REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université "Yahia FARES" Médéa
Faculté des Sciences
Département de Mathématiques et Informatique



projet de fin d'étude

Présenté pour l'obtention du diplôme
De master en informatique

Spécialité : Ingénierie des Systèmes et Technologie de Web

Thème

Proposition et Construction d'une Ontologie Pour la Composition Dynamique Des Services Web

Proposé et dirigé par :

Mr. TOUBAL El Bahi

préparé et présenté par :

** BOUGUEDRA Mounira*

**MOKHBAT Narimen*

Promotion 2020

Remerciement

Tout d'abord, on remercie dieu le tout puissant de nous avoir donné la santé et la volonté d'entamer et de terminer ce mémoire.

Un profond respect et un remerciement particulier à notre encadreur Mr. TOUBAL El Bahi pour ses conseils et sa disponibilité, sa gentillesse et la patience qu'il a manifesté dans ce projet.

Nous adressons aussi nos remerciements à tous les enseignants et au corps administratif de la faculté de Sciences de Médéa.

Un profond remerciement pour les membres de jurées qui ont accepté d'évaluer ce travail.

Dédicace

Avec un énorme plaisir, un cœur ouvert et une immense joie, que je dédie mon travail à mon très cher, respectueux grand-père.

A l'homme de ma vie, celui qui s'est toujours sacrifié pour me voir réussir, à toi mon père.

A la lumière de mes jours, la source de mes efforts, ma vie et mon bonheur, maman que j'adore.

A mes sœurs, mon frère et toute ma grande famille.

A ma très chère copine « Lilia ».

A mon binôme « Mounira ».

« NARIMEN »

Dédicace

Avec un énorme plaisir, un cœur ouvert et une immense joie, que je dédie mon travail à mes chers parents.

A mes chères sœurs et mes chers frères.

A mes amies « Amira, Asma, Fatima, Amani »

A mon binôme « Narimen »

A tous mes collègues

« MOUNIRA »

Résumé

Résumé :

Avec l'évolution de la quantité d'information sur le web, le web classique ne peut donner des résultats adéquats seulement avec le traitement syntaxique, c'est pour cela il a été proposé d'étendre le web classique vers le web sémantique. Le service web sémantique est la fusion du web sémantique avec les services web, cette technologies permet aux agents logiciels de découvrir, sélectionner, composer et exécuter automatiquement sans besoin de l'intervention de l'être humain.

Ce mémoire consiste à faire la découverte et la composition des services web selon une approche sémantique des ontologies, pour permettre de sélectionner les meilleurs services web selon les concepts de l'ontologie et de faire la composition et avoir un bon composite.

Mots clé : Ontologie, services web, découverte, composition, web sémantique.

الملخص:

مع تطور كمية المعلومات على الويب , يمكن للويب الكلاسيكي أن يعطي نتائج مناسبة فقط من خلال المعالجة النحوية , لهذا السبب تم اقتراح توسيع الويب الكلاسيكي إلى الويب الدلالي. خدمة الويب الدلالية هي اندماج الويب الدلالي مع خدمات الويب , تسمح هذه التقنية لوكلاء البرامج بالاكتشاف , الاختيار , التأليف و التنفيذ تلقائيا دون الحاجة الى العنصر البشري.

يتكون هذا العمل من اكتشاف خدمات الويب وتركيبها مع بعضها البعض وفقًا لمنهج دلالي للانطولوجيا , لتمكين اختيار أفضل خدمات الويب وفقًا لمفاهيم الأنطولوجيا و تكوين خدمات الويب والحصول على مركب جيد.

الكلمات المفتاحية : خدمات الويب , الويب الدلالي , الأنطولوجيا , لاكتشاف , التأليف

Résumé

Abstract:

With the evolution of the amount of information on the web, the classic web cannot give adequate results only with syntactic processing, that's why it has been proposed to extend the classic web to the semantic web. The semantic web service is the fusion of the semantic web with web services, this technology allows software agents to discover, select, compose and execute automatically without the need of human beings.

This dissertation consists in discovering and composing web services according to a semantic approach of ontologies, to allow the selection of the best web services according to the ontology concepts and to make the composition and have a good composite.

Keywords: Ontology, web service, semantic web, web service, select, discover, composition.

Sommaire :

Introduction générale…….1

Chapitre 01 : le web sémantique et les ontologies

I. Introduction ……..03

II. Le web sémantique ……03

II.1.Qu'est ce qu'un web sémantique ……..03

II.2.L'objectif du web sémantique ……..04

II.3.Les principes du web sémantique ……..05

II.4.Architecture du web sémantique ...…….07

II.4.1.Niveau nommage ou adressage ……..07

II.4.2.Niveau syntaxique ……..07

II.4.3.Niveau sémantique ……..07

II.5.Application du web sémantique ……..08

II.5.1.E-commerce…….08

II.5.2.E-Learning……...09

III. Les ontologies ……09

III.1.Antécédent ……09

III.2.Qu'est ce qu'une ontologie…….10

III.2.1.Ontologie pour le web……10

III.2.2.Ontologie et web service ……..10

III.3.Les composants d'une ontologie……11

III.3.1.Concept……..11

III.3.2.Terme ……11

III.3.3.Notion ……11

III.3.4.Relation ……11

III.3.5.Axiomes ……12

III.4.Les types d'ontologie……12

III.4.1.Ontologie de représentation de connaissance ……12

III.4.2.Ontologie de haut niveau ……..12

III.4.3.Ontologie générique ……..12

III.4.4.Ontologie de domaine ……...13

III.4.5.Ontologie de tache……..13

III.4.6.Ontologie d'application……..13

III.5. Les critères d'ontologie ……13

III.5.1.La clarté ……13

III.5.2.La cohérence ……13

III.5.3.Extensibilité ……..13

III.5.4.Une déformation d'encodage minimal……...13

III.5.5.Un engagement ontologique minimal……14

III.6.Domaine d'application d'ontologie ……..14

III.7.Les outils de construction d'ontologie ……14

III.7.1.Les outils dépendants de formalisme de représentation…….14

III.7.2.Les outils indépendants de formalisme de représentation……..15

III.8.Langages de spécification d'ontologie……...16

III.8.1.SHOE (Simple HTML Ontology Extension) ……16

III.8.2.RDF (Resource Description Framework)……..16

III.8.3.OWL (Ontology Web Language).....16

IV. Conclusion ……17

Chapitre 02 : La composition des services web

I. Introduction……..18

II. Les services web……18

II.1.Historique ……..18

II.2.Définition ……19

II.3.Architecture orienté service et service web ……..19

II.4.Fonctionnement des services web ……21

II.4.1. Invocation……..21

II.4.2. Découverte ……21

II.4.3. Description ……21

II.5. Description en couche des services web ……22

II.5.1. Couche de transport ……...23

II.5.2. Couche de communication ……23

II.5.3. Couche de description de service ……..24

II.5.4. Couche de découverte ……24

II.5.5. Couche de composition ……25

III. La découverte des services web ……25

III.1. Introduction ……25

III.2. Critère de découverte ……..25

III.2.1. Critère d'architecture ……..26

III.2.2. Critère d'automatisation ……26

III.2.3. Critère de matchmaking……..26

IV. La composition des services web……..26

IV.1. Définition……..26

IV.2. Cycle de vie d'une composition des services web ……..27

IV.2.1. La phase de définition……..27

IV.2.2. La phase de sélection ……..27

IV.2.3. La phase de déploiement ……27

IV.2.4. La phase d'exécution ……..27

IV.3. Type de composition des services web ……..28

IV.3.1. Degré de participation de l'utilisateur dans la définition du schéma de composition ……28

IV.3.2. La sélection des services web la gestion du flot ……29

IV.4. Langage de composition des services web ……29

IV.4.1. BPEL4WS (Business Process Execution Language for Web Services).....29

IV.4.2. WS-CDL (Web Services Choreography Description Language)..30

V. La qualité de service web (QOS) ……..30

V.1. Introduction ……30

V.2. Les mesure de qualité de service ……31

V.2.1. Disponibilité ……31

V.2.2. Accessibilité ……..31

V.2.3. Intégrité ……31

V.2.4. Fiabilité ……31

V.2.5. Sécurité ……..32

V.2.6. Transaction ……32

V.2.7. cout d'exécution…….32

V.2.8. temps de réponse……32

V.2.9 avis utilisateur……. ……32

VI. Conclusion ……32

Chapitre 03 : Etude conceptuelle

I. Introduction ……..33

II. Conception de l'ontologie……33

II.1. Le partage sémantique……..33

II.2. Permettre la réutilisation du savoir d'un domaine……..33

II.3. Permettre l'explicitation des spécifications du domaine……33

II.4. Analyser le savoir sur un domaine……..33

III. Approche proposé……..33

IV. Architecture conceptuelle de l'ontologie……..34

V. Modélisation sémantique ……..35

V.1. Algorithme de composition...…….36

V.2. La similarité sémantique……38

VI. Construction de l'ontologie pour la découverte et la composition des services web ……38

VI.1. La démarche de la construction……38

VI.2. Langage utilisé……..39

VII. Le modèle de la découverte des services web ……40

VII.1. La création et la définition des concepts……41

VII.1. La définition des relations……..……41

VIII. Le modèle de la composition des services web…….41

VIII.1. Création des classes……..43

VIII.1.Relations obtenus……..44

IX. Conclusion……...44

Chapitre 04 : Implémentation

I. Introduction…….45

II. Les outils et les langages utilisé en développement……..45

II.1. Langage python……..45

II.2. Bibliothèque utilisé……46

II.3. L'environnement de développement anaconda……..46

II.4. Spyder…….47

III. La mise en œuvre…….48

IV. Etude de cas d'utilisation…….50

IV.1. Description des services web utilisé…….50

IV.2. Scénario de la découverte……..54

IV.3. Scénario de la composition……...55

V. Résultat et discutions.....…...57

VI. Conclusion……...59

Conclusion générale……...60

Bibliographie……...61

Liste des figures

La table des figures

Figure 01 : Les couches du web sémantique……04

Figure 02 : Type de ressource et liens du web…..…...……06

Figure 03 : Architecture du web sémantique ……...……08

Figure 04 : Exemple de fichier RDF..……..16

Figure 05 : Processus de découverte...…….20

Figure 06 : schéma du fonctionnement des services web ….……..22

Figure 07 : la pile des couches des services web…….22

Figure 08 : Entête du protocole SOAP…….24

Figure 09 : Le contenu de l'annuaire UDDI…….25

Figure 10 : Cycle de vie de composition des services web……..28

Figure 11 : Architecture globale de l'application……35

Figure 12 : Algorithme de composition……...37

Figure 13 : Similarité entre les paramètres d'entré et de sortie……38

Figure 14 : Schéma du modèle de la découverte……..40

Figure 15 : Schéma du modèle de découverte avec protégé……40

Figure 16 : Schéma générale du modèle de composition……42

Figure 17 : Schéma de l'ontologie de composition avec protégé……43

Figure 18 : Organigramme du modèle de découverte……..49

Figure 19 : Schéma explicatif des étapes de la composition……50

Figure 20 : Les services web sélectionnés……54

Figure 21 : Schéma de composition…….56

Liste des tableaux

Tableau 01 : couches technologique des services web…….21

Tableau 02 : les classes de l'ontologie de découverte……..41

Tableau 03 : les relations de l'ontologie de découverte……...41

Tableau 04 : les classes de l'ontologie de composition……43

Tableau 05 : les relations de l'ontologie de composition…….44

Tableau 06 : Liste des services web utilisés…….51

Tableau 07 : Résultat de similarité entre la requête et les services web……..58

Tableau 08 : Le nombre de service web sélectionné pour chaque mot clé……..58

Tableau 09: Choix d'un service web selon QOS…….58

Tableau 10 : Les services web sélectionné selon QOS pour chaque mot clé……...59

Introduction générale

Introduction générale :

Depuis l'apparition de l'informatique, beaucoup de taches administratives dans divers domaines ont été simplifié, ce qui a amélioré la prestation des services aux bénéficiaires.

Le web est un Ensemble de liaisons hypertextes appliquées à des données multimédias (textes, sons, images, graphique) .Grâce au format HTML, il permet de relier différents documents ou parties de documents entre eux et de décrire comment les pages seront affichées, ces pages web sont traitées que par des êtres humains car le format HTML ne permet pas de décrire ni le contenu, ni la sémantique des documents.

Le web sémantique fait clarifier la recherche sur internet, on ajoutant aux informations existantes une couche de métadonnées RDF pour que les ordinateurs puissent l'exploiter et comprendre les documents et les données sémantiques. [1]

Un service web est un programme informatique ou un logiciel d'application, identifié par l'URI (Uniform Resource Identifier), il permet la communication et l'échange de données entre application et système hétérogène dans des environnements distribués, ce qui a rendu le web plus dynamique. Les services web peuvent être utilisés pour implémenter l'interaction directe entre les applications en messagerie du XML sur des protocoles Internet. [14]

Dernièrement, le domaine des services Web et le domaine du Web sémantique se croisent de plus en plus, afin d'enrichir les services Web de descriptions sémantique pour automatiser la découverte, la sélection, l'invocation, et la composition des services Web.

Ontologie ou l'étude de l'existence est un terme emprunté du domaine de la philosophie ou il veut dire l'étude sur l'existence de l'être en tant qu'être. La notion d'ontologie est devenue un élément clé dans toute une gamme d'applications faisant appel à des connaissances. Les ontologies sont des représentations structurées de connaissances d'un domaine sous forme de réseau conceptuel. Elles sont aujourd'hui considérées comme un support indispensable à la communication entre des agents. Elles sont utilisées dans divers domaines comme la représentation des connaissances, l'extraction d'information, l'intelligence artificielle ...etc.

Pour faire la construction d'une ontologie d'un n'importe quel domaine donné on doit déterminer l'ensemble des termes qui le définissent, ainsi que les relations entre eux. Ensuite, on doit les présenter par des langages pour permettre une compréhension commune et partageable du domaine.

L'objectif de notre travail est de proposer et construire une ontologie pour la découverte et la composition des services web, pour permettre de sélectionner des services pertinent à la requête de l'utilisateur et faire une bonne composition selon les concepts de l'ontologie afin d'obtenir un meilleur service composite.

Plan du mémoire :

Nous allons diviser notre mémoire en quatre chapitres :

Chapitre 01 : Le web sémantique et les ontologies

Dans ce chapitre on va définir le web sémantique, ses principes son architecture et ses domaines d'application, ainsi que nous allons voir qu'est ce qu'une ontologie, ses composants ses différents types et enfin les langages développé et les outils pour la spécification des ontologies.

