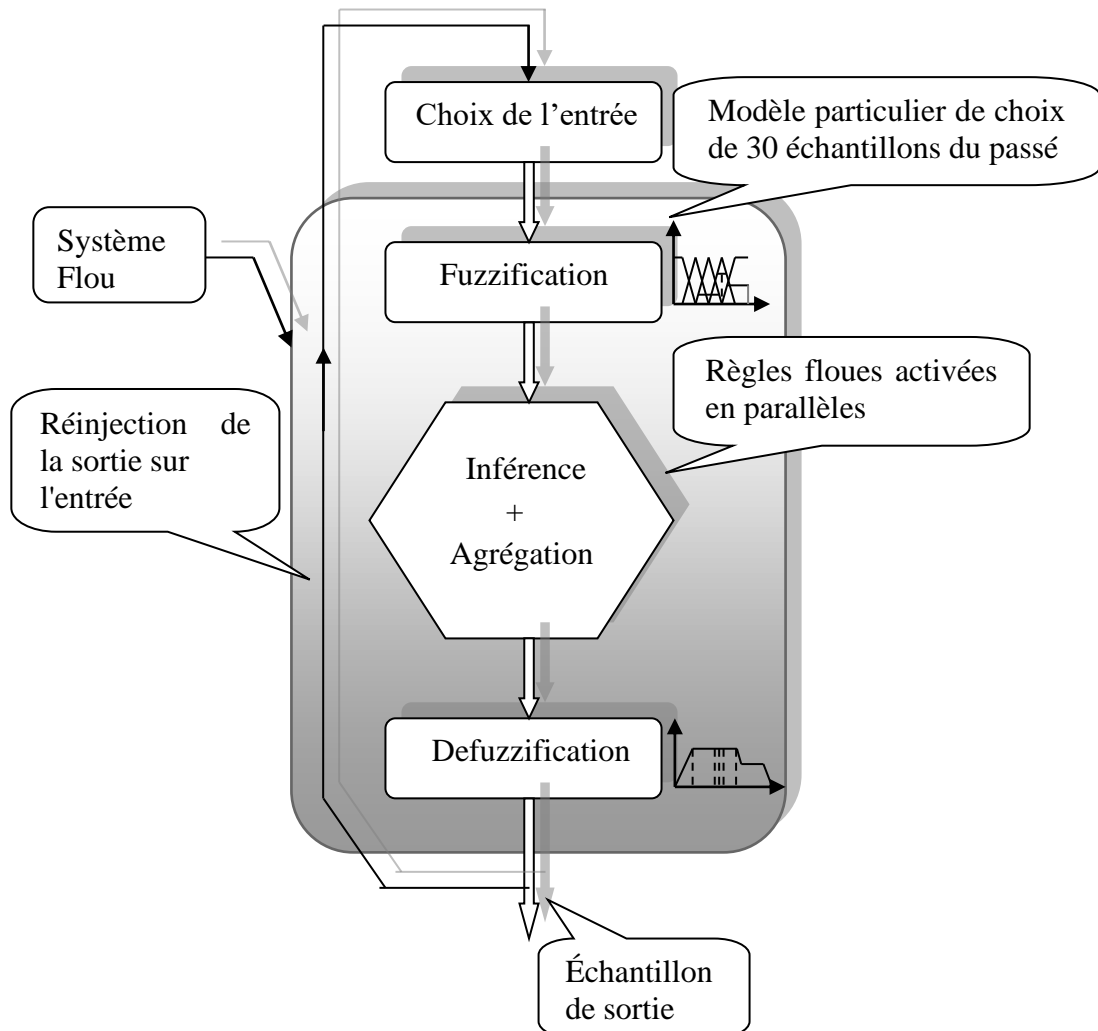


Figure (2.8) - Schémas en blocs un système d'inférence floue

A- Fuzzification

En évaluant la fonction d'appartenance, la valeur numérique est convertie en degré d'appartenance flou.

- Le processus qui consiste à transformer une grandeur numérique en un sous ensemble flou.
- Qualifier une valeur numérique avec un terme linguistique



✓ Comment fuzzifier?

Pour fuzzifier, il faut donner:

- L'univers du discours. i.e.: Plage de variations possibles de l'entrée considérée.
- Une partition en classe floue de cet univers.
- Les fonctions d'appartenances de chacune de ces classes.

B- Inférence

Le cycle de calcul du degré d'activation de toutes les règles de base et de tous les ensembles flous de variables linguistiques trouvés dans la conclusion de ces règles.

Pour le réglage par logique floue, l'une des méthodes suivantes est généralement utilisée:

- Méthode d'inférence max-min.
- Méthode d'inférence max-prod.
- Méthode d'inférence somme-prod.

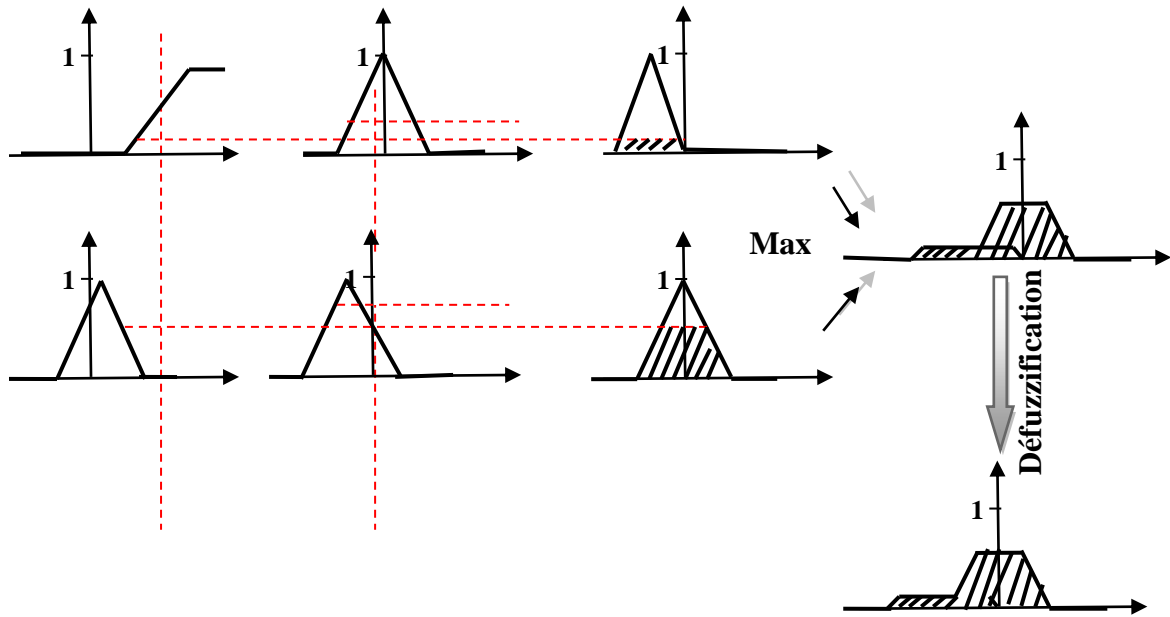


Figure (2.9) - Schémas d'inférence floue

C- Mécanisme D'inférence De MAMDANI

Le mécanisme d'inférence le plus couramment utilisé est celui dit « de Mamdani ». Il représente une simplification du mécanisme plus général basé sur « l'implication floue ». Seules les bases de règles « de Mamdani » sont utilisées.

✓ Principe

Une base de règles floues de Mamdani comprend donc des règles linguistiques faisant appel à des fonctions d'appartenance pour décrire les concepts utilisés.

Le mécanisme d'inférence comprend les étapes suivantes :

✓ Fuzzification

La fuzzification consiste à évaluer les fonctions d'appartenance utilisées dans les prédicats des règles.

✓ Degré d'activation

Le degré d'activation d'une règle est l'évaluation du prédicat de chaque règle par combinaison logique des propositions du prédicat. Le « ET » est réalisé en effectuant le minimum entre les degrés de vérité des propositions.