Chapitre 02 : Les services web

Dans ce deuxième chapitre, nous allons voir les concepts des services web, les méthodes de découverte et de composition des services web, et les différents mesures de QOS (Qualité Of Service).

Chapitre 03 : Etude conceptuelle

Au chapitre 03, nous allons présenter notre approche proposée et l'architecture conceptuelle de l'ontologie, ainsi que nous allons voir la construction de notre ontologie avec les deux modèles découverte et composition.

Chapitre 04 : Implémentation

Dans ce chapitre on va définir les outils et les langages utilisés avec l'environnement de développement, la mise en œuvre de notre ontologie et on va illustrer notre approche avec une étude de cas.

I. Introduction :

Le web est essentiellement syntaxique. Les machines ne peuvent que parcourir les pages du web pour un traitement ou un lien vers une autre page. Cela veut dire que presque tout le contenu du Web est destiné à être lu par un utilisateur humain. Il ne peut donc pas être manipulé de façon intelligente par des programmes informatiques.

Aujourd'hui, avec le volume de la quantité d'information sur le web le traitement ne suffit pas pour donner des résultats adéquats, c'est pour cela il a été proposé d'étendre le web actuel vers le web sémantique qui permet aux machine de comprendre la sémantique, la signification de l'information sur le web, l'un des principaux objectifs du web sémantique est le partage et la réutilisation d'information, dans ce cas on a besoin d'ontologie pour la représentation sémantique des documents.

II. Le web sémantique :

II.1. Qu'est ce qu'un web sémantique :

Le web classique est un ensemble de documents multimédia relié entre eux par des liens hypertexte, et conçu pour être traité par des utilisateurs humains. Mais avec le volume de la quantité d'informations existante sur le web, le traitement syntaxique ne peut pas fournir des résultats adéquat, le web classique ne dispose pas de mécanisme pour décrire les ressources, c'est pour remédier ce défaut qu'il a été proposé d'étendre le web classique vers le web sémantique ou l'information aura un sens afin de faciliter les tâches des utilisateurs, et de permettre aux ordinateurs de regrouper l'information de manière utile. [1]

Donc, un des objectifs du web sémantique est de clarifier la recherche sur internet, on ajoutant aux informations existantes une couche de métadonnées RDF pour que les ordinateurs puissent l'exploiter et comprendre les documents et les données sémantiques.

L'architecture du web sémantique s'appuie sur une pyramide de langage pour représenter des connaissances sur le web dont chaque couche a une fonction principale [1]:

- ✓ Couche XML est utilisé comme couche de base syntaxique. Le langage XML est actuellement considéré comme un standard pour le transport de données sur le web.
- ✓ Couche RDF représente les métadonnées pour les ressources web
- ✓ Couche « ontologie » spécifie la sémantique de métadonnée
- ✓ Couche « logique » s'appuie sur des règles permettent le raisonnement intelligent

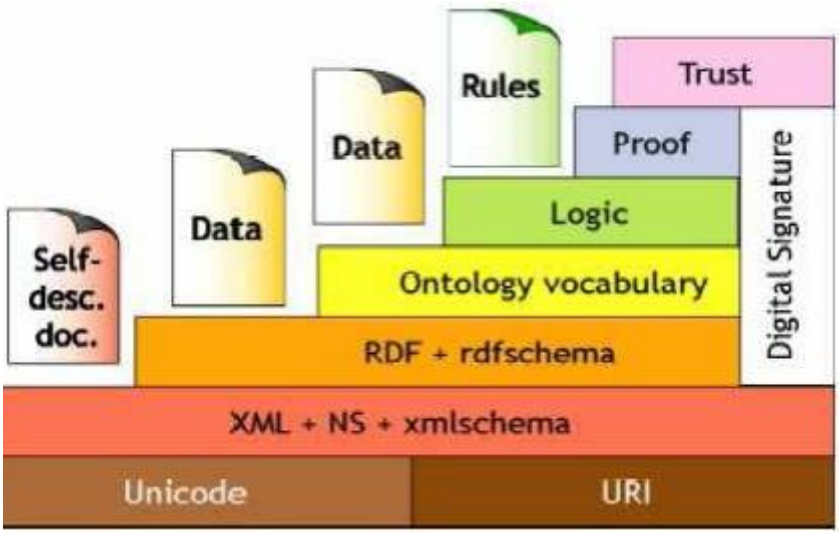


Figure 01. Les couches du web sémantique [1]

II.2. L'objectif du web sémantique :

Un des principaux objectifs du Web sémantique est de permettre aux utilisateurs d'utiliser la totalité du potentiel du Web, ainsi, ils pourront partager et combiner des informations plus facilement. Aujourd'hui tout le monde est capable d'utiliser des réseaux sociaux, de chatter, de faire des recherches ou même d'acheter différents produits. Néanmoins, il serait mieux que la machine fasse tout ceci à la place de l'homme, car actuellement, les machines ont besoin de l'homme pour effectuer ces tâches. Le Web sémantique a donc comme principal objectif que ces mêmes machines puissent réaliser seules toutes les tâches fastidieuses comme la recherche ou l'association d'informations et d'agir sur le Web lui-même. [2]

II.3. Les principes du web sémantique :

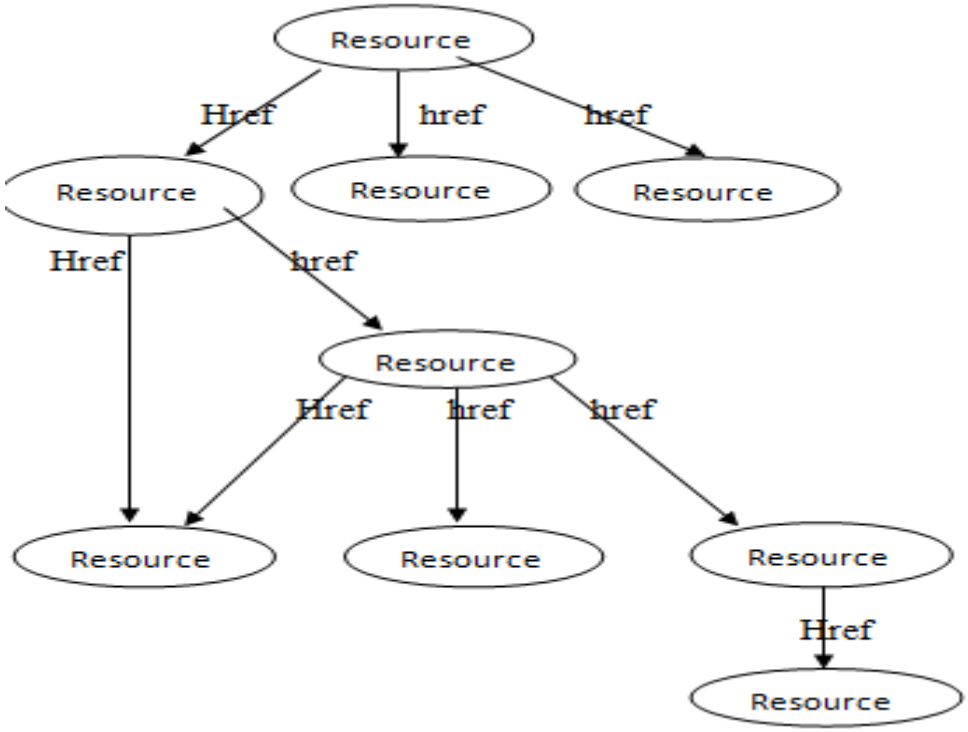
Le web sémantique a pour but d'ajouter une information caractéristique au web pour aider à l'automatisation des services, de découvrir, relier et inférer des connaissances qui sont vraisemblable.[1]

- ✓ **Principe1 : identification par URI :**

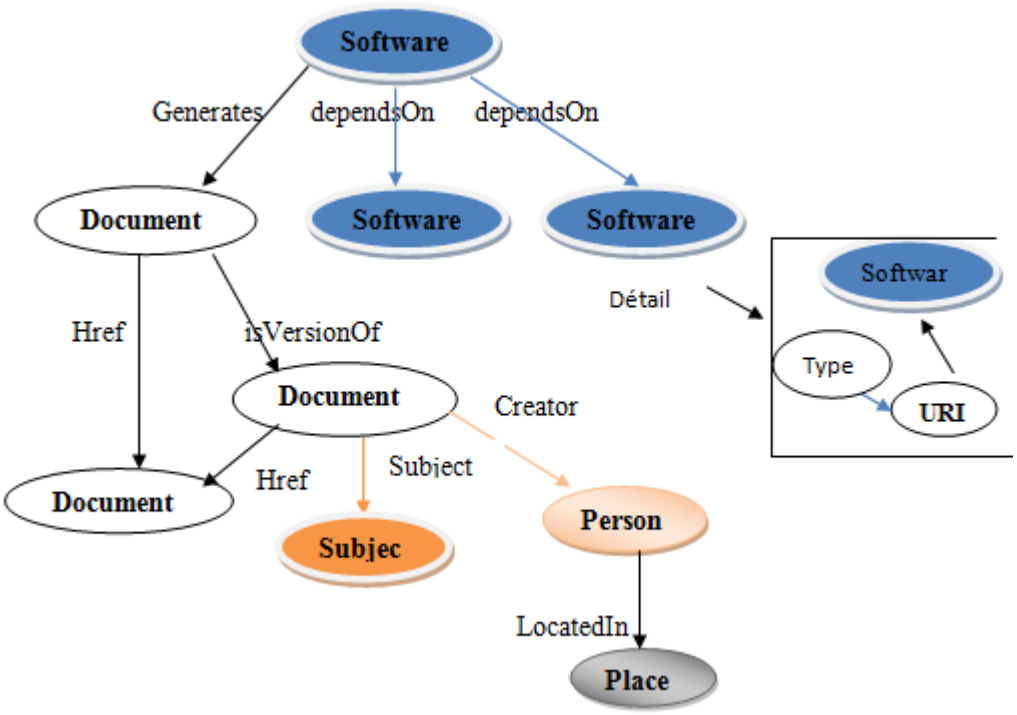
Un URI (Uniform Resource Identifier*)*, soit littéralement identifiant uniforme de ressource, est une courte chaîne de caractères identifiant une ressource sur un réseau.

- ✓ **Principe2 : typage des ressources et liens web :**

Le Web actuel est composé de ressources et de liens. Les ressources sont des documents web conçus pour la compréhension humaine et ne possèdent pas de descriptions expliquant en quoi elles sont utilisées et quelles relations entretiennent-elles avec d'autres documents. Quand une personne peut à directement connaître si un tel document est une facture, un roman ou encore un article de recherche, une machine ignore fréquemment ce type d'information. De la même manière, une personne peut connaître le lien entre deux ressources en lisant le libellé du lien hypertexte qu'il y a entre elles. Alors que pour une machine il est impossible de procéder de la même manière. L'un des objectifs du Web Sémantique est de typer les ressources et les liens du web.



A. web classique



B. Web sémantique

Figure 02 : type de ressources et liens du web [1]

- ✓ **Principe3 : tolérance de l'information partielle :**

Lorsqu'une ressource indexée n'est plus disponible, le fonctionnement doit continuer. Le web sémantique doit continuer de tolérer une information incomplète.

- ✓ **Principe4 : l'indisponibilité d'une vérité absolue :**

Un des problèmes du web est que n'importe qui peut donner des informations sur n'importe quel sujet, alors qu'on ne peut pas tous croire. Les signatures numériques nous permettent de savoir qui a donné cette information.

- ✓ **Principe5 : Architecture minimale :**

Le web sémantique ne doit pas être compliqué et cela dans le but d'une adaptation simple et rapide.

II.4. Architecture du web sémantique :

Le web sémantique nécessite une architecture partagé par tous pour échanger des ressources sur internet, cette dernière est composée de trois niveaux. [3]

II.4.1. Niveau (nommage/adressage) :

La world Wide Web repose sur un concept important qu'est l'URI. Tous ce qui est disponible sur internet doivent être identifié par un URI. Un URI identifié de manière unique et non ambigüe.

II.4.2. Niveau syntaxique :

Le niveau syntaxique est le niveau de la structuration des documents. La spécification de la structure logique repose sur XML, il fournit un ensemble de règles pour la création de vocabulaire qui structurent à la fois les documents et les données sur le web.

II.4.3. Niveau sémantique :

RDF est un standard permettant la mise en place de description simple, il représente la sémantique, on a besoin d'ontologies pour associer les concepts de base d'un domaine précis et les relations entre ces concepts d'une manière compréhensible par la machine.

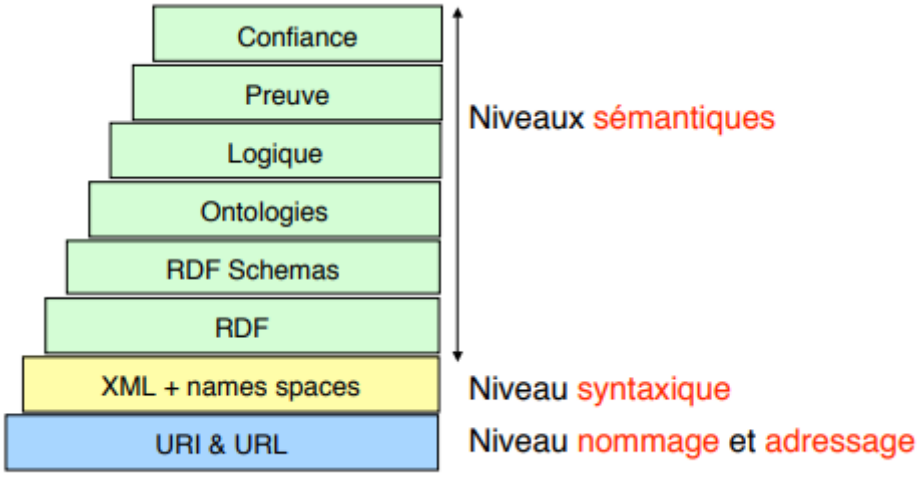


Figure 3 : architecture du web sémantique [4]

II.5. Quelques exemples d'application du web sémantique :

II.5.1. E-commerce :

Le e-commerce va nous permettre d'avoir un échange plus fluide d'information et de transactions entre tous les acteurs économiques, depuis l'offreur de produits ou les services jusqu'aux clients. Les applications du B2B (business to business) ont une plus longue histoire et utilisent les échanges informatisés via des structures de messages et de protocoles très codifiées, préétablies et normalisées. Jusqu'à aujourd'hui, la plupart des transactions B2B sont basées sur EDI (Electronic Data Interchange ou Échange de Données Informatisés) qui récemment assouplies via des standards basés sur XML (eXtensible Markup Language). Cependant, cette approche semble qu'il sera saturé dans un proche avenir. Le commerce électronique basé sur Internet fournit un niveau beaucoup plus élevé de flexibilité et de franchise qui aidera à optimiser des rapports d'affaires. Au lieu de mettre en application un lien à chaque fournisseur, un fournisseur est lié à un grand nombre de clients potentiels. Par conséquent, un fournisseur ou un client peut choisir entre un grand nombre de clients potentiels et peut optimiser ses rapports d'affaires. Donc, on a besoin d'une autre approche : Pair à Pair (P2P pair to pair). C'est à dire : Tout le monde devrait pouvoir échanger et négocier avec tout le monde. Cependant, un commerce électronique si ouvert et flexible doit traiter beaucoup d'obstacles avant que ça devienne réalité. [5]

II.2.2. E-Learning :

On sait que l'objectif du e-Learning est de remplacer les anciennes façons concernant le temps, la place, le contenu de l'apprentissage prédéterminé avec des processus d'apprentissage à temps, à la place de travail, de manière personnalisés et à la demande de l'utilisateur. On trouve que les exigences principales du système de propriété clé de l'architecture du Web sémantique avec le sens partagé commun, métadonnées traitables par les machines, ces exigences peuvent être satisfaites. Par exemple : le matériel d'apprentissage sera annoté de façon plus sémantique par une Ontologie et facilement combiné en un nouveau contenu du cours pour la demande de l'utilisateur. On peut le faire en utilisant les requêtes Web sémantique et la navigation à travers le matériel activée par un fond ontologique. [6]

Avec le Web sémantique, on peut avoir une plateforme adéquate pour implémenter un système e-Learning. On a besoin les moyens pour développer une ontologie d'apprentissage. Le matériel d'apprentissage va être annoté par ontologie, et puis, on doit leur composer dans des cours et faire la livraison active des cours à travers des portails d'apprentissage. [3]

III. Les ontologies :

III.1. Antécédent :

Tout au long de l'évolution des systèmes d'information, les gens se sont ingéniés à concevoir des outils de description du contenu des documents de recherche documentaire. Ces outils ont pris plusieurs formes : classification, thésaurus, taxinomies.

Après l'apparition des systèmes de gestion documentaire, on a des nouveaux outils comme les réseaux sémantiques.

Au début des années soixante, il y avait l'apparition de langage documentaire dont le but était le développement de la syntaxe avec une grammaire.

Le besoin d'outils de description des contenus des documents a été développé on arrivant à l'apparition d'une nouvelle famille linguistique : Les ontologies. [7]

III.2. Qu'est ce qu'une ontologie:

Une ontologie en informatique est un ensemble structuré de concepts ou une spécification formelle explicite d'une conceptualisation partagée permettant de donner un sens aux informations. Son objectif est de modéliser un ensemble de connaissances dans un domaine donné, de plus, les ontologies informatiques sont des outils qui permettent précisément de représenter un corpus de connaissances sous forme utilisable par un ordinateur. Elle fournit le vocabulaire spécifique à un domaine de la connaissance et, selon un degré de formalisation variable, fixe le sens des concepts et les relations qui les unissent. [8]

Une ontologie correspond à un vocabulaire contrôlé et organisé et à la formalisation explicite des relations créées entre les différents termes du vocabulaire. Une ontologie formelle peut être considérée comme la différenciation entre les entités du monde réel et les classes utilisées pour décrire des objets (concepts, propriétés, relations). [1]

III.2.1. Ontologie pour le web :

Pour le Web, simplement, une ontologie est un document ou fichier qui définit de façon formelle les relations entre les termes. Ce type d'ontologie possédera une taxonomie et un ensemble de règles d'inférence. La taxonomie définit des classes d'objets et les relations entre eux. Les classes, les sous-classes et les relations entre les entités sont un très puissant outil pour utiliser sur le Web. On peut exprimer ces relations en attribuant des propriétés aux classes et en permettant à des sous-classes d'hériter de leurs classes mères des propriétés. [3]

III.2.2. Ontologie et service web :

Jusqu'à moment, Ontologie est un des utilisations les plus puissantes pour des agents sur Web et pour des services Web. Récemment, les nombreuses entreprises dans le domaine de e-Commerce B2B avaient discuté le rôle des ontologies pour contrôler les interactions de machine à machine. Cependant, on doit supposer que les ontologies ne utilisent que dans les applications pour s'assurer qu'ils conviennent sur des termes utilisées, des types et des contraintes ..Etc. Sur le Web sémantique, on peut créer des ontologies compréhensible par une machine et utilisable par les agents. Donc, on peut trouver les services Web et automatiser leur utilisation.

Aujourd'hui, il y a un problème pour trouver les nombreux services Web disponibles sur le Web. De plus en plus, le nombre de service se développe avec l'augmentation de nos besoins, donc, la capacité des moteurs de recherche courant pour trouver les services les plus appropriés est tendue à la limite. Actuellement, un certain nombre d'efforts qu'on peut appeler une "publicité de service" nous aide à améliorer cette situation, par exemple UDDI, ebXML, eSpeak. Elle a crée un vocabulaire des annonces de service qui aide les moteur de recherche ayant la capacité de trouver ces services. [3]

La première utilisation des Ontologies pour but de rechercher les services sur le web et de créer les annonces de services dans une langue Ontologie. Comment faire ? Les outils de recherche vont utiliser la hiérarchie et des restrictions de propriété pour trouver les plus correspondances grâce aux propriétés de classe, de sous-classe ou d'autres liens sémantique, c'est à dire une autre ontologie concernée. [3]

III.3. Les Composants d'une ontologie

Les principales composantes d'une ontologie sont : [9]

III.3.1. Concept :

C'est la représentation abstraite des éléments du domaine, appelé aussi termes ou classes. Un concept peut être divisé en trois parties : un terme (ou plusieurs), une notion et un ensemble d'objets.

III.3.2. Terme (label) :

Terme d'un concept est l'expression linguistique utilisée couramment pour y faire référence.

III.3.3. Notion :

La notion désigne ce qui est appelé, au sens de la représentation des connaissances, l'intension du concept. Elle contient sa sémantique qui est définie à l'aide de propriétés (relations et attributs), de règles et de contraintes.

III.3.4. Relation :

Elle exprime les associations entre les différents concepts. Les ontologies contiennent des relations binaires. Le premier argument d'une relation binaire est dit « domaine » et le deuxième argument est dit « Co-domaine ». Cela permet de designer la façon dont la relation doit être lue. Ces relations sont caractérisées par un terme une signature qui précise le nombre d'instances de concepts que la relation lie et leurs types. Les différents types de relations qui peuvent exister sont : sous-classes de «

Spécialisation/Généralisation », partie de« Agrégation ou Composition », « associé à», « composé de ».

III.3.5. Axiomes :

Constituent des assertions considérées toujours comme vraies .Leur inclusion dans une ontologie pour les objectifs: interviennent dans la définition des significations des composants d'ontologie, les contraintes sur les valeurs des attributs, les arguments de relations et dans l'inférence de nouvelles informations, et D'après Les axiomes formels sont utilisés pour vérifier la consistance de l'ontologie. Instances (individus): Ce sont des exemples particuliers de concepts. L'ontologie et l'ensemble des instances constituent la base de connaissance du système informatique.

III.4. Les types d'ontologie :

On peut distinguer six types d'ontologies : [8]

III.4.1. Ontologie de représentation de connaissance

Modélise les représentations primitives utilisées pour la formalisation des connaissances sous un paradigme donnée. Par exemple, une ontologie sur le formalisme des Topic Maps comportera les concepts : Topic, Type de Topic, Association, Occurrence, Type Occurrence.

III.4.2. Ontologie de haut niveau

Elle exprime des conceptualisations valables dans différents domaines. Elle décrit des concepts généraux comme l'espace, le temps, la matière, les objets, les événements, les actions, ces concepts ne dépendent pas d'un problème ou d'un domaine. Ce type d'ontologies est fondé sur la théorie de la dépendance. Son sujet est l'étude des catégories des choses qui existent dans le monde.

III.4.3. Ontologie générique

Elle est appelée également noyau ontologique, modélise des connaissances moins abstraites que celles adapté par l'ontologie de haut niveau mais assez générales néanmoins pour être réutilisées à travers différents domaines. Cette ontologie inclut un vocabulaire relatif aux choses, évènements, temps, espace, causalité, comportement, fonction.

III.4.4. Ontologie de domaine

Cette ontologie exprime des conceptualisations spécifiques à un domaine. Elle fournit les concepts et les relations permettant de couvrir les vocabulaires, activités et théories de ces domaines. L'ontologie du domaine caractérise la connaissance du domaine ou la tache est réalisée.

III.4.5. Ontologie de tache

L'ontologie de taches fournit un vocabulaire systématisé des termes employés pour résoudre des problèmes liés aux taches qui peuvent être ou non du même domaine. Elle fournit un ensemble de termes au moyen desquelles nous pouvons décrire généralement comment résoudre un type de problèmes. Elle inclut des noms, des verbes et des adjectifs génériques dans les descriptions de taches.

III.4.6. Ontologie d'application

C'est l'ontologie la plus spécifique, elle contient des concepts dépendants d'un domaine et d'une tache particuliers, elle est spécifique et non réutilisable. Ces concepts correspondent souvent aux rôles joués par les entités du domaine lors de l'exécution d'une certaine activité.

III.5. Les Critères d'ontologies*:*

Les critères permettent de mettre en évidence les aspects important d'une ontologie sont : [10]

III.5.1. La clarté : définition d'un concept doit faire passer le sens voulu du terme, de manière aussi objective que possible. Une définition doit de plus être et documentée en langage naturel.

III.5.2. La cohérence : rien qui ne puisse être inféré de l'ontologie ne doit entrer en contradiction avec les définitions des concepts.

III.5.3. Extensibilité : les extensions qui pourront être ajoutées à l'ontologie doivent être anticipées. Il doit être possible d'ajouter de nouveaux concepts sans avoir à toucher aux fondations de l'ontologie.

III.5.4. Une déformation d'encodage minimale : une déformation d'encodage a lieu lorsque la spécification influe sur la conceptualisation (un concept donné peut être plus simple à définir d'une certaine façon pour un langage d'ontologie

donné, bien que cette définition ne corresponde pas exactement au sens initial). Ces déformations doivent être évitées autant que possible.

III.5.5. Un engagement ontologique minimal : Le but d'une ontologie est de définir un vocabulaire pour décrire un domaine, si possible de manière complète, c'est la théorie la plus faible couvrant un domaine.

III.6. Domaine d'application d'ontologie : [1]

Elles sont utilisées dans divers domaines y compris:

- ✓ la description de l'information dans différentes spécialités (biologie, médecine..).
- ✓ la recherche d'informations.
- ✓ le commerce électronique afin de permettre, par exemple, la communication entre les fournisseurs et les acheteurs.
- ✓ les environnements de formation à distance.

III.7. Les outils de construction d'ontologies :

On distingue deux familles d'outils : les outils de construction d'ontologie dépendants de formalisme de représentation et les outils de construction d'ontologie indépendants de formalisme de représentation. [8]

III.7.1. Les outils dépendants de formalisme de représentation

- ✓ **Ontolingua** : Ontolingua est un serveur d'édition d'ontologies. Il utilise des classes, des relations, des fonctions, des instances et des axiomes pour décrire une ontologie. Une relation peut contenir des propriétés nécessaires (contraintes) ou nécessaires et suffisantes qui définissent la relation. En plus le serveur Ontolingua offre la possibilité d'intégrer les ontologies Ontolingua, ce qui permet la construction modulaire des ontologies. Il fournit un environnement collaboratif distribué pour parcourir, créer, éditer, modifier et utiliser des ontologies. Le serveur prend en charge plus de 150 utilisateurs actifs, dont certains nous ont fourni des descriptions de leurs projets. [8]
- ✓ **OntoSaurus :** OntoSaurus est basé sur le Web pour permettre un accès rapide à une ontologie à partir de n'importe quelle machine compatible Netscape

connectée au Web. De plus, l'architecture basée sur serveur permet de stocker une ontologie dans un emplacement central, ce qui facilite les problèmes de maintenance de cohérence. Bien qu'une implémentation basée sur le Web présente des avantages évidents, elle introduit également certains problèmes, tels que la latence du réseau, qui doivent être pris en compte dans la conception d'un navigateur. [11]

- ✓ **WebOnto :** WebOnto du Knowledge Media Institute de l'Open University, est une application Web pour naviguer et développer collaborativement les ontologies. Il supporte la navigation collaborative, la création et l'édition d'ontologies sur le Web. Les ontologies WebOnto sont implémentées dans le langage OCML. WebOnto distingue quatre types d'ontologies : ontologie de domaine, ontologie de tâche, ontologie de méthode, et ontologie d'application. [8]

III.7.2. Les outils indépendants de formalisme de représentation

- ✓ **Protégé-2000 :** est une interface modulaire permettant l'édition, la visualisation, le contrôle d'ontologie, l'extraction d'ontologies à partir de sources textuelles, et la fusion semi-automatique d'ontologies. Le modèle de connaissances sous-jacent à protégé 2000 est issu du modèle des frames et contient des classes, des slots (propriétés) et des facets (valeurs des propriétés et contraintes), ainsi que des instances des classes et des propriétés. Il autorise la définition de méta-classes, dont les instances sont des classes, ce qui permet de créer son propre modèle de connaissances avant de bâtir une ontologie. [8]
- ✓ **OntoEdit :** OntoEdit (Ontology Editor) est également un environnement de construction d'ontologies indépendant de tout formalisme. Il permet l'édition des hiérarchies de concepts et de relations et l'expression d'axiomes algébriques portant sur les relations, et de propriétés telles que la généricité d'un concept. Des outils graphiques dédiés à la visualisation d'ontologies sont inclus dans l'environnement. OntoEdit intègre un serveur destiné à l'édition d'une ontologie par plusieurs utilisateurs. Un contrôle de la cohérence de l'ontologie est assuré à travers la gestion des ordres d'édition. [8]