✓ **Implication**

Le degré d'activation de la règle permet de déterminer la conclusion de la règle, c'est l'implication. Il existe plusieurs opérateurs d'implication, mais le plus utilisé est le « minimum ». L'ensemble flou de conclusion est construit en réalisant le minimum entre le degré d'activation et la fonction d'appartenance, sorte d' « écrêtage » de la fonction d'appartenance de conclusion.

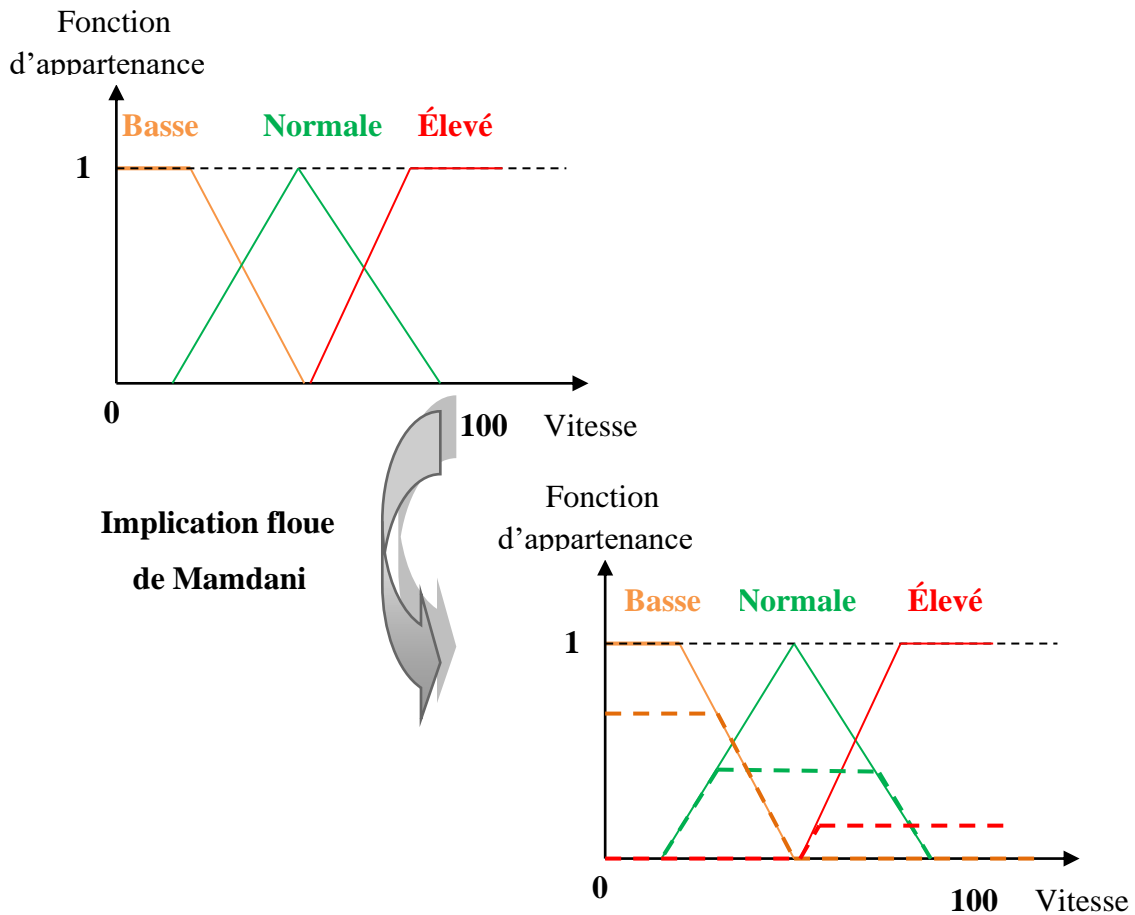


Figure (2.10) - Schémas d'implication floue de Mamdani

✓ **Agrégation**

L'ensemble flou global de sortie est construit par agrégation des ensembles flous obtenus par chacune des règles concernant cette sortie.

On considère que les règles sont liées par un « OU » logique, et on calcule donc le maximum entre les fonctions d'appartenance résultantes pour chaque règle.

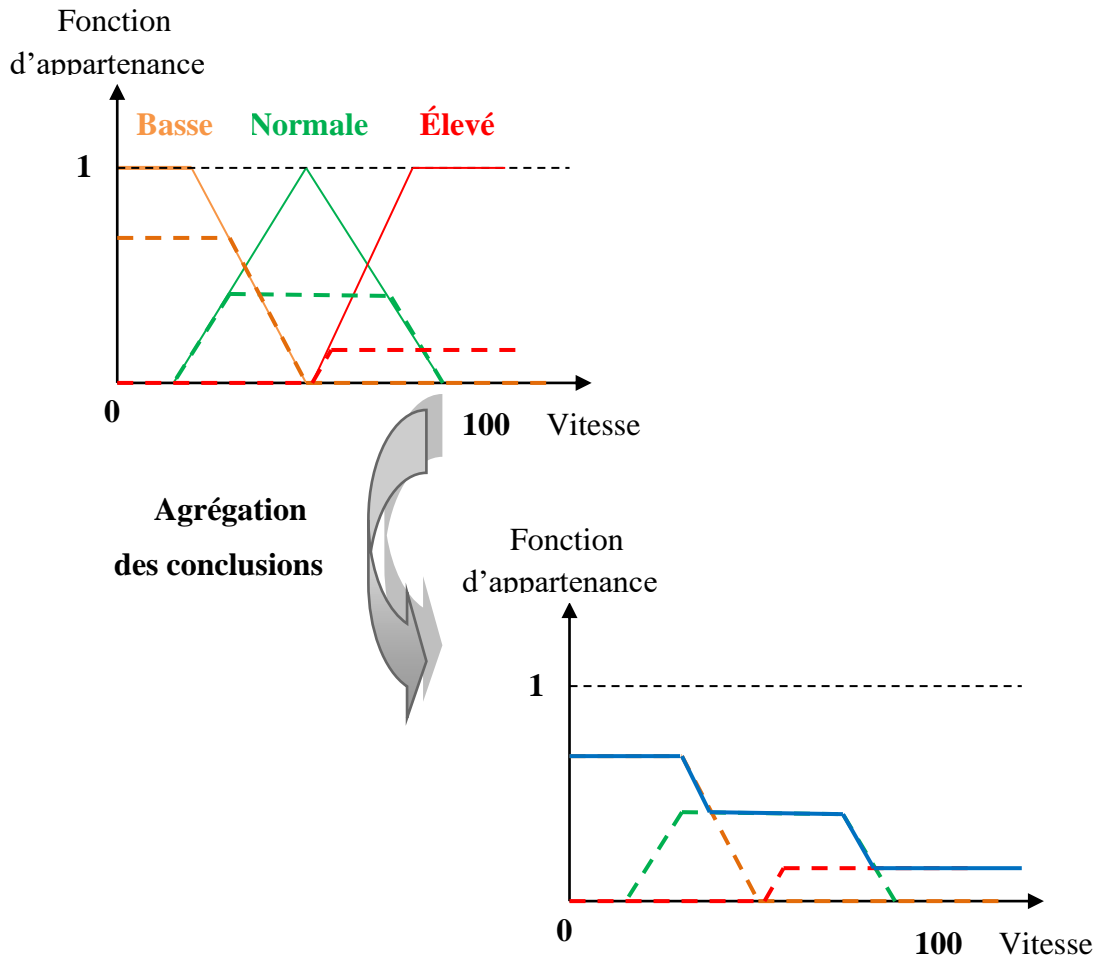


Figure (2.11) - Schémas d'agrégation des conclusions

D- Défuzzification

Transformation, après inférence, d'un ensemble flou d'une variable linguistique de sortie en valeur numérique.