III.8. Langages de spécification d'ontologie

Plusieurs langages sont développé et évoluée pour la spécification des ontologies : [3]

III.8.1. SHOE (Simple HTML Ontology Extension) : SHOE est une langue HTML basée de représentation de la connaissance, elle a été développé en 1996 par Lean Luke, Lee Spector, James Hendler, Jeff Heflin et David Rager à l'Université du Maryland. Le consortium W3C a proposé Extensible Markup Language (XML) qui nous permet de créer les ensembles des étiquettes pour adapter aux besoins du client dans ses documents. Puis pour montrer cette information dans quelque format approprié, on peut utiliser StyleSheet. SHOE est basé sur XML, il permet à SHOE d'être ajoutée aux Web pages et SHOE ajoute à XML une manière standard pour exprimer la sémantique dans un contexte indiqué. [3]

III.8.2. RDF (Resource Description Framework) : RDF est un langage formel qui permet d'affirmer des relations entre des « ressources », il a été adopté la première fois en 1999. Il sera utilisé pour annoter des documents écrits dans des langages non structurés, ou comme une interface pour des documents écrits dans des langages ayant une sémantique équivalente. RDF a été adopté comme recommandation du W3C en 1999. La spécification RDF 1.0 a été publiée en 2004, la spécification RDF 1.1 en 2014. [3]

```
<?xml version = "1.0" ?>
<rdf:RDF
        Xmlns:rdf = http://www.w3.org/1999/02/22-rdf-syntax-ns#>
        <rdf:Description
                About = http://www.ifi.edu.vn/index.html>
                <Creator> Institut de la francophonie pour l'Informatique </Creator>
        </rdf:Description>
</rdf:RDF>
```

Figure 04 : Exemple d'un fichier RDF [3]

III.8.3. OWL (Ontology Web Language) : OWL est un langage basé sur une logique de calcul tel que les connaissances exprimées en OWL peuvent être exploitées

par des programmes informatiques.la version actuelle de OWL a été développé et publié en 2009 avec une deuxième version publié en 2012 qui est une extension de la version publié en 2004.OWL basant sur RDF donne plusieurs possibilités aux ontologies [12]:

- ✓ Capacité d'être distribué à travers beaucoup de systèmes.
- ✓ Compatibilité avec des normes du Web pour l'accessibilité.
- ✓ Ouverture et extensibilité.

IV.8.3. Conclusion :

Dans ce chapitre on a vu qu'est ce qu'un web sémantique, son objectif, ses domaines d'application et ses principe, comme nous avons présenté les composants d'ontologies ses différents types et les langages utilisé pour la représenter et l'implémente.

I. introduction

Avec l'évolution de l'informatique, le besoin d'utilisation du web s'est augmenté et la nécessité des outils pour faciliter la communication entre les machines et les applications connectées via le réseau, les chercheurs ont développé des logiciels pour assurer et simplifier cette communication. Parmi les logiciels développé il y avait l'apparition d'une nouvelle technologie permet à des applications de dialoguer à distance et qui est les services web, ce logiciel permet de communiquer avec les autres applications et services web.

Le défit supposé, est que si un client a besoin des fonctionnalités, et qu'un service ne peut pas les fournir seul alors il voudrait être possible de composer des services existants pour répondre aux besoins de ce client.

En effet, la composition des services web vise à faire coordonner plusieurs services pour la réalisation d'un but.

II. Les services web

II.1. Historique :

Une fois que la mise en réseau s'est généralisée à travers les universités et les industries, il est devenu nécessaire de partager les données et les ressources. Dans les premières années de l'informatique distribuée, au début des années 80 la transmission de messages était la méthode de communication dominante.

Il s'agissait d'encoder les données dans un format de message (c'est-à-dire la manière dont une information structurée est encodée avant sa transmission) et d'envoyer les données encodées par le fil. L'interface de socket permettait le passage de messages en utilisant des primitives d'envoi et de réception sur les protocoles de transport TCP (Transmission Control Protocol) ou UDP (User Datagram Protocol) pour la messagerie de bas niveau sur le protocole Internet (IP).

Avec le développement des applications de calcul distribué, de nouveaux mécanismes et de nouvelles approches sont devenus nécessaires pour faciliter la construction d'applications distribuées. La première technologie de calcul distribué à être largement utilisée a été l'appel de procédure à distance (RPC). La

technologie RPC a été rendue populaire dans les années 1980 par Sun Microsystems. La RPC utilise le modèle client/serveur et étend les capacités des appels de procédure traditionnels à travers un réseau. [13]

Avec de telles applications, plusieurs machines peuvent avoir besoin de fonctionner simultanément sur le même ensemble de données. Par conséquent, l'état de ces données est devenu très préoccupant. La recherche dans le domaine des objets distribués a permis de surmonter ce problème grâce à la spécification de deux technologies concurrentes : Common Object Request Broker Architecture (CORBA) et Distributed Common Object Model (DCOM). Plus tard, le Java Remote Method Invocation (RMI) a été développé et est également devenu un concurrent. [13]

CORBA, DCOM et Java RMI ont connu un succès considérable, mais ils présentent un ensemble de défauts et de limites lorsqu'ils sont utilisés dans des environnements Web. Par exemple, ils tendent à créer des systèmes distribués étroitement couplés, certains sont spécifiques à un fournisseur et à une plate-forme (par exemple, COM/DCOM ne fonctionne que sous Windows), les systèmes distribués développés fonctionnent dans un environnement étroitement administré, certains utilisent des protocoles complexes et propriétaires, ainsi que des formats de message et de représentation de données spécifiques. Avec la croissance du Web, La recherche d'un remplacement de cette technologie conforme au Web a rapidement commencé. [13]

II.2. Définition :

Un service web est un système logiciel identifié par une URI, dont les interfaces publiques et les liaisons sont définies et décrites en utilisant le langage XML. Sa définition peut être découverte par d'autres systèmes logiciels. Les services Web peuvent interagir entre eux d'une manière prescrite par leurs définitions, en utilisant des messages XML portés par les protocoles Internet.[14]

II.3. Architecture orienté service et service web :

La plupart des technologies de calcul distribué ont le concept de services et sont définies par des interfaces. Bien qu'il existe de nombreuses possibilités différentes pour développer une SOA (Services Oriented Architecture) (par exemple, les

services Web, Java RMI, DCOM et CORBA), les services Web sont actuellement la solution la plus souhaitable car ils éliminent de nombreux problèmes d'interopérabilité entre les applications et les services. Les services Web fournissent un grand nombre des normes nécessaires qui sont cruciales pour faire fonctionner un système distribué. Il convient de noter que l'utilisation des services Web ne signifie pas nécessairement qu'il existe une SOA. Il est également possible d'avoir une architecture orientée services sans services web. [13]

Il existe trois actions communes associées à un service dans le cadre de la SOA : la découverte, la demande et la réponse.

La découverte est le processus qui consiste à trouver le service qui fournit la fonctionnalité requise. Une demande fournit l'entrée au service. La réponse donne la sortie du service. Il s'ensuit facilement que cette architecture doit avoir trois acteurs principaux : le demandeur, le fournisseur et le registre. [13]

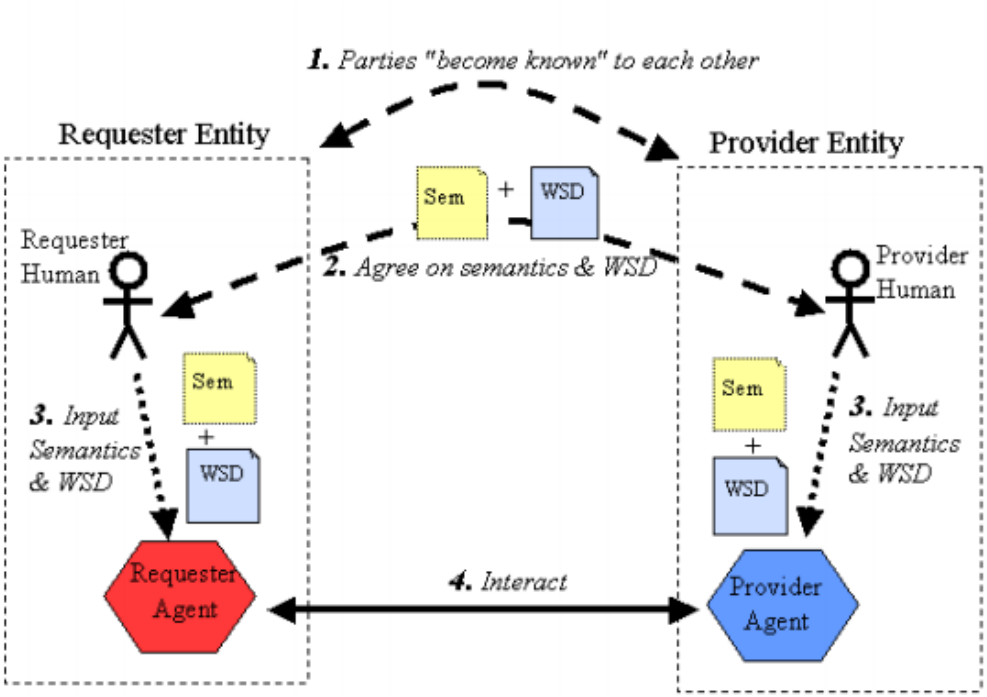


Figure 05. Processus de découverte [13]

Etape 01 : Dans cette étape la figure montre le processus par lequel deux participants prendraient conscience l'un de l'autre. Cela se produit lorsque le fournisseur de services publie la description du service Web (WSD) et la sémantique (Sem.) dans un registre, après quoi le demandeur de services découvre ce service.

Etape 02 : la sémantique et la description sont convenues afin d'éviter tout malentendu sur les données qui sont échangées au cours de cette communication.

Etape 03 : Une fois que WSD et la sémantique sont acceptées par les deux participants et chargées dans les deux, ils peuvent interagir pour effectuer l'opération qui était nécessaire.

II.4. Fonctionnement des services web :

Le fonctionnement des services web repose sur trois couches fondamentales présentées comme suit: [15]

II.4.1. Invocation : visant à établir la communication entre le client et le fournisseur en décrivant la structure des messages échangés.

II.4.2. Découverte : permettant de localiser un service web particulier dans un annuaire de services décrivant les fournisseurs ainsi les services fournis.

II.4.3. Description : l'objectif est de décrire les interfaces des services web.

Pour assurer ces fonctionnalités, trois technologies ont été proposées et disposées en couches pour construire l'architecture de base des Services Web, comme le montre dans le tableau suivant :

UDDI	Découverte de service
WSDL	Description de service
SOAP	Communication
XML	

Tableau. 01. couches technologique des services web [15]

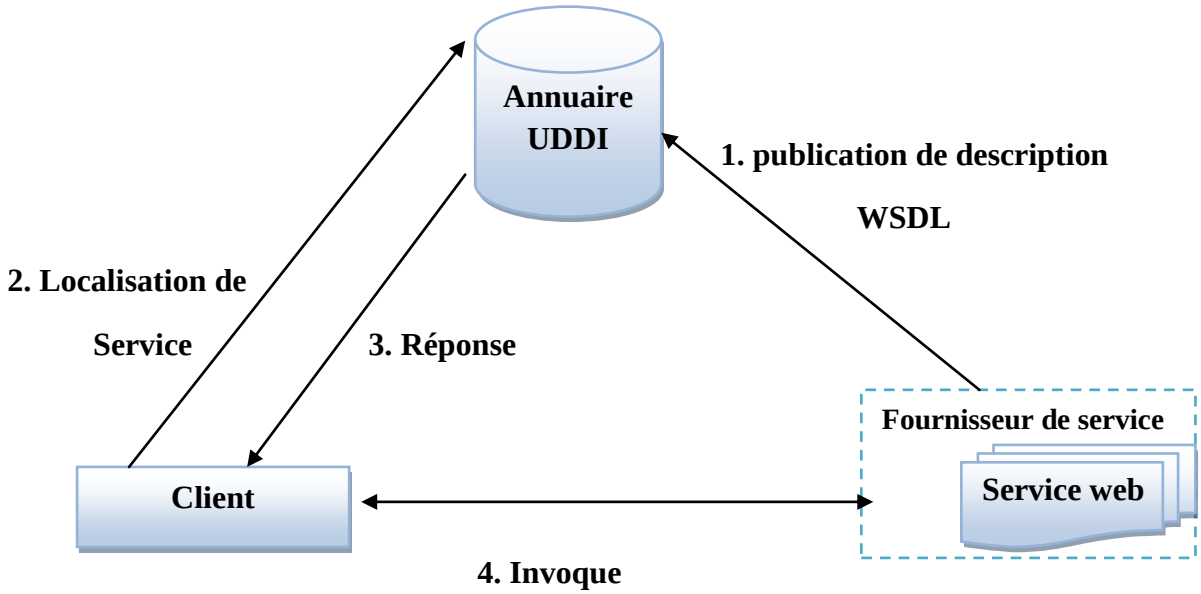


Figure 6 : schéma du fonctionnement des services web

II.5. Description en couche des services web :

Les services Web emploient un ensemble de technologies qui ont été conçues afin de respecter une structure en couches sans être dépendante de façon excessive de la pile des protocoles. Nous distinguons des couches horizontales et des couches verticales. [16]

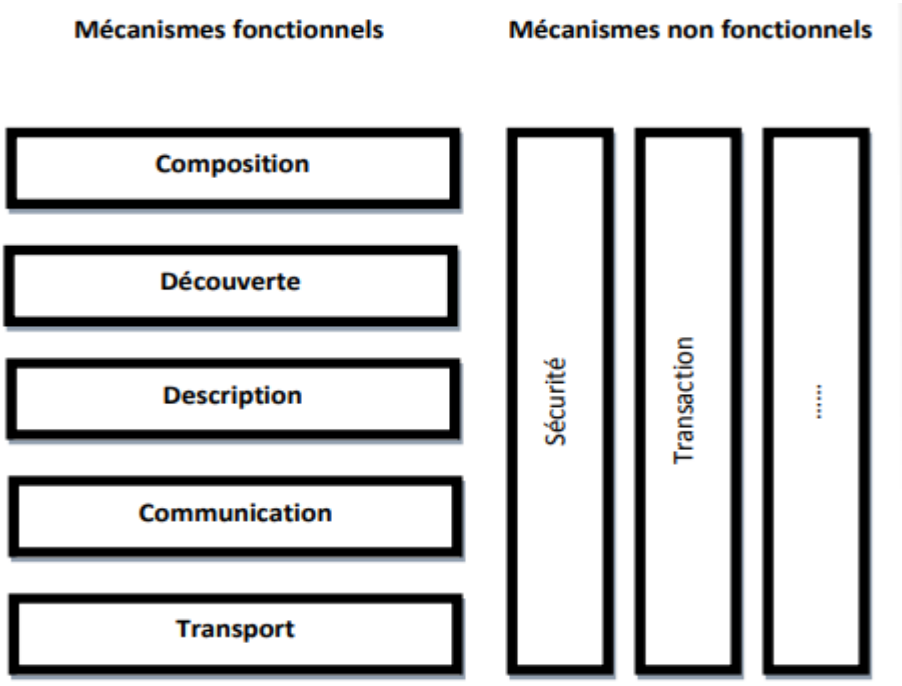


Figure 07. La pile des couches des services web [16]

II.5.1. Couche de transport :

Cette couche s'intéresse aux protocoles de transport de bas niveau, ces derniers vont transporter les requêtes et les réponses échangées entre services. Le protocole le plus utilisé (et recommandé par le consortium « Web Service Interoperability») est le http, mais d'autres implémentations peuvent utiliser un autre ensemble de protocoles tels que : FTP, SMTP, JMS (java messagerie services).

II.5.2. Couche de communication (messagerie) :

Cette couche spécifie les protocoles d'échanges de documents XML entre le service web et ses clients, elle caractérise aussi le mode d'échange (s'il est bloquant ou non). Le protocole SOAP est adopté comme un standard pour la messagerie entre les services web.

- **SOAP**

SOAP (Simple Object Access Protocol) est un protocole d'échange de message indépendant des plateformes, c'est un produit de Microsoft et IBM. Sa première version a été acceptée par le W3C (Word Wide Web Consortium) en 2000.

Il est constitué de deux parties : une enveloppe XML, et un entête d'un protocole de transport. La spécification du protocole SOAP ne donne aucune indication sur le mécanisme de transport du message.

SOAP fait une séparation entre le message et le moyen de transport utilisé. Actuellement, SOAP utilise les protocoles HTTP, SMT ou pour assurer le transport. L'enveloppe XML « envelope » contient à son tour deux sous éléments : un entête « «header » et un corps appelé body.

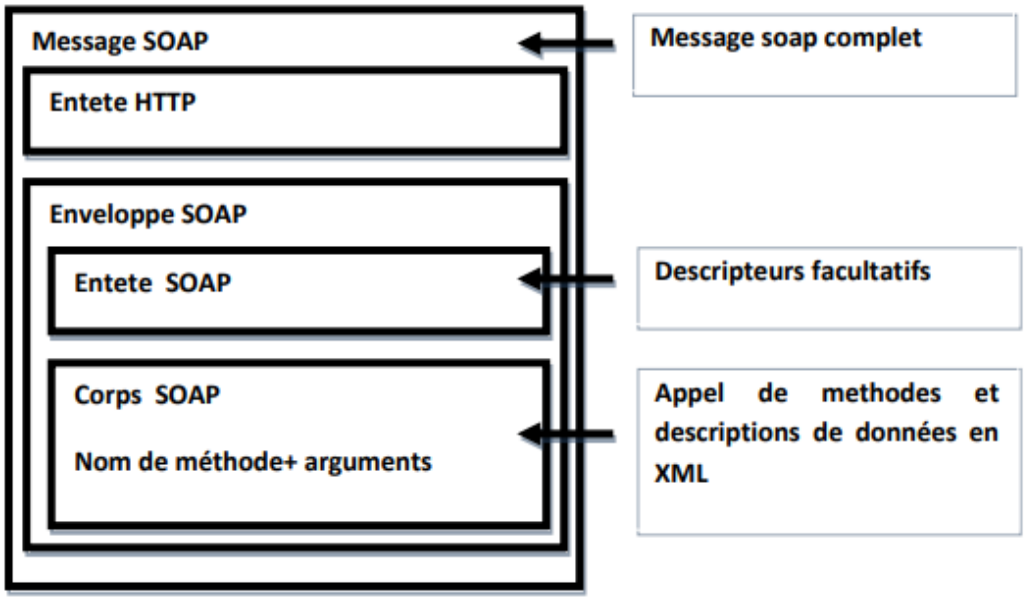


Figure 08. Entête du protocole SOAP [16]

L'entête est facultatif, il peut contenir des informations de sécurité (telles que les signatures électroniques), des informations transactionnelles, des informations de traçabilités.

La partie body est obligatoire elle contient les éléments suivants:

• soit le nom de méthode, avec les données correspondantes, ou un simple document XML, pour le cas d'une requête.

• les valeurs de retour ou un message d'erreur pour le cas d'une réponse.

II.5.3. Couche de description de service :

Beaucoup d'efforts ont été proposés pour décrire les capacités d'un service web, nous distinguons les descriptions syntaxiques et les descriptions sémantiques, chacune d'elle offre des facettes décrivant soit la structure et/ou le comportement du composant.

II.5.4. Couche de découverte :

UDDI (Universal Description, Discovery and Integration) est une spécification d'annuaires de services Web : cette norme W3C propose un ensemble de structures à publier par les fournisseurs de services. Ces structures sont formalisées en XML, elles proposent 03 types d'informations :

- Les pages blanches qui décrivent les informations de contacts sur les entreprises.
- Les pages jaunes qui décrivent des informations de classification de services.
- Les pages verte qui donnent des informations techniques (telles que l'accès) des services

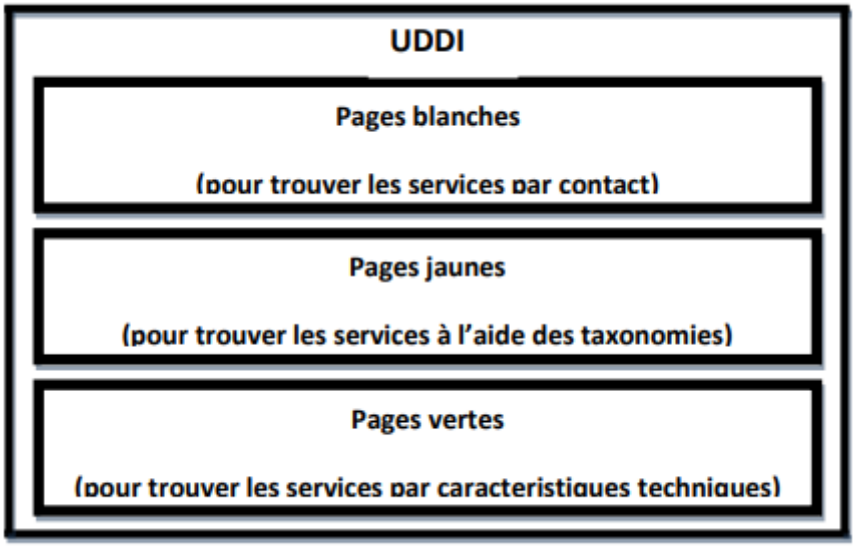


Figure 09. Le contenu de l'annuaire UDDI [16]

II.5.5. Couche de composition :

Plusieurs langages de compositions de services ont été proposés dans la littérature, le langage BPEL est l'un des plus importants formalismes.

III. La découverte des services web

III.1. Introduction

Selon W3C, La découverte des services Web est l'acte de localisation d'une description exploitable par une machine d'un service Web qui peut avoir été précédemment inconnu et qui répond à certains critères fonctionnels. Elle consiste à faire correspondre un ensemble de critères avec un ensemble de descriptions de services web. L'objectif est de trouver un service Web approprié. [16]

III.2. Critères de découverte

On peut distinguer trois critères : [16]

III.2.1. Critère d'Architecture (Centralisation /Distribution)

Ce critère concerne la manière de stockage et de localisation des descriptions de services dans le réseau. Selon ce point de vue, les systèmes peuvent être centralisés ou décentralisés.

III.2.2. Critère d'Automatisation

Ce critère concerne le degré d'intervention de l'utilisateur dans la découverte, nous distinguons les approches manuelles et automatiques.

La découverte manuelle est initiée et assistée par un utilisateur humain. La découverte automatique n'implique pas l'intervention de l'utilisateur, pendant la recherche.

III.2.3. Critère de Matchmaking

Ce point de vue concerne les données à comparer ainsi que l'algorithme d'appariement de la requête avec la description du service. On distingue les approches fonctionnelles et les approches non fonctionnelles:

Les approches non fonctionnelles comparent les valeurs de QOS des services web et les classent selon le besoin de l'utilisateur. Les algorithmes de comparaison s'inspirent des méthodes de décision multicritères des systèmes inférentiels flous.

Les approches fonctionnelles comparent les descriptions d'interfaces ou de comportement des services web, elles peuvent être syntaxiques (WSDL), sémantiques (OWLS, SAWSDL, WSMO) ou comportementales.

IV. La composition des services web

IV.1. Définition

La composition des services Web fait référence au processus qui consiste à combiner les fonctionnalités de plusieurs services, simples ou eux-mêmes composés. Dans le but de répondre à des demandes complexes qu'un seul service ne pourrait satisfaire, et atteindre un objectif particulier. Le résultat de la composition des services peut être un nouveau service, appelé service composite. [17]

IV.2. Cycle de vie d'une composition des services web

La composition des services web est faite par plusieurs activités et étapes, on peut dire qu'elle inclut 4 phases : [17]

IV.2.1. la phase de définition : Durant cette phase le demandeur de service précise la spécification de la composition, qui devrait fournir suffisamment d'informations sur les besoins des utilisateurs et les priorités pour le service composé, Ensuite la spécification est décomposée en un modèle abstrait qui englobe un ensemble des activités, le contrôle et le flux des données entre ces activités, la qualité de service (QoS) et les comportements exceptionnels.

IV.2.2. la phase de sélection : cette phase consiste à découvrir pour chaque activité dans le service composé les services web appropriés qui correspondent aux besoins définies, cette découverte est fondée sur les informations contenues dans les descriptions des services web publiés. Il est possible que plus d'un service réponde aux besoins. Par conséquent, le meilleur service doit être sélectionné. Après que tous les services Web requis sont identifiés et liés aux activités correspondantes, le service composé est produit.

IV.2.3. La phase de déploiement : Dans cette phase, le service composé est déployé pour permettre son instanciation et l'invocation par les utilisateurs finaux. Le résultat de cette phase est le service composé exécutable.

IV.2.4. La phase d'exécution : Dans cette phase, l'instance de service composé est créée et exécutée par un mécanisme d'exécution, qui est également chargé d'invoquer les services participants. Pendant cette étape d'exécution les tâches de contrôle et de surveillance de la composition, mesure de la performance et la gestion des exceptions, doivent être assurées.

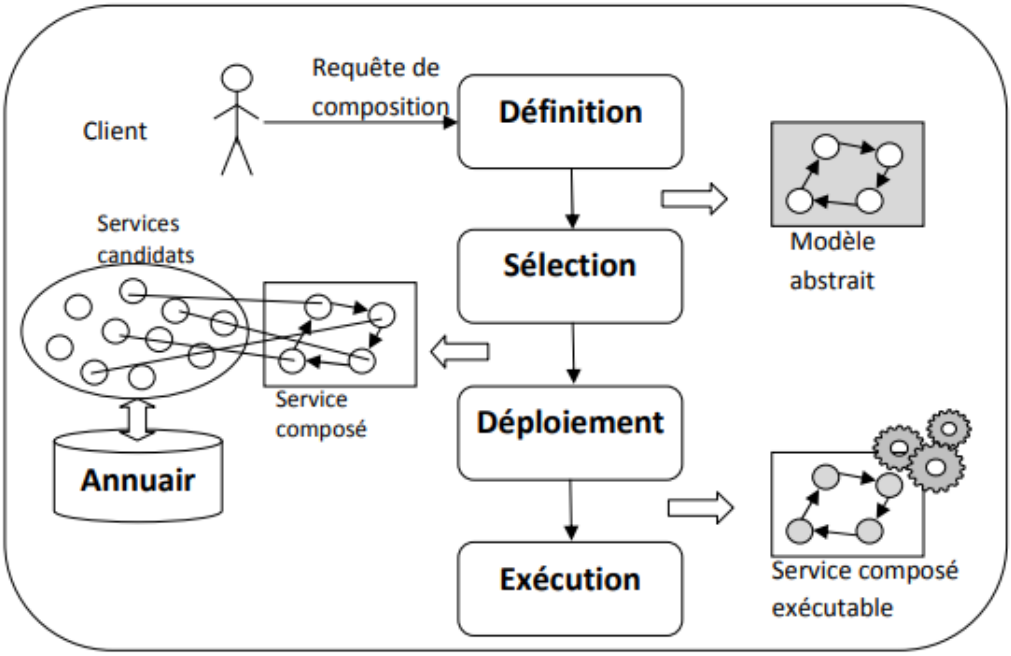


Figure 10. Cycle de vie de composition des services web [17]

IV.3. Types de composition des services web

L'idée de la composition de services Web consiste à définir comment les services Web vont être rassemblés selon certains règles, pour atteindre le but demandé par un utilisateur. Une fois que, la description de la composition est réalisée, il est possible de savoir facilement quels services web appartiendront à cette composition. Les différents types proposés peuvent être classifiée selon deux axes. En fonction de : [18]

IV.3.1. degré de participation de l'utilisateur dans la définition du schéma de composition : [18]

a. **La composition manuelle :** la composition manuelle des services Web suppose que l'utilisateur gère la composition à la main via un éditeur de texte et sans l'aide d'outils dédiés.

b. **La composition semi-automatique :** les techniques de composition semi-automatiques sont un pas en avant en comparaison avec la composition manuelle, dans le sens qu'elles font des suggestions sémantiques pour aider à la sélection des services Web dans le processus de composition.

c. **La composition automatique :** la composition totalement automatisée prend en charge tout le processus de composition et le réalise automatiquement, sans qu'aucune intervention de l'utilisateur ne soit requise.