A la fin de l'inférence, l'ensemble flou de sortie est déterminé mais il n'est pas directement utilisable pour donner une information précise à l'opérateur ou commander un actionneur. Il est nécessaire de passer du « monde flou » au « monde réel », c'est la défuzzification. Les méthodes les plus couramment utilisées sont:

- La méthode de centre de gravité
- La méthode de maximum.
- La méthode des surfaces.
- La méthode des hauteurs.

➤ **Défuzzification par centre de gravité**

La méthode de défuzzification la plus utilisée est celle de la détermination du centre de gravité de la fonction d'appartenance résultante $\mu_{RES}(x)$. Dans ce contexte, il suffit de calculer l'abscisse x^* . La figure (2.12) montre le principe de défuzzification.

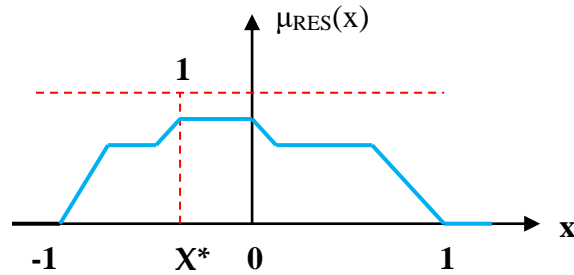


Figure (2.12) - Défuzzification par centre de gravité

Il existe plusieurs méthodes, la plus souvent rencontrée étant le calcul du « centre de gravité » de l'ensemble flou.

L'abscisse de centre de gravité peut être déterminée à l'aide de la relation générale:

$$CG = \frac{\int_{-1}^1 x \mu_{RES}(x) dx}{\int_{-1}^1 \mu_{RES}(x) dx} \qquad CG = \frac{\sum_{i=1}^n x_i \mu_{RES}(x_i)}{\sum_{i=1}^n \mu_{RES}(x_i)}$$

➤ **Défuzzification par valeur maximum**

La défuzzification par centre de gravité exige en général une envergure de calcul assez importante. Par conséquent, il sera utile de disposer d'une méthode de défuzzification plus simple.

Comme signal de sortie x^* , on choisit l'abscisse de la valeur maximale de la fonction d'appartenance résultante $\mu_{RES}(x)$. Comme le montre la figure (2.13)

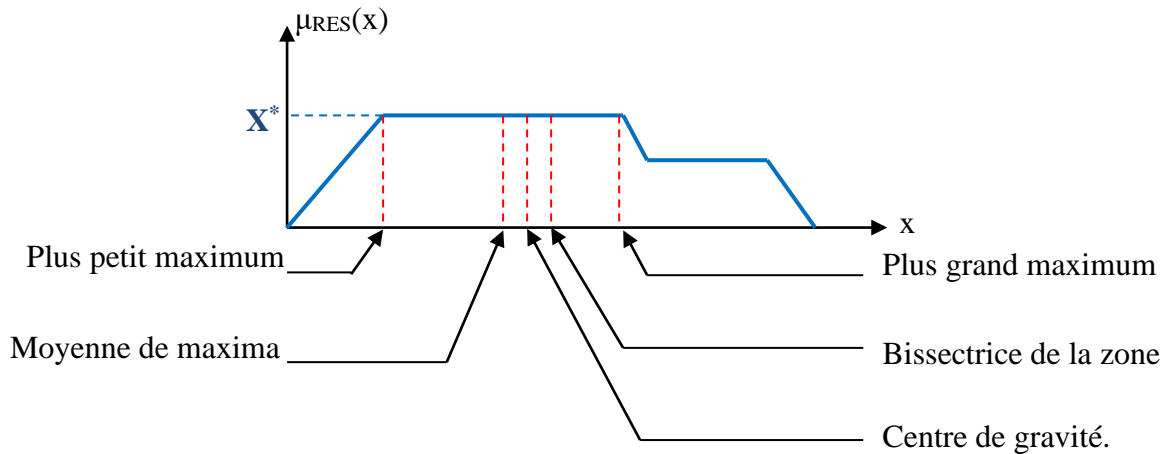


Figure (2.13) - Défuzzification par valeur maximum

2.6 Conclusion

Les outils fournis par la logique floue permettent de modéliser des phénomènes en quelque sorte proches du raisonnement humain. Le tout ou rien au-delà de l'ordinateur introduit de la flexibilité et rend les outils flous utiles dans de nombreux domaines.

Nous avons présenté dans ce chapitre les concepts de base de SIF, le concept de variable linguistique. Nous avons constaté que les SIF peuvent effectivement raisonner sur la base d'informations inexacts et interpréter leurs décisions, mais ne peuvent pas obtenir automatiquement les règles qu'ils ont utilisées pour prendre ces décisions.

CHAPITRE 03 :

Modélisation et Implémentation

Chapitre 3 : Modélisation et Implémentation

3.1 Introduction

Le nombre de services web offrant la même fonctionnalité est en constante augmentation, donc la tâche de sélectionner un service web parmi plusieurs services similaires est une tâche très délicate pour le consommateur de service car généralement, les services fournissant la fonctionnalité en question seront retournés dans l'ordre de leur inscription dans le registre UDDI en question et les approches de sélection prennent en compte le besoin du consommateur dans la sélection des Web Services mais elles ne prennent pas en compte d'autres contraintes telles que les qualités fonctionnelles et non fonctionnelles. Pour remédier à ce genre de problème, il a été proposé de prendre en compte les critères de qualité de service et plus précisément lors du processus de sélection.

Là encore, l'appréciation du service web par le consommateur n'est pas toujours précise et parfois floue et ambiguë, d'où la nécessité de traduire cette appréciation dans une logique analogue à la logique humaine et d'adopter des critères de qualité de service pour aider le consommateur dans le choix de ses propres services. Par conséquent, la logique floue peut être appliquée pour prendre en charge une représentation imprécise des contraintes de QoS et présenter les préférences des utilisateurs en tant que propriétés de QoS avec une présentation floue car elles sont mieux adaptées à l'interprétation des termes linguistiques.

3.2 Qualité de service

La qualité de service peut être définie comme la comparaison entre les attentes des clients et les services réellement fournis. Il est défini comme une combinaison de plusieurs critères, qui peuvent être considérés comme des critères de sélection lors de la sélection d'un service satisfaisant aux contraintes imposées par plusieurs services Web découverts.

3.2.1 Paramètres de qualités de services

Le choix des services Web basés sur la qualité de service (QoS) implique de sélectionner les services Web en fonction des exigences de QoS non fonctionnelles qui peuvent le mieux répondre aux besoins des utilisateurs parmi les services Web découverts.

Il existe des métriques pour quantifier les services Web. Ceux-ci se reflètent dans la qualité du service. De plus, la QoS est un aspect non fonctionnel du service. Ces valeurs permettent de sélectionner les services liés à la demande de l'utilisateur.

Plusieurs critères de qualité de service sont décrits dans différents travaux et les propriétés de QoS les plus représentatives sont présentées comme suit [18]:

- **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 ce travail nous avons choisi d'utiliser 3 Critères de services Web non fonctionnels lors du processus de sélection (Prix, disponibilité, Temps de Réponse).

3.3 La sélection des services Web basée sur la logique floue

Initialement, l'utilisateur envoie sa requête fonctionnelle au registre UDDI qui contient les noms des listes de services Web et leurs paramètres QoS. En réponse à cette demande, une liste de services similaires est produite. L'extraction de QoS est effectuée pour chaque service Web afin de produire une matrice de services Web floue et leur QoS correspondante. Une fois que la matrice floue est prête, elle est transmise au moteur d'inférence floue qui utilise la base de règles floues pour produire un poids pour chaque service Web. En fin de compte, ces valeurs de score sont classées pour fournir le meilleur service à recommander au client.