IV.3.2. la sélection des services Web et la gestion du flot [18]

a. **La composition statique des services web :** dans cette approche, les services web à composer sont choisis à l'heure de faire l'architecture et le design. Les composants sont choisis et reliés ensemble, avant d'être compilés et déployés.

b. **La composition dynamique des services web :** Dans cette approche, les services sont sélectionnés et composés à la volée en fonction des besoins formulés par l'utilisateur. Cette approche offre le potentiel de réaliser des applications flexibles et adaptables, en sélectionnant et en combinant les services de manière appropriée sur la base de la requête et du contexte de l'utilisateur. [19]

IV.4. Langages de composition des services web

De nombreux industriels (tels qu'IBM ou Microsoft) et consortium (tel que le W3C) travaillent afin de mettre en œuvre un langage de composition de services Web standard. Nous distinguons dans cette section deux langages. [18]

IV.4.1. BPEL4WS (Business Process Execution Language for Web Services)

BEA, IBM, SAP, Siebel Systems et Microsoft ont uni leurs efforts afin de produire un langage de composition de services Web, conçu pour supporter les processus métier à travers les services Web. Ce langage, BPEL4WS (Business Process Execution Language for Web Services), est issu de la fusion de deux langages : WSFL – Web Service Flow Language d'IBM et XLANG de Microsoft. BPEL4WS combine les caractéristiques d'un langage de processus structuré par bloc (XLANG) avec ceux d'un langage de processus basé sur les processus métier (WSFL).la première version de BPEL4WS a été apparu en 2004.[18]

Ce langage distingue les processus abstraits des processus exécutables :

Le processus abstrait : ce processus spécifie les messages échangés entre les différentes parties (services Web composants) sans indiquer le comportement de chacune d'elles. Ce processus abstrait peut être relié à une composition de type chorégraphie. Les services Web communiquent alors à l'aide d'échanges de messages. [18]

Le processus exécutable : ce processus permet de spécifier l'ordre d'exécution des activités, le partenaire concerné, les messages échangés entre ces partenaires, et les mécanismes des erreurs et des exceptions. En d'autres termes, il s'agit du moteur de l'orchestration donnant une représentation indépendante des interactions entre les partenaires. [18]

IV.4.2. WS-CDL (Web Services Choreography Description Language)

Le WS-CDL (Web Services Choreography Description Language) apparu en 2005, est un langage qui permet de décrire une vision globale des collaborations entre les services. Il décrit les séquences ordonnées de messages impliquant plusieurs entités visant à accomplir un objectif commun.WS-CDL reprend et développe la spécification Web Service Choreography Interface WSCI. [18]

WS-CDL permet :

1. de désigner les variables et les types de données échangées.

2. de décrire les activités impliquées.

3. de décrire les structures illustrant les interactions entre les activités.

V. La qualité de service (QOS)

V.1. Introduction :

QoS (Quality Of Service) fait référence à la capacité du service Web à répondre aux invocations attendues et à les réaliser au niveau correspondant aux attentes mutuelles de son fournisseur et de ses clients. Plusieurs facteurs de qualité qui reflètent les attentes des clients, tels que : [20]

- ✓ la disponibilité constante du service
- ✓ la connectivité
- ✓ la grande réactivité

Ces facteurs deviennent essentiels pour maintenir la compétitivité et la viabilité d'une entreprise car ils peuvent avoir un impact sérieux sur la fourniture de services.

V.2. Les mesures de QOS (qualité of service) :

Traditionnellement, la qualité de service est mesurée par la mesure dans laquelle les applications, les systèmes, les réseaux et tous les autres éléments de l'infrastructure informatique soutiennent la disponibilité des services à un niveau de performance requis dans toutes les conditions d'accès et de charge.

Les éléments clés pour soutenir la qualité de service dans un environnement de services Web sont résumés dans ce qui suit : [20]

V.2.1. **Disponibilité** : La disponibilité est l'absence de temps d'arrêt du service. La disponibilité représente la probabilité qu'un service soit disponible. Des valeurs plus élevées signifient que le service est toujours prêt à être utilisé, tandis que des valeurs plus faibles indiquent l'imprévisibilité quant à la disponibilité du service à un moment donné. Le temps de réparation est également associé à la disponibilité. Le TTR (time-to-repair) représente le temps nécessaire pour réparer un service qui a échoué. Idéalement, des valeurs plus petites de TTR sont souhaitables.

V.2.2. **Accessibilité** : L'accessibilité représente le degré de satisfaction d'une demande de service Web. Elle peut être exprimée comme une mesure de probabilité indiquant le taux de réussite ou les chances de réussite de l'instanciation d'un service à un moment donné. Un degré élevé d'accessibilité signifie qu'un service est disponible pour un grand nombre de clients et que ces derniers peuvent utiliser le service relativement facilement.

V.2.3. **Intégrité** : Décrit la mesure dans laquelle un service Web accomplit ses tâches selon sa description WSDL ainsi que la conformité avec l'accord de niveau de service SLA (Service Level Agreement). Un degré d'intégrité plus élevé signifie que la fonctionnalité d'un service est plus proche de sa description WSDL ou de son SLA (Service Level Agreement).

V.2.4. **Fiabilité** : La fiabilité représente la capacité d'un service à fonctionner correctement et de manière cohérente et à fournir la même qualité de service malgré les défaillances du système ou du réseau. La fiabilité d'un service Web est généralement exprimée en termes de nombre de défaillances transactionnelles par mois ou par an.

V.2.5. **Sécurité** : La sécurité implique des aspects tels que l'authentification, l'autorisation, l'intégrité des messages et la confidentialité. La sécurité est d'autant plus importante que l'invocation des services Web se fait sur l'internet. Le niveau de sécurité requis par un service web particulier est décrit dans l'accord de niveau de service (SLA) qui l'accompagne, et les fournisseurs de services doivent maintenir ce niveau de sécurité.

V.2.6. **Transactions** : Il existe plusieurs cas où les services Web nécessitent un comportement transactionnel et une propagation du contexte. Le fait qu'un service web particulier nécessite un comportement transactionnel est décrit dans l'accord de niveau de service SLA (Service Level Agreement) qui l'accompagne, et les fournisseurs de services doivent conserver cette propriété.

V.2.7. **Cout d'exécution :** le prix d'exécution est la somme d'argent qu'un demandeur de service doit payer pour exécuter une opération d'un service.les allocataires de service annoncent directement le prix d'exécution de leurs opérations. [21]

V.2.8. **Temps de réponse :** latence (temps de réponse) mesure le retard prévu en secondes entre le moment ou l'exécution commence et moment ou l'exécution finit. [21]

V.2.9. **Avis utilisateur :** la réputation d'un service est une mesure de sa fidélité. Elle dépend principalement des expériences de l'utilisateur d'employer ce service. Les différents utilisateurs peuvent avoir le service à peu pré identique de différent avis. La valeur de la réputation est définie comme rang moyen indiqué au service par des utilisateurs. [21]

VI. Conclusion :

Dans ce chapitre on a vu qu'est ce qu'un service web, comment il est modéliser, ses caractéristiques et les méthodes utilisé pour mesurer la qualité des services web. En outre, on a spécifié les types de composition des services et les langages utilisé

I. Introduction :

La description formelle de la sémantique des noms des services web consiste à rendre son contenu explicite et faciliter la recherche ainsi que la composition par l'utilisateur Elle peut être réalisé en utilisant les ontologies qui offrent une description formelle de la sémantique.

Au niveau de ce chapitre nous allons présenter notre approche, commençant par la conception de l'ontologie, son architecture, ses différents concepts et son rôle dans la composition dynamique des services web.

II. Conception de l'ontologie :

Les facteurs qui peuvent nous mener au développement des ontologies sont divers, on peut citer quatre : [22]

II.1. Le partage sémantique : représente l'une des premières motivations. La sémantique représente l'interprétation selon un point de vue ou un contexte particulier. Le partage sémantique doit donc être guidé par le contexte et le but à atteindre.

II.2. Permettre la réutilisation du savoir d'un domaine : ceci se fait à travers la réutilisation de l'ontologie pour différents domaines ou différentes applications.

II.3. Permettre l'explicitation des spécifications du domaine: Les postulats implicites sur un domaine exprimés en langage codé de programmation sont difficiles à deviner, à comprendre et à modifier. Rendre explicites ces postulats et les spécifications du domaine grâce à une ontologie permet leur modification en cas d'évolution du savoir sur le domaine.

II.4. Analyser le savoir sur un domaine : la spécification des termes du domaine est très importante, c'est ce qui permet d'analyser le savoir sur un domaine. Ce savoir peut par la suite être utilisé dans différentes applications.

III. Approche proposé :

Notre solution consiste à proposer une ontologie pour la découverte et composition des services web à fin de répondre aux besoins des utilisateurs.

La structure d'un service web peut être implicite ou explicite. Elle est dite implicite, quand elle n'est accessible qu'à l'être humain. Elle est dite explicite, quand elle est accessible aux applications informatiques.

L'ontologie offre aux utilisateurs un environnement d'annotation sémantique qui permet une meilleure recherche, elle associe une sémantique à certains éléments des services web en référence à une ontologie.

IV. Architecture conceptuelle de l'ontologie :

Dans un environnement dynamique, quand on reçoit une demande de composition de service web on prend en charge la sémantique de la requête et des descriptions pour le matching des services web à fin de diminuer les fautes et augmenter la qualité du résultat.

A la composition des services web plusieurs conflits sont possibles à cause de la diversité du vocabulaire des services web. Donc, une nouvelle méthode est abordée pour éliminer ces problèmes d'expression qui est l'utilisation des ontologies.

L'architecture proposée est la construction d'une ontologie pour la découverte et la composition des services web, donc on va rechercher des services web dans une ontologie selon une requête écrite par l'utilisateur, après le résultat de la découverte, on va exécuter un algorithme de composition pour produire un schéma de composition.

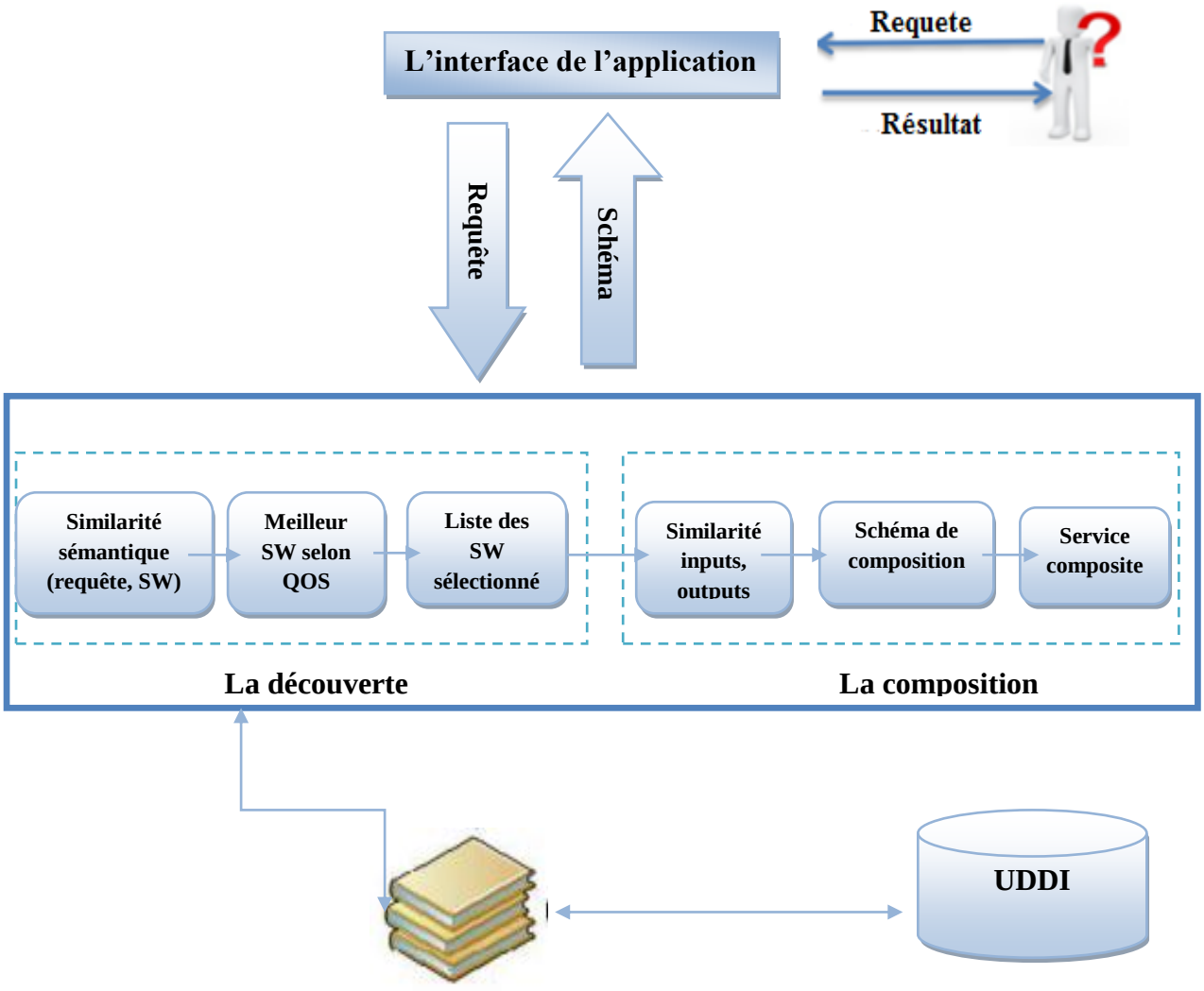


Figure 11 : Architecture globale de notre application

V. Modélisation sémantique

Les méthodes de la découverte et la composition des services web sont très nombreux, mais pour sélectionner des services pertinents à la requête de l'utilisateur et également faire une meilleure composition et avoir un bon composite, Alors, on sera contraint d'adopter le coté sémantique. Pour cela, il est nécessaire de développer un modèle qui favorisera la modularité, la réutilisation et le partage de données dans les systèmes informatiques et permettra de modéliser l'aspect sémantique de la découverte et la composition.

Tout d'abord, nous aurons besoin d'établir un lien sémantique entre les mots clés de la requête et les propriétés des services web.

Ensuite, on va sélectionner les services pertinents selon des contraintes de QOS (avis utilisateur, temps de réponse, cout d'exécution), âpres ça on va produire un schéma de compositions selon l'algorithme proposé suivant :

V.1. Algorithme de composition :

L'algorithme suivant est utilisé pour construite le schéma de composition, il prend en entré un ensemble de service web sélectionné dans la partie de découverte et construit le schéma. L'algorithme recherche les services web, qui ne nécessitent que les données fournies dans la requête de l'utilisateur, ces services seront ajouté dans la liste des services indépendant (L_independant), le reste sera ajouté dans la liste (L_middle). Ensuite pour chaque service web de la liste (L_middle) et chaque service de la liste (L_independant) on calcule les similitudes entres leurs paramètres d'entré et de sortie (figure 12)

Enfin, l'algorithme élimine les compositions impossibles en supprimant les arcs du graphe entre les services Web où le nombre de paramètres d'entrée nécessaires est inférieur au nombre de paramètres de sortie.

Algorithme de construction de schéma de composition

Entrée :

L_SW :SW1,… SWn : liste des services web sélectionné de l'ontologie.

L_input_user : input_user1,…input_userm La liste des entrées saisie par l'utilisateur.

Sortie G : un graphe représentant le schéma de composition.

Debut

L_independant← ∅;

L_middle ← ∅;

Pour (Swi dans L_SW) **faire**

Pour (inputj dans Swi) **faire**

Calcule similarité (L_input_user, inputj)

Si (similarité<0.50) **alors**

L_middle ←SWi;

Break;

Fisi;

Fin;

L_independant←Swi;

G ←(Swi);

Fin

Pour (Swi dans L_middle) **faire**

Pour (SWj dans L_independant) **faire**

Calcule similarité (input(Swi), output (Swj))

Si (similarité >=0.50) **alors**

Déplacer Swi de L_middle à L_independant ;

G ←(swj, swi) ;

Finsi ;

Fin

Fin

Figure 12 : algorithme de composition

V.2. La similarité sémantique :

La composition détermine les paramètres d'entrées et de sortie de chaque service sélectionné. Ensuite, on va calculer des similitudes entre les services web afin de déterminer le schéma de composition en fonction de l'algorithme **(section V.1 figure 12)**.

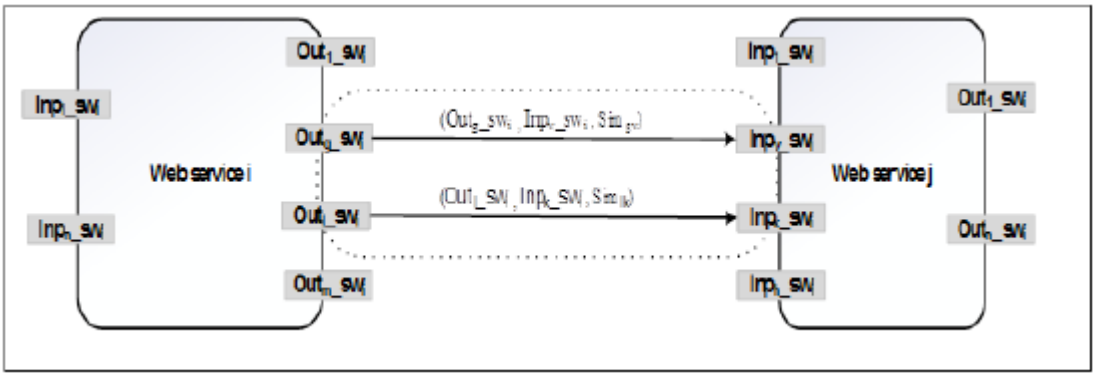


Figure 13: similarité entre les paramètres d'entré et de sortie

VI. Construction de l'ontologie pour la découverte et composition des services web :

VI.1. La démarche de la construction :

Une ontologie peut etre construite selon trois méthodes distinctes : [22]

VI.1.1. Méthode manuelle : les experts réalisent une nouvelle ontologie d'un domaine ou étendent une ontologie existante.

VI.1.2. Méthode automatique : l'ontologie est construite par des techniques d'extraction des connaissances : des concepts et des relations entre eux sont extraits et ensuite vérifiés par des inférences, pour construire une ontologie.

VI.1.3. Méthode mixte : les ontologies sont construites par des techniques automatiques mais elles permettent d'étendre des ontologies ayant été construites manuellement

D'après les recherches que nous avons faites on n'a pas trouvé une ontologie pour la composition des services web, nous procéderons alors à une construction manuelle d'une ontologie de découverte et de composition des services web.

Par apport à la méthode de construction, nous avons suivi les étapes suivantes :

- ✓ Création de l'ontologie
- ✓ Création des concepts
- ✓ Création des relations entre les concepts
- ✓ Création des instances des concepts

VI.2. Langage utilisé :

Depuis plusieurs années, une nouvelle idée du Web prend corps : celle d'un Web Sémantique. Peu à peu, le World Wide Web Consortium (W3C) se dote de nouveaux langages et d'outils plus performants : XML, RDF, OWL, n'en sont que des exemples. Tous ont pour objectif commun de participer à une formalisation des savoirs, en permettant ainsi un meilleur partage et une transmission plus aisée. Les champs d'application éventuels sont vastes : raisonnement automatique, résolution de problèmes par inférences, représentation de données structurées, traduction automatisée, etc. [22]

Par rapport à nous, on a choisi le langage OWL pour les raisons suivantes : [22]

- ✓ Langage OWL est issu de ses prédécesseurs. Parmi les impératifs de conception de ce langage figure la compatibilité avec RDF. Celle-ci se traduit par l'utilisation de RDF/XML comme syntaxe concrète d'OWL. Seulement, cette compatibilité ne doit pas être uniquement syntaxique mais aussi sémantique.
- ✓ Un autre objectif important d'OWL est de pouvoir assurer qu'il est possible de créer un moteur d'inférences décidable basé sur ce langage.
- ✓ Il permet de construire des ontologies qui peuvent être linguistiques, conceptuelles ou intermédiaire entre ces deux catégories. Cette particularité vient du fait qu'OWL permet de définir les concepts primitifs des ontologies.

Donc chaque ontologie OWL possède les caractéristiques suivantes : Multilingue, Modulaire, Interopérable, Déductive.

Sur la base de ces causes, nous allons développer un nouveau modèle OWL qui servira de squelette pour la création d'ontologie de découverte et de composition.

VII. Le modèle de la découverte des services web

Un système basé sur notre modèle doit répondre à la requête de l'utilisateur en lui sélectionnant la liste des services web et aussi doit pouvoir afficher le résultat de la sélection selon les concepts proposé.

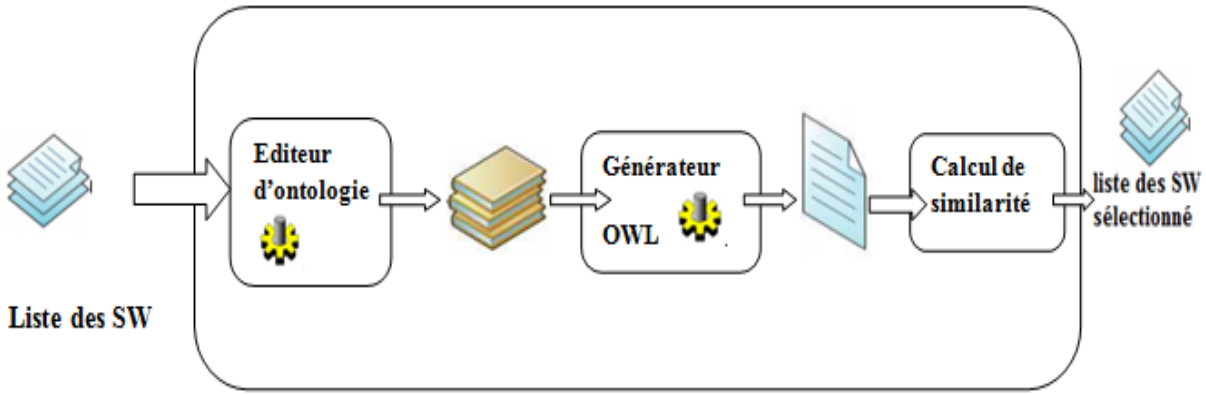


Figure 14 : schéma du modèle de la découverte

La description de notre ontologie et de sa structure est assurée par un ensemble de classes et les relations qui relient ces classes. Nous proposons notre modèle par les schémas suivants :

- **Schéma conceptuel :**

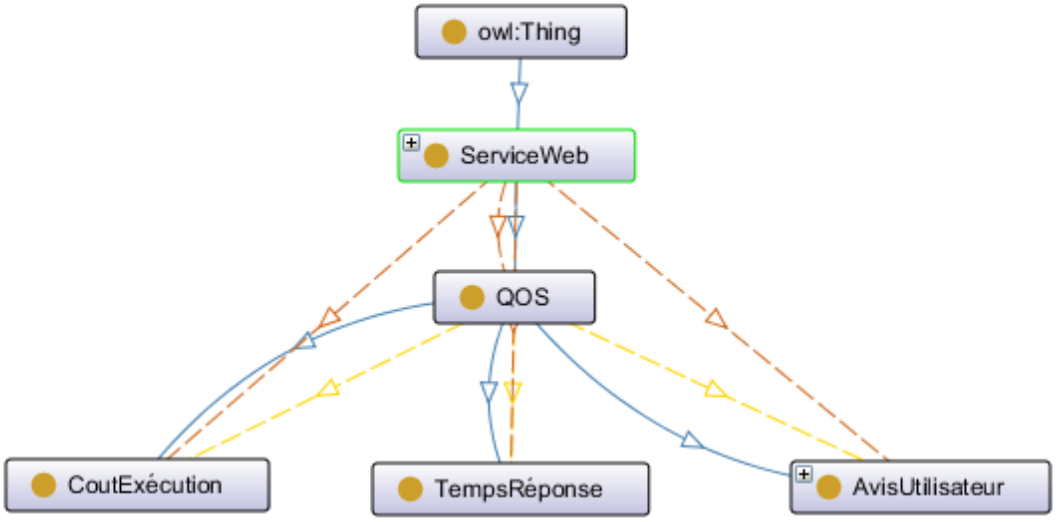


Figure 15 : schéma du modèle de découverte avec protégé

La construction de cette ontologie est faite par les étapes suivantes :

VII.1. La création et la définition des concepts (entités) : on a tout d'abord la classe mère (service web) elle a une sous classe (QOS).

Nom	Type	Propriétés
Service web	Classe	- Nom_S - Id_S
QOS	Classe	- IdQ
Cout exécution	Classe	- Cout
Temps réponse	Classe	- Temp
Avis utilisateur	Classe	- AvisU

Tableau 02 : les classes de l'ontologie de découverte

VII.1. La définition des relations : c'est une relation qui peut lier deux classes ou classe avec sous-classe :

Nom	Type	Domain	Range
MesuréPar	Relation	ServiceWeb	QOS
DécrisPar	Relation	QOS	- CoutExécution - TempRéponse - AvisUtilisateur

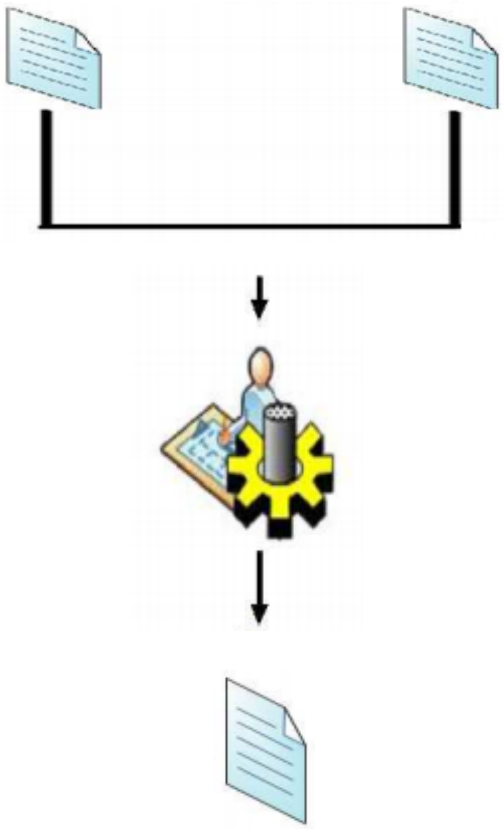
Tableau 03 : les relations de l'ontologie de découverte

VIII. Le modèle de composition des services web

Le modèle de composition combine de manière dynamique les services Web renvoyés par le modèle de découverte pour former un service Web composite qui renvoie les données demandées. En fait, une fois que le modèle de découverte a sélectionné les services pertinents. On peut obtenir le meilleur composite qui a les meilleurs critères.

Le module de composition détermine les paramètres d'entrée et de sortie de chaque service sélectionné. Des similarités entre les paramètres d'entré et de sortie des services Web sont ensuite calculées pour déterminer le schéma de composition en fonction de l'algorithme (section V, figure 09).

Liste des SW sélectionné **les similarités entre les inputs et les outputs**



Le schéma de composition

Figure 16 : schéma générale du modèle de composition

La description de l'ontologie de compostions et de sa structure est assurée par un ensemble de classes et les relations qui relient ces classes. Nous proposons notre modèle par le schéma suivant :

➢ **Schéma conceptuel :**

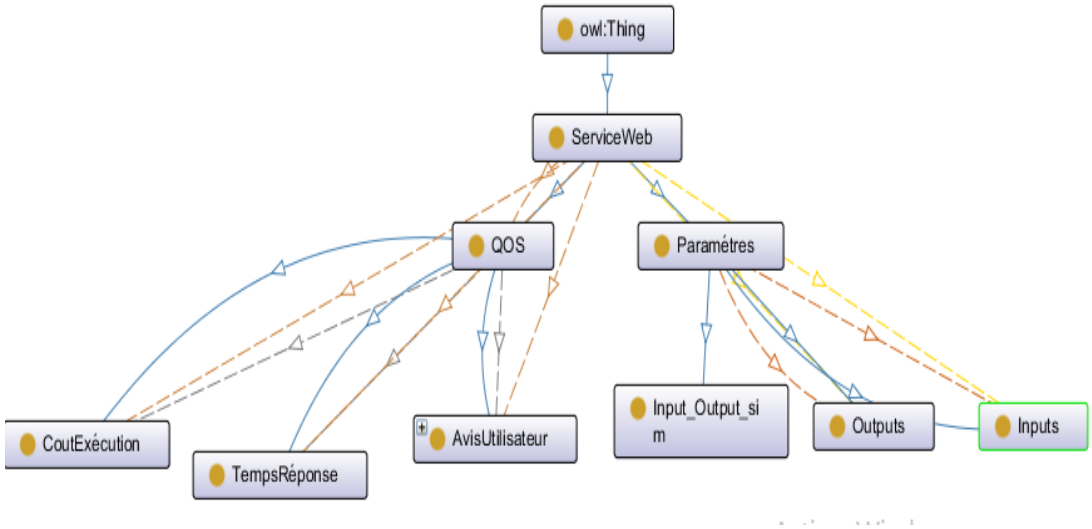


Figure 17 : schéma d'ontologie de composition avec protégé

VIII.1. Création des classes :

Nom	Type	Propriétés
Service web	Classe	- Nom_S - Id_S
QOS	Classe	- IdQ
Cout exécution	Classe	- Cout
Temps réponse	Classe	- Temp
Avis utilisateur	Classe	- AvisU
Paramètres	Classe	-
Input	Classe	-
Output	Classe	-
Input_Output_Sim	Classe	- Service_in - Service_out - Similarité

Tableau 04 : les classes de l'ontologie de composition

VIII.2. Relation obtenue :

Après la représentation des classes et les sous classes, une fois que les relations entre ces classes établies on peut définir le tableau suivant

Nom	Type	Domain	Range
MesuréPar	Relation	ServiceWeb	QOS
HasParamétres	Relation	ServiceWeb	Paramétres
DécrisPar	Relation	QOS	- CoutExécution - TempRéponse - AvisUtilisateur
CaractériséPar	Relation	Paramètres	- Inputs - Outputs - Input_Output_Sim

Tableau 05 : Les relations d'ontologie de composition

IX. Conclusion

Dans ce chapitre nous avons proposé notre approche conceptuelle pour le travail, les facteurs principaux qui ont mené aux développements des ontologies, la construction d'ontologies avec ses différents concepts et les relations entrer eux dans protégé et enfin la composition des services selon cette ontologie

I. Introduction :

La construction et l'utilisation des ontologies nécessite des exigences ontologiques qui sont des obligations résulte du choix d'utiliser certains concepts plutôt que d'autres pour représenter un phénomène. Dans ce chapitre nous allons présenter l'implémentation et la réalisation de notre ontologie sur les services web et également nous devons spécifier les outils que nous avons utilisé et qui nous servi et faciliter l'implémentation.