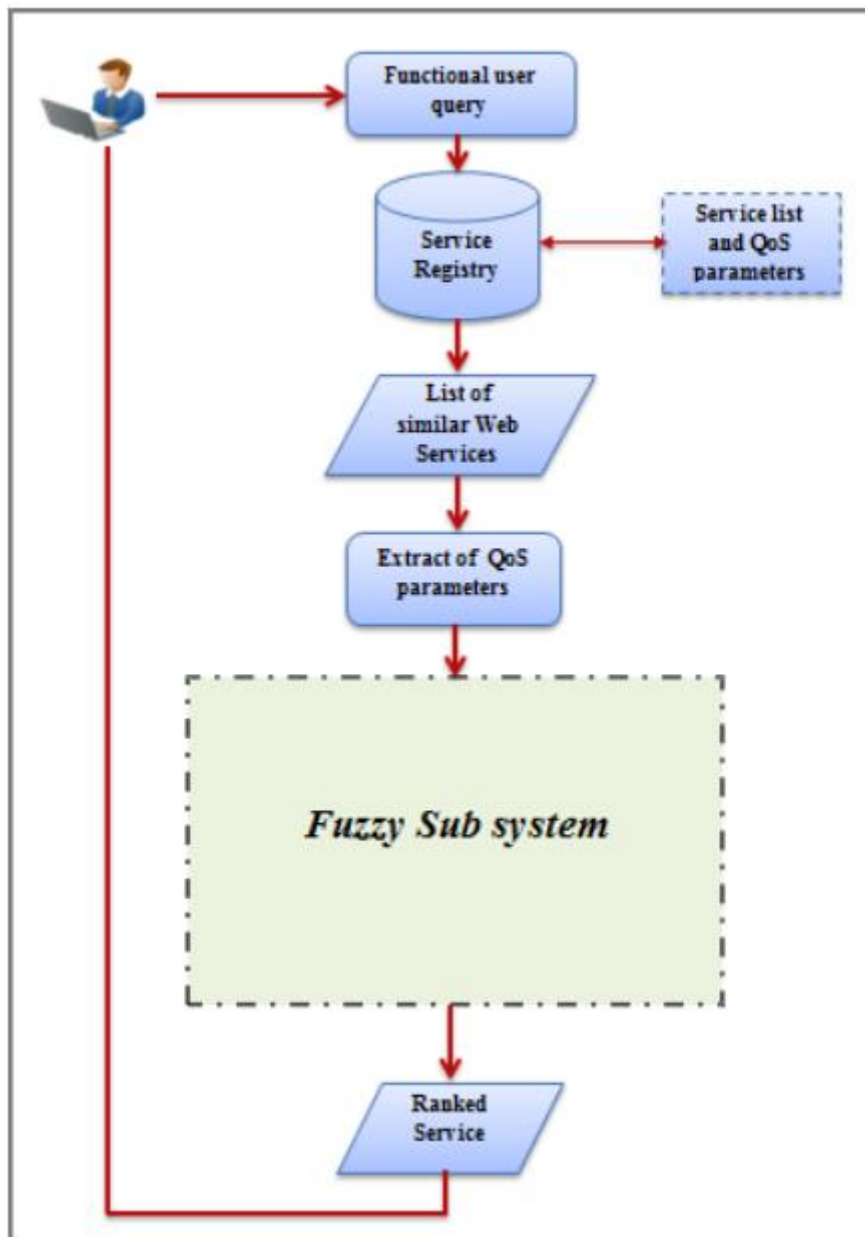


Figure (3.1) - Architecture de la sélection des services Web basée sur la logique floue[4].

3.4 Sous-système flou

Ce sous-système est décrit comme suit (figure.3.2) :

Il a en entrée les valeurs quantitatives des paramètres de qualité. il construit une matrice qui a des services Web en ligne et des colonnes les critères de qualité appropriés. Ces valeurs nettes sont transmises au module de fuzzification qui, avec l'aide d'experts humains et avec le choix d'une fonction d'appartenance donnée, procède à la fuzzification pour donner comme résultat une matrice floue.

Ensuite, cette matrice est reçue par le moteur d'inférence floue qui, avec l'exploitation de la base de règles floues produite pour chaque service web, aboutit à un terme flou nommé poids.

Ce poids est redirigé vers le module de défuzzification qui à l'aide d'une méthode de défuzzification produit une valeur quantitative caractérisant le service web en question.

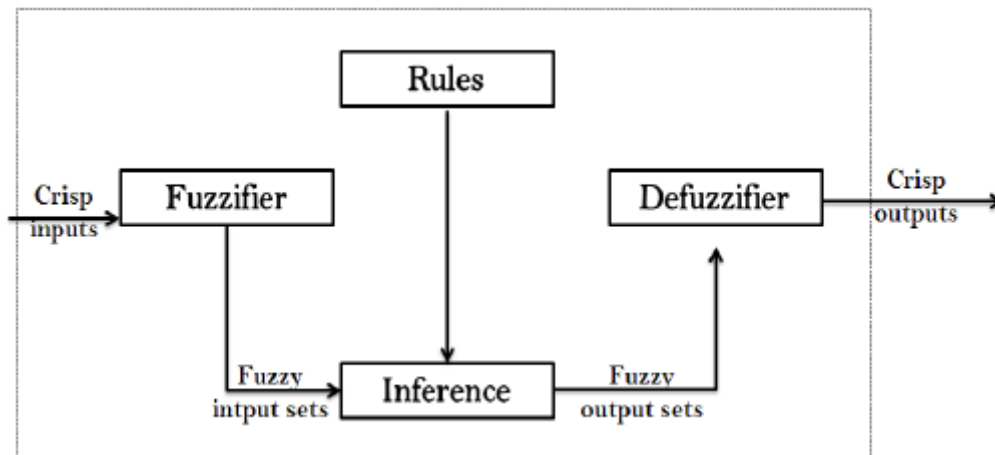


Figure (3.2) – Architecture de l'approche proposée

3.4.1 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) en valeurs quantitatives. Pour chaque critère, nous pouvons créer plusieurs fonctions d'appartenance selon le critère dont nous disposons. Si les services web que nous voulons tester répondent aux critères : Prix, disponibilité et temps de réponse : chacune de ces données aura plusieurs fonctions d'appartenance. Par exemple : Si nous voulons transformer Le Prix en variable linguistique. On peut trouver plusieurs variables linguistiques qualifiant ces données numériques : Cheap, Acceptable et Expensive. Même principe pour les deux autres critères : la disponibilité peut être présentée par les variables linguistiques : Low, Medium, Good, VeryGood. et le Temps de Réponse peut être présentée par les variables

linguistiques : Low, Medium, High, VeryHigh. Puisque nous avons choisi la fonction d'appartenance triangulaire pour présenter nos données linguistiques, le tableau suivant présente les valeurs attribuées de critère Disponibilité:

Terme Linguistique	Nombre flou triangulaire
Disponibilité	
Low	[0 0 25]
Medium	[15 30 50]
Good	[35 50 70]
VeryGood	[60 80 100]

Tableau 3.1 - Terme linguistique de Critère Disponibilité

3.4.2 Moteur d'inférence floue

L'entrée du moteur d'inférence floue est une matrice floue résultant de la fuzzification dont les valeurs linguistiques correspondant à chaque service web.

	q1	q2	qn
S1	a11	a12	a1n
S2	a21	a22	a2n
·
Sm	an1	an2	anm

Tableau 3.2 - Matrice de dépendance.

Où s_1, s_2, \dots, s_m sont les services Web, q_1, q_2, \dots, q_n sont les paramètres de qualité de service et x_{ij} est une valeur floue présentant la valeur de qualité q_j pour le service s_i pour chaque service Web, ses données sont transmises au moteur d'inférence floue qui utilise la base de règles d'inférence pour son raisonnement. Chaque règle est décrite en fonction des connaissances dont elle dispose. Pour son fonctionnement, il applique chaque règle aux variables linguistiques calculées à l'étape Fuzzification et le résultat de cette étape est une valeur floue caractérisant chaque service web.