II. Les outils et les langages utilisés en développement :

Nous avons choisi quelques outils qui nous ont aidés pendant notre implémentation :

II.1. Langage python :

Python est un langage de programmation puissant. Il dispose des structures de données de haut niveau et permet une approche simple mais efficace de la programmation orientée objet. Parce que sa syntaxe est élégante, que son typage est dynamique et qu'il est interprété, Python est un langage idéal pour l'écriture de scripts et le développement rapide d'applications dans de nombreux domaines et sur la plupart des plateformes. [23]



Nous avons utilisé le langage python pour divers avantages :

- Il constitue une source de documentation
- L'interpréteur Python et sa vaste bibliothèque standard sont disponibles librement, sous forme de sources ou de binaires.
- L'interpréteur Python peut être facilement étendu par de nouvelles fonctions et types de données implémentés en C ou C++.
- Il permet de créer plus de fonction moins de lignes de codes.

- Python est un langage facile à apprendre et open source

II.2. Bibliothèques utilisées

a) Bibliothèque « BeautifulSoup » :

Beautiful Soup est un package python et, comme son nom l'indique, analyse les données indésirables et aide à organiser et à formater les données Web désordonnées en corrigeant le mauvais HTML et en nous les présentant dans des structures XML facilement traversables. En bref, Beautiful Soup est un package python qui nous permet d'extraire des données de documents HTML et XML. [24]

Avec cette bibliothèque on peut avoir l'un des besoins suivants :

- Analyser une page web pour déterminer combien de balises sont trouvées, combien d'éléments de chaque balise sont trouvés et leurs valeurs. Vous pourriez vouloir les modifier.
- Déterminer les noms et les valeurs des éléments, afin de les utiliser en conjonction avec d'autres bibliothèques pour l'automatisation des pages web.

b) Bibliothèque « Codecs » :

Le module codecs fournit des interfaces de flux et de fichiers pour le transcodage des données dans votre programme. Il est le plus souvent utilisé pour travailler avec du texte Unicode, mais d'autres codages sont également disponibles à d'autres buts.

Unicode est une spécification qui vise à répertorier chaque caractère utilisé par les langues humaines et à donner à chaque caractère son propre code unique. Les spécifications Unicode sont continuellement révisées et mises à jour pour ajouter de nouveaux langages et symboles. [25]

II.3. L'environnement de développement Anaconda :

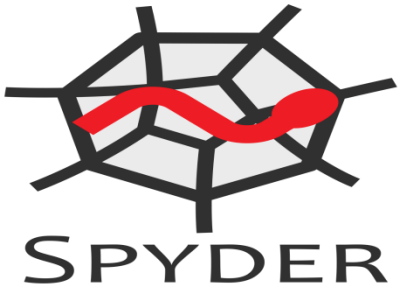
Anaconda est une plateforme scientifique de données prête à l'emploi, sécurisée et évolutive qui permet aux équipes de gérer les ressources scientifiques, de collaborer et de déployer des projets scientifiques. [26]



Elle comprend ces capacités :

- Déployez facilement vos projets dans des applications de données interactives, des ordinateurs portables en direct et des modèles d'apprentissage automatique grâce aux API.
- Partagez ces applications avec vos collègues et collaborateurs.

II.4. **Spyder** :



Spyder est un environnement scientifique puissant écrit en Python, pour Python, et conçu par et pour les scientifiques, les ingénieurs et les analystes de données. Il présente une combinaison unique de fonctionnalités avancées d'édition, d'analyse, de débogage et de profilage d'un outil de développement complet avec les capacités d'exploration de données, d'exécution interactive, d'inspection approfondie et de visualisation magnifique d'un progiciel scientifique. [27]

III. La mise en œuvre:

Dans cette partie nous allons présenter le fonctionnement de notre ontologie pratiquement.

La figure suivante (**figure 18**) représente l'organigramme de l'implémentation de modèle de découverte :

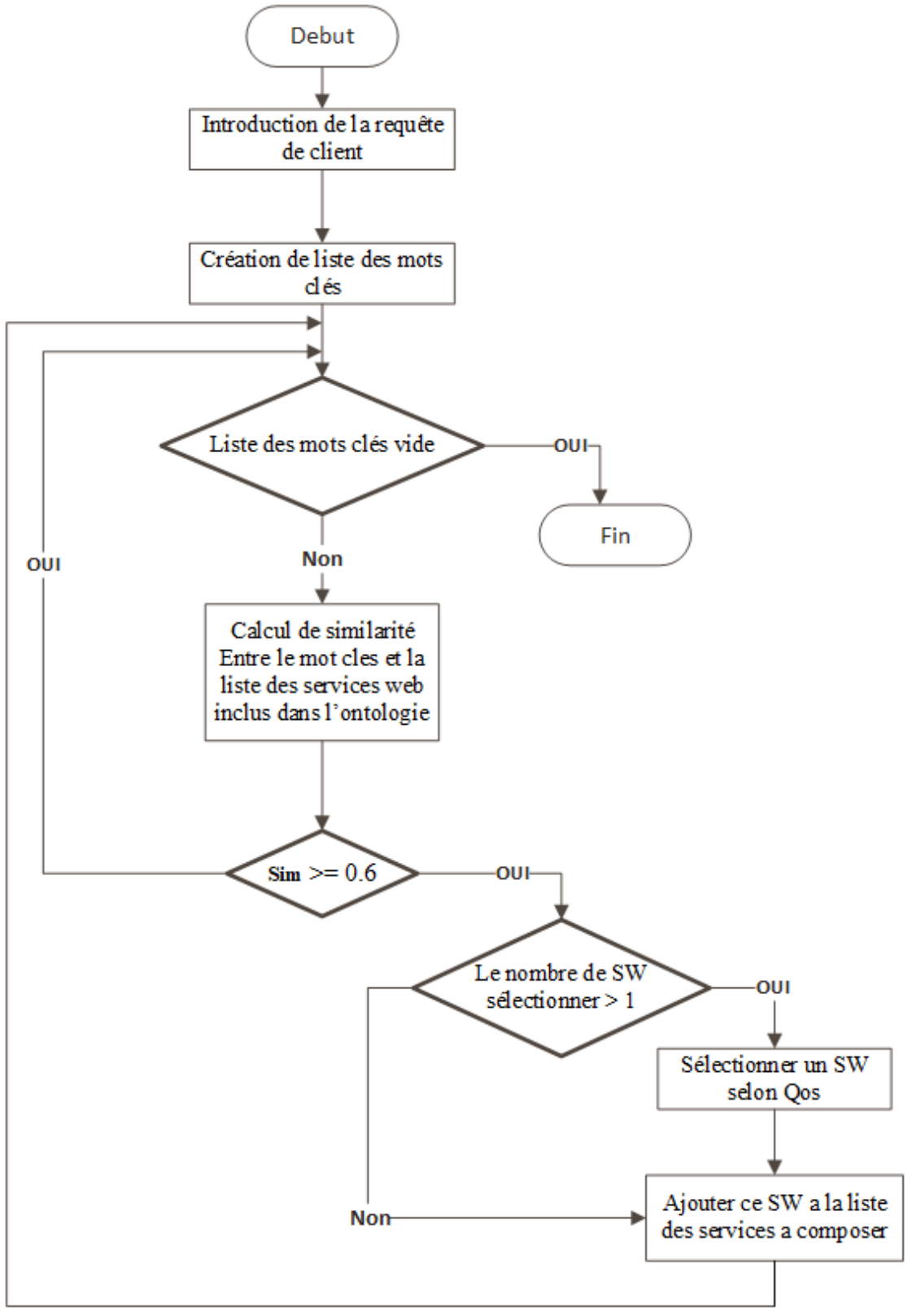


Figure 18 : organigramme du modèle de découverte

La figure suivante (**figure 19**) représente un schéma explicatif du modèle de composition :

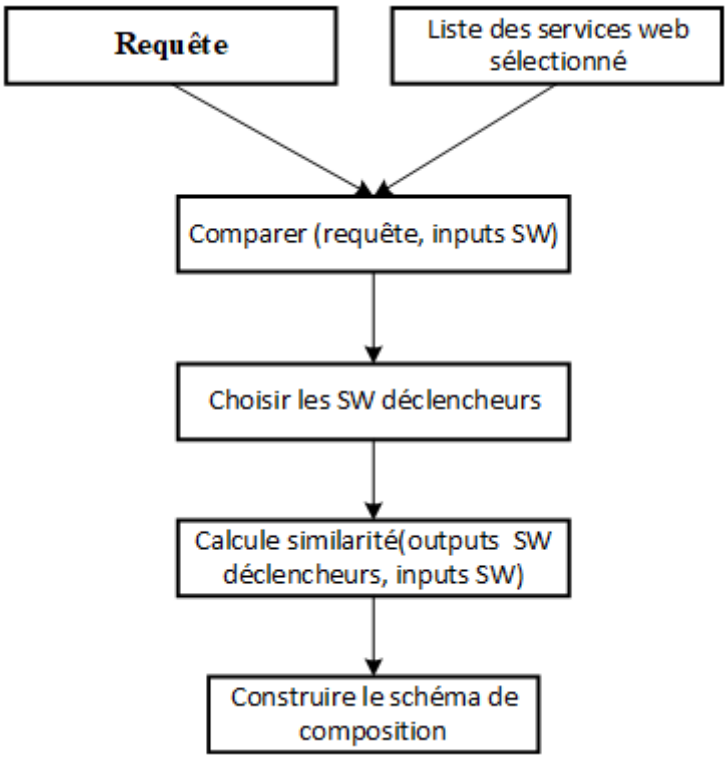


Figure 19 : schéma explicatif des étapes de la composition

IV. Etude de cas d'utilisation :

L'objectif de cette phase est implémenter l'ontologie de découverte et de composition des services web. Cette ontologie permet à la machine de manipuler les services web. Notre ontologie est réalisée par l'outil « OWL-S Editor For Protégé » et la découverte et la composition des services web sont implémentés par le langage python.

IV.1. Description des services web utilisés :

Dans l'ontologie on a une classe « service web », on a créer des instances de cette classes, ces derniers sont représentés dans le tableau suivant :

Nom de service web	Paramétres		QOS		
	Inputs	Outputs	Temp réponse	Cout exécution	Avis utilisateur
Etat civil01	Numéro identification nationale(NIN)	NIN Nom c, Prénom c, Date de naissance, Sexe c, Situation familiale	10	9	10
Etat civil02	Numéro identification nationale(NIN)	NIN Nom c, Prénom c, Date de naissance, Sexe c, Situation familiale	15	20	2
Diplôme universitaire	NIN Nom étudiant, Prénom étudiant, Date de naissance,	Identifiant du diplôme Nom du diplôme Nature de diplôme Nom de l'université	3	18	8
Diplôme éducation	Nom de l'élève, Prénom de l'élève, Date de naissance de	Identifiant du diplôme Nom du diplôme Nature de diplôme	14	20	8

	l'élève, Nom du diplôme	Institution			
Diplôme professionnel	NIN Nom du stagiaire, Prénom du stagiaire, Date de naissance du stagiaire, Nom du diplôme	Identifiant du diplôme Nom du diplôme Nature de diplôme Institution de livraison Institution	43	41	9
Qualification	NIN, Identifiant du diplôme	Liste de (nom d'ingénieur Description de l'expérience Date de début Date de fin)	4	18	8
Etat santé	Nom du patient, Prénom du patient, Date de naissance du patient	NIN Dossier médicale	3	17	7
Casier judiciaire	NIN, Nom citoyen, Prénom citoyen,	Casier judiciaire Numéro identification	5	18	8

	Type de casier judiciaire	nationale			
Reserve vol 01	Nom voyageur, prénom voyageur, NIN, Ville départ, ville arrivé, date départ, date arrivé	Code vol, Nom voyageur, prénom voyageur Ville départ, ville arrivé, date départ, date arrivé prix vol	24	24	6
Reserve vol 02	Nom voyageur, prénom voyageur, NIN, Ville départ, ville arrivé, date départ, date arrivé	Code vol, Nom voyageur, prénom voyageur Ville départ, ville arrivé, date départ, date arrivé prix vol	25	25	8
Reserve hôtel	Nom visiteur, prénom visiteur, NIN, ville	Nom visiteur, prénom visiteur, adresse, Prix	27	20	5
Recherche étudiant	Matricule	Nom, prénom, date naissance, spécialité	13	29	8
Rechercher livre	Matricule,	Livre	5	24	7

	code livre, date				
Location voiture	Nom locataire, prénom locataire, date naissance, NIN, ville, date de location, date