3.4.2.1 Base de règles floues

Ces règles ont la forme (si X alors Y), elles sont créées par le concepteur et seront exploitées par le moteur d'inférence flou pour produire un résultat flou qui sera traduit plus tard pour donner une valeur quantitative. Les opérateurs les plus utilisés dans l'évaluation des règles sont l'union traduite par MAX et l'intersection traduite par MIN.

3.4.3 Défuzzification

Fuzzification nous aide à évaluer les règles, et après avoir combiné les règles, il faut maintenant produire un chiffre net comme sortie [19].

La technique la plus populaire est la méthode de centre de gravité, elle est exprimée comme [19]:

$$x = \frac{\sum_{i=1}^n m^i w^i}{\sum_{i=1}^n m^i}$$

Où :

- x = La sortie défuzzifiée.
- m^i = L'appartenance de la sortie de chaque règle.
- w^i = Le poids associé à chaque règle.

Cette méthode est utilisée car elle est plus rapide, plus facile et donne un résultat assez précis.

3.5 Détails d'implémentation

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
Disponibilité	
Low	[0 0 25]
Medium	[15 30 50]
Good	[35 50 70]
VeryGood	[60 80 100]

Tableau 3.3 – Paramètre Disponibilité en représentation floue

Terme Linguistique	Nombre flou triangulaire
Prix	
Cheap	[2 2 8]
Acceptable	[6 10.5 15]
Expensive	[12.8 16 20]

Tableau 3.4 – Paramètre Prix en représentation floue

Terme Linguistique	Nombre flou triangulaire
Temps de Réponse	
Low	[2 2 45]
Medium	[30 60 90]
Good	[70 90 120]
VeryGood	[100 130 150]

Tableau 3.5 – Paramètre Temps de Réponse en représentation floue

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

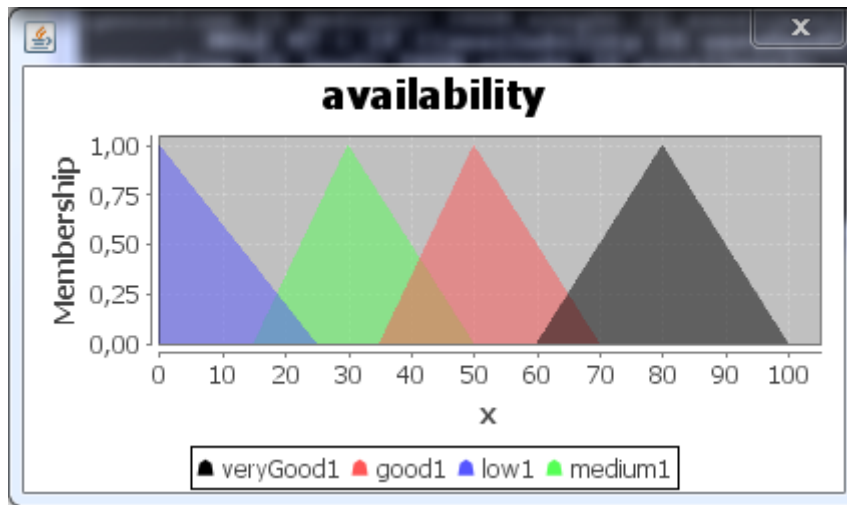


Figure (3.3) – Variable linguistique Disponibilité

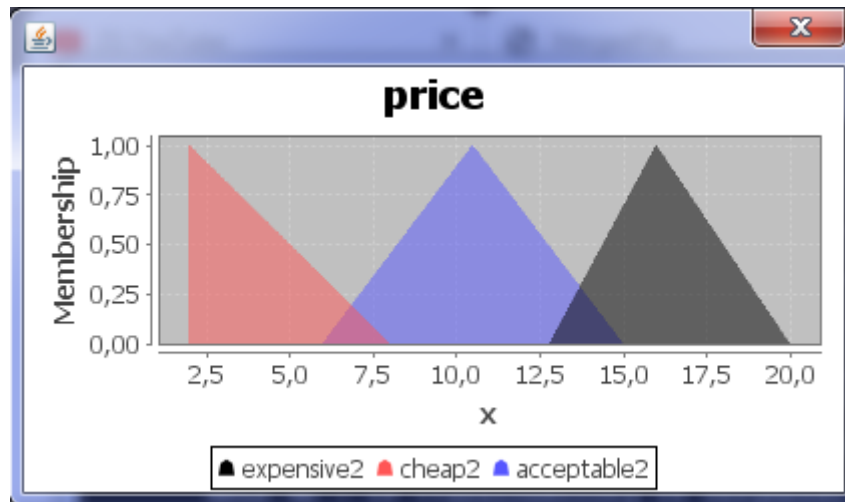


Figure (3.4) – Variable linguistique Prix

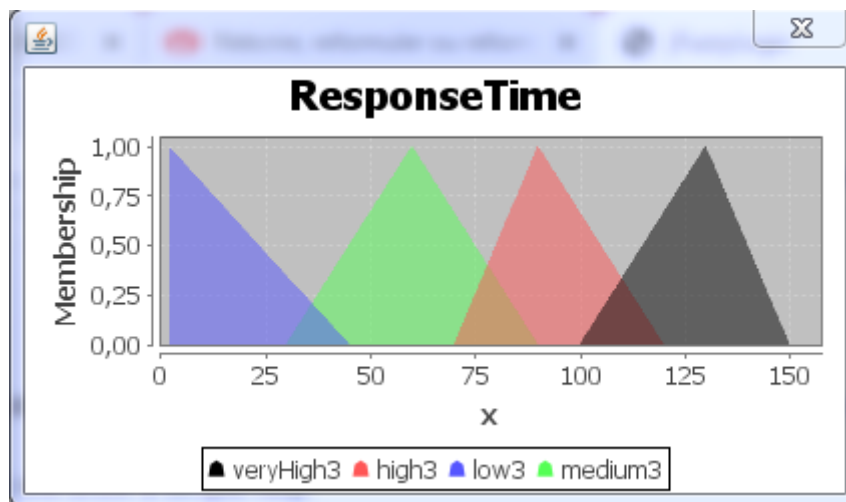


Figure (3.5) – Variable linguistique Temps de Réponse

Dans cette expérience, nous avons écrit un ensemble de règles qui seront utilisées par le moteur d'inférence floue. Dans ce qui suit, nous présentons quelques règles :

1. **IF** availability **IS** low **AND** price **IS** expensive **AND** ResponseTime **IS** veryHigh **THEN** wieht **IS** low.
2. **IF** availability **IS** medium **AND** price **IS** expensive **AND** ResponseTime **IS** medium **THEN** wieht **IS** medium.
3. **IF** availability **IS** good **AND** price **IS** acceptable **AND** ResponseTime **IS** medium **THEN** wieht **IS** good.
4. **IF** availability **IS** veryGood **AND** price **IS** acceptable **AND** ResponseTime **IS** low **THEN** wieht **IS** excellent.

3.5.1 Algorithme de sélection floue

L'algorithme de sélection de services Web similaires est représenté comme suit :

Entrée: La requête contient des paramètres fonctionnels

Sortie: Meilleur Services Web

- 1: Obtenir la requête de l'utilisateur
- 2: Retrouvez la liste pour la recherche de services similaires dans le registre UDDI.
- 3: Extraire les propriétés non fonctionnelles (QoS) du service.
- 4: Passer chaque QoS au Sous-système flou
- 5: En choisissant un type de fonction d'appartenance (par exemple méthode d'appartenance triangulaire, trapézoïdale ou gaussienne, traduisez les valeurs nettes des paramètres QoS en valeurs floues, dans ce travail nous avons utilisé la méthode d'appartenance triangulaire.
- 6: Pour chaque service web, le moteur flou calcule les poids QoS en utilisant la base de règles floues d'inférence.
- 7: Défuzzifier les valeurs de chaque service Web en utilisant la méthode du centre de gravité pour obtenir valeurs nettes de chaque service Web
- 8: Meilleur Services Web obtenus

3.6 Implémentation et expérimentation

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

- Utilisateur (Fournisseur et Client)
- webService

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

#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/> 1	service_id	int(11)			No	None	AUTO_INCREMENT
<input type="checkbox"/> 2	service_type	varchar(20)	utf8_bin		No	None	
<input type="checkbox"/> 3	service_name	varchar(20)	utf8_bin		No	None	
<input type="checkbox"/> 4	availability	double			No	None	
<input type="checkbox"/> 5	price	double			No	None	
<input type="checkbox"/> 6	ResponseTime	double			No	None	

Figure (3.6) – la table " webService".

3.6.1 Scenario pour un seul service web

Comme entrée en a 3 valeurs de critères QoS:

S1 : disponibilité: 45.0 , Prix: 13.5 , Temps de Réponse: 57.0

Pour la Fuzzification en utilise pour chaque Variable Linguistique (VL) une méthode (Figure 3.7):

```
double membership_availability_veryGood1(double x) {
    if ( x <= 60.0 )      return 0.0;
    if ( x > 100.0 )     return 0.0;
    if ( x <= 80.0 )     return 0.0 + ( 1.0 - 0.0 ) * ( ( x - 60.0 ) / ( 80.0 - 60.0 ) );
    if ( x <= 100.0 )   return 1.0 + ( 0.0 - 1.0 ) * ( ( x - 80.0 ) / ( 100.0 - 80.0 ) );
    return 0.0;
}
```

Figure (3.7) – Méthode de Fuzzification pour VL Disponibilité VeryGood.

Ce tableau est le résultat de la Phase de fuzzification:

	availability_low1	availability_medium1	availability_good1	availability_veryGood1
S1	0.0	0.25	0.6666666666666666	0.0

Tableau 3.6 – la matrice flou de la variable disponibilité de S1

	price_cheap2	price_acceptable2	price_expensive2
S1	0.0	0.33333333333333337	0.21874999999999983

Tableau 3.7 – la matrice flou de la variable Prix de S1

	ResponseTime_low3	ResponseTime_medium3	ResponseTime_high3	ResponseTime_veryHigh3
S1	0.0	0.9	0.0	0.0

Tableau 3.8 – la matrice flou de la variable Temps de Réponse de S1

Les règles utilisées pour ce service sont :

- **IF** availability **IS** medium1 **AND** price **IS** acceptable2 **AND** ResponseTime **IS** medium3 **THEN** waight **IS** good2;
- **IF** availability **IS** medium1 **AND** price **IS** expensive2 **AND** ResponseTime **IS** medium3 **THEN** waight **IS** medium2;
- **IF** availability **IS** good1 **AND** price **IS** acceptable2 **AND** ResponseTime **IS** medium3 **THEN** waight **IS** good2;
- **IF** availability **IS** good1 **AND** price **IS** expensive2 **AND** ResponseTime **IS** medium3 **THEN** waight **IS** medium2;

Pour l'inférence on utilise pour chaque Règle une méthode (Figure 3.8):

```
// RULE 26 : IF ((availability IS good1) AND (price IS expensive2)) AND (timeOut IS medium3) THEN wieght IS medium2;
double degreeOfSupport_26 = 1.0 * ( ruleConnectionMethod_and(ruleConnectionMethod_and
    (availability_good1 , price_expensive2) , timeOut_medium3) );
if( degreeOfSupport_26 > 0 ) {
    for (int i = 0 ; i < 1000 ; i++ ) {
        double x = 0.0 + i * 0.001;
        double membership = membership_wieght_medium2(x);
        double y = ruleActivationMethod_min( degreeOfSupport_26 , membership );
        defuzzify_wieght[i] += ruleAccumulationMethod_max( defuzzify_wieght[i], y );
    }
}
```

Figure (3.8) – Méthode D'inférence de la règle 26.

Après on doit défuzzifier les valeurs floue par la Méthode centre de Gravité (Figure 3.9):

```
// Defuzzify
private void defuzzify(){
    double sum_wieght = 0.0;
    double wsum_wieght = 0.0;
    for (int i = 0; i < 1000 ; i++){
        double x = 0.0 + i * 0.001;
        sum_wieght += defuzzify_wieght[i];
        wsum_wieght += x * defuzzify_wieght[i];
    }
    wieght = wsum_wieght / sum_wieght;
}
```

Figure (3.9) – Méthode de Défuzzification pour VL Disponibilité VeryGood.

Cette dernière étape nous donne la valeur Coefficient pour ce Service Web:

$$\text{Coeff_S1} = 0.445804246859231$$

3.6.2 Présentation des interfaces graphiques

On se limitera à quelques interfaces:

- **Interface d'accueil**

Les Utilisateurs Soit Un Fournisseur ou Un Client doit d'abord authentifier l'identité afin de pouvoir réaliser ses taches. La figure (3.10) montre l'interface d'inscription du L'utilisateur.

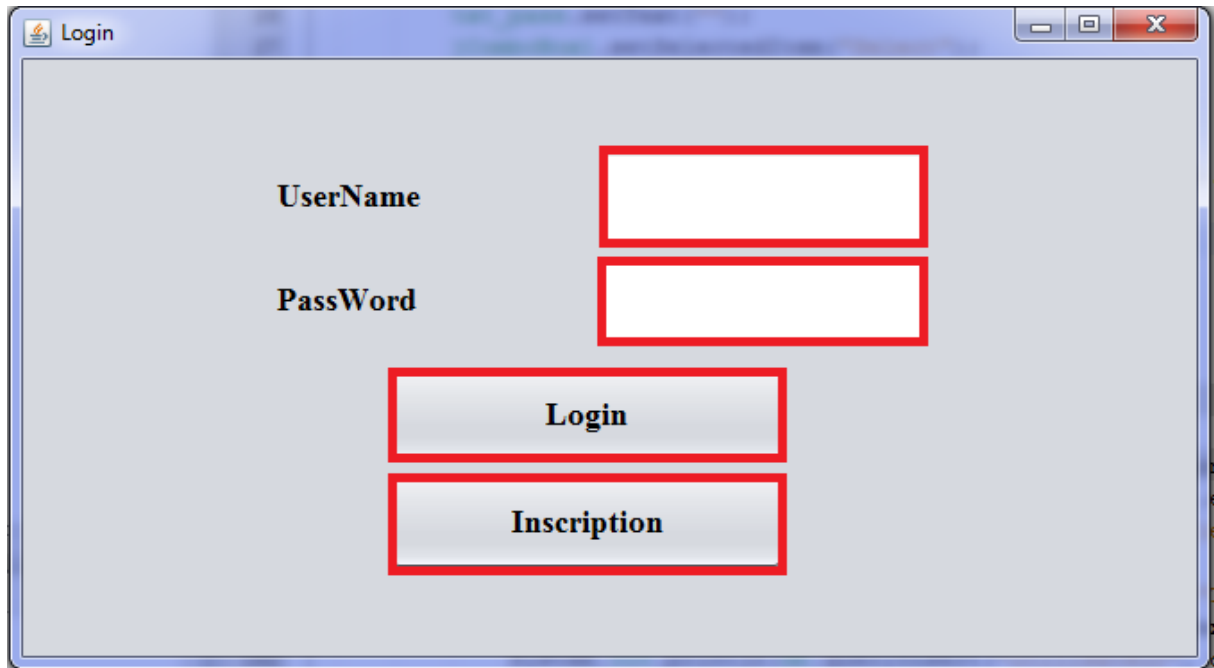
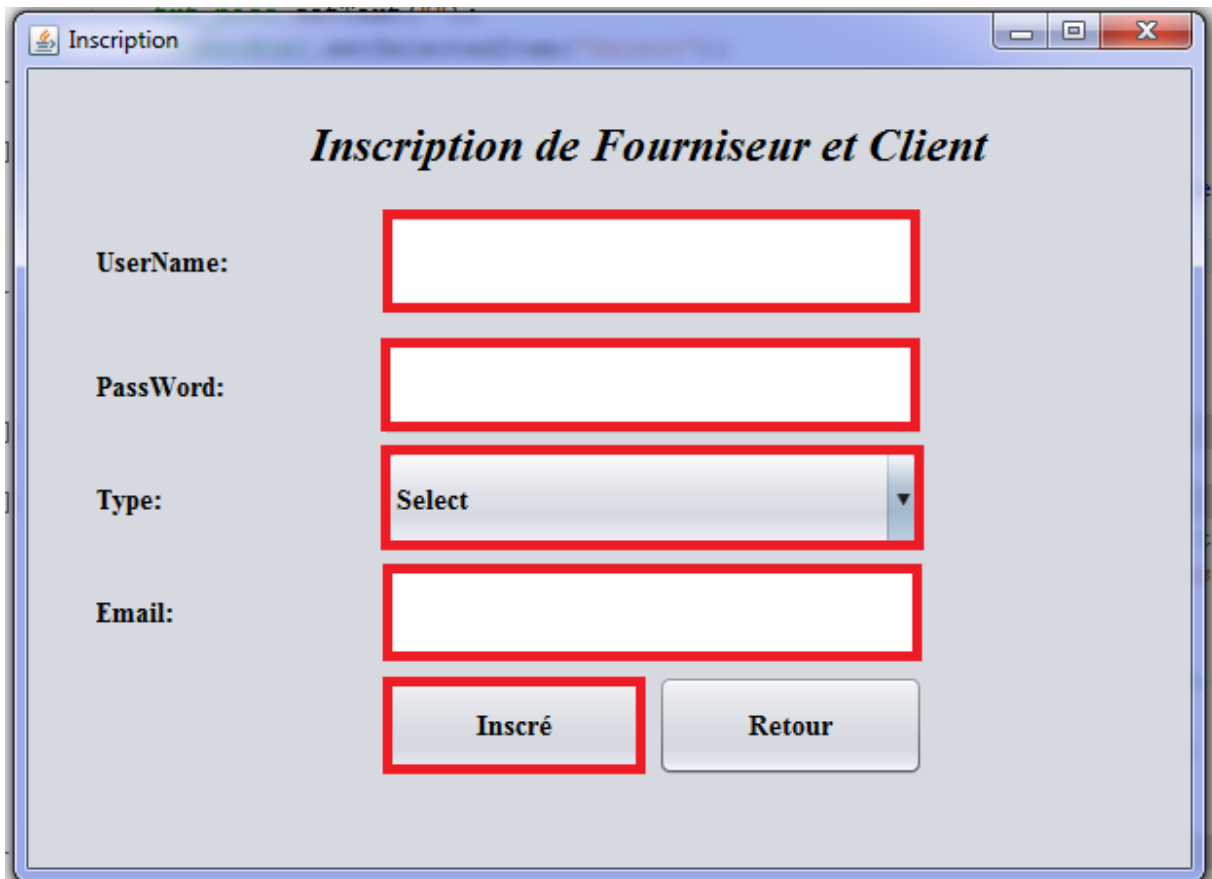


Figure (3.10) – Login du Fournisseur et Client

- **Interface d’Inscription du Fournisseur et Client**

Les Utilisateurs Soit Un Fournisseur ou Un Client doit s’inscrire afin de pouvoir réaliser ses tâches, il doit sélectionner le type soit Fournisseur ou Client. La figure (3.11) montre l’interface d’inscription du L'utilisateur.

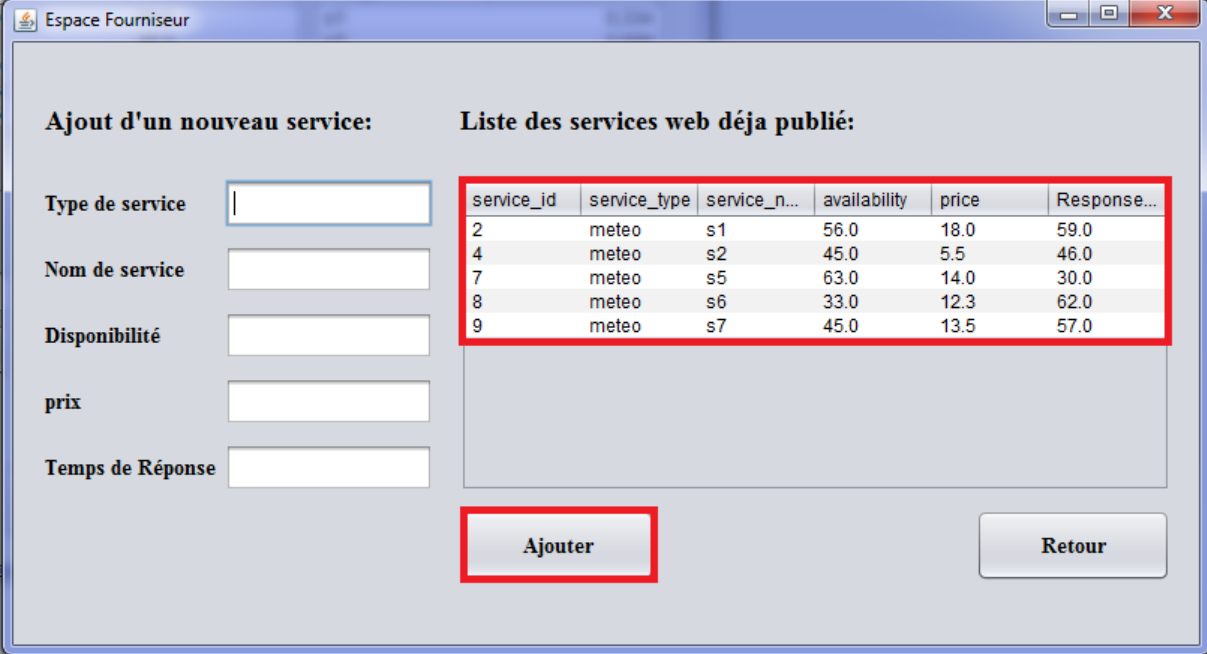


The image shows a software window titled "Inscription". The window has a title bar with standard minimize, maximize, and close buttons. The main content area has a light gray background and is titled "Inscription de Fournisseur et Client" in a bold, italicized font. Below the title, there are four input fields, each with a red rectangular border: a text box for "UserName:", a text box for "PassWord:", a dropdown menu for "Type:" with "Select" as the current selection, and a text box for "Email:". At the bottom of the form, there are two buttons: "Inscré" and "Retour". The "Inscré" button is highlighted with a red border.

Figure (3.11) – Inscription du Fournisseur et Client

- **Interface d'Espace Fournisseur**

Les Fournisseurs Peuvent ajouter un ou plusieurs service web. La figure (3.12) montre l'interface d'Espace Fournisseur permettant la description des services à publier.



The screenshot shows a window titled "Espace Fournisseur" with two main sections:

- Ajout d'un nouveau service:** A form with five input fields: "Type de service", "Nom de service", "Disponibilité", "prix", and "Temps de Réponse".
- Liste des services web déjà publié:** A table with 6 columns: "service_id", "service_type", "service_n...", "availability", "price", and "Response...". The table contains 5 rows of data.

Below the table, there are two buttons: "Ajouter" and "Retour".

service_id	service_type	service_n...	availability	price	Response...
2	meteo	s1	56.0	18.0	59.0
4	meteo	s2	45.0	5.5	46.0
7	meteo	s5	63.0	14.0	30.0
8	meteo	s6	33.0	12.3	62.0
9	meteo	s7	45.0	13.5	57.0

Figure (3.12) – Espace Fournisseur

- **Interface d'Espace Client**

L'interface montrée dans la figure (3.13) permet au Client de sélectionner les services satisfaisant à ses besoins. Elle présente dans le tableau à droite la liste des services web similaires pour un besoin donné (meteo) et dans le tableau à droite les résultats d'exécutions de sous système flou, à savoir la liste des services web(SW) avec les coefficients calculé. Le meilleur service web (dont le coefficient est le plus élevé) est calculé en cliquant sur « Best Web Service ».

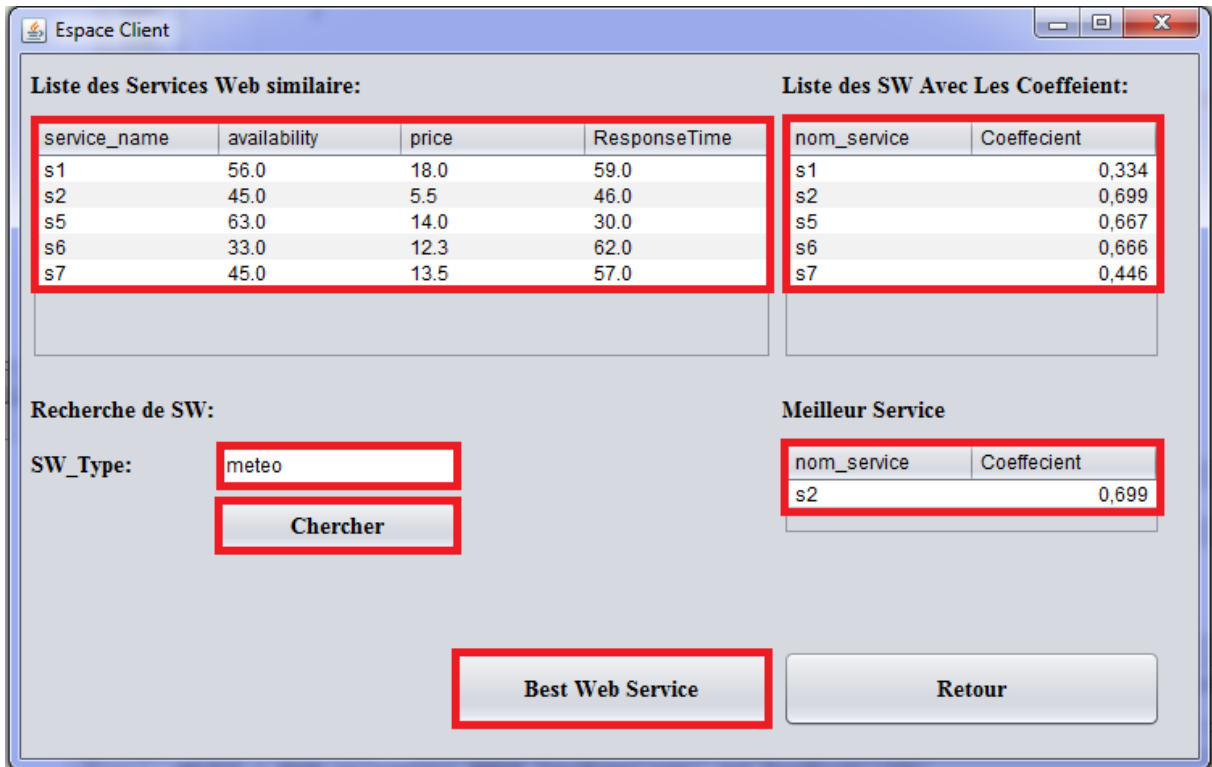


Figure (3.13) – Espace Client

3.7 Conclusion

Dans ce chapitre, nous avons présenté un modèle de sélection de services Web basé sur la logique floue qui prendra en compte les données linguistiques du consommateur, puis sélectionnera les services appropriés et restituera les services Web souhaités. Ce modèle a été illustré par un exemple de sélection de services Web composé de cinq services Web et de trois critères de qualité (Prix, disponibilité, Temps de Réponse). Cette approche permet aux clients d'obtenir un classement dynamique des services Web disponibles en fonction de divers critères de QoS.

Conclusion générale:

Afin d'atteindre notre objectif, nous avons commencé notre travail par une introduction détaillée aux concepts de base des services Web, qui sont des technologies émergentes et prometteuses pour le développement, le déploiement et l'intégration d'applications sur Internet. Ce sont les technologies de base pour développer une architecture orientée services. Ces architectures sont de plus en plus courantes sur le Web.

Autour de ce point, nous proposons une méthode de sélection basée sur la logique floue, car la logique floue est similaire au raisonnement humain et utilise des informations approximatives et des informations incertaines pour prendre des décisions.

Nous avons étudié en détail les différents Critères de qualité de service utilisées pour sélectionner les services Web. Un algorithme de sélection automatique de service Web basé sur des règles floues est proposé. Enfin, sur la base de règles et de plusieurs Critères majeures de qualité de service (telles que la disponibilité, le temps de réponse, le coût) pour mettre en œuvre le processus de sélection des services Web.

Ce travail nous permet de considérer plusieurs points intéressants, mais dans un premier temps, il serait intéressant d'améliorer le processus de sélection des services Web en fonction de Critères de qualité afin de mieux s'assurer que les besoins des utilisateurs sont satisfaits. Certaines améliorations des outils proposés sont encore une perspective de ce travail (par exemple, rendre les outils aussi paramétrés que possible en termes de catégories de services et de nombre de Critères de qualité).

Bibliographie:

- [1]: 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.
- [2]: Site web, URL : <http://www.service-architecture.com/>. Consulté le 12/05/2021.
- [3]: Chouhal Ouahiba. Calcul Orienté Service. 2021. Cours Master. Université Abbes Laghrour de Khenchela.
- [4]: Okba Zakar, Belouaar Houcine, Nadia Kabachi. A new model for web services selection based on fuzzy logic. 2018. Article. Université Mohamed Khider Biskra.
- [5]: Site web, URL : web.maths.unsw.edu.au/ Consulté le 21/05/2021.
- [6]: W3C World Wide Web Consortium; "Web Services Architecture"; W3C Working Group Note 11; February 2004; <http://www.w3.org/TR/ws-arch>
- [7]: O'Sullivan Justin, Edmond David, Ter Hofstede Arthur, What's in a service Distrib. Parallel Databases ?, 12(2-3): 117–133, 2002.
- [8]: IBM.BPEL4WS (version1.1), <http://www.ibm.com/>, 2003. Consulté le 19/05/2021.
- [9]: BENKHALED Hamid Naceur, REGRAGUI SidAhmed Tidjani. SELECTION DE WEB SERVICES A BASE DE LA QUALITE DE SERVICE. 2016. MEMOIRE DE MASTER. Université Dr. Tahar Moulay SAIDA.
- [10]: MERZOUG Mohammed. La découverte et la sélection des services web. 2018. Thèse de doctorat. Université abou bekr belkaid tlemcen
- [11]: Remadelia Abdellah, Matallah Imane. Composition des Services Web. 2016. MEMOIRE DE MASTER. Université de Djilali BOUNAËMA Khemis Miliana
- [12]: Mihi Europa. Une approche de sélection des services Web basée sur la logique floue. 2019. MEMOIRE DE MASTER. Université Mohamed Khider Biskra.
- [13]: BOUZID Houssam Eddine, BENMERIEM 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.
- [14]: Site Web. https://fr.wikipedia.org/wiki/Logique_floue. Consulté le 21/05/2021.
- [15]: Article : Abel Tiemtore, Théorie des ensembles flous : une application à l'assurance indicielle au Burkina Faso. 2012. Université Ouaga II, Burkina Faso.
- [16]: Belounnas Adil. Commande d'un système photovoltaïque par contrôleur Génétique Neuro-Flou. 2020. Univrsity badji mokhtar – annaba.

[17]: Haitam Alilouch. Conception et analyse des performances d'un contrôleur flou pour un système de réglage de tension automatique. 2020. Université du québec en abitibi témiscamingue

[18]: Idris Igoulalene, Développement d'une approche floue multicritère d'aide à la coordination des décideurs pour la résolution des problèmes de sélection dans les chaînes logistiques, Thèse de doctorat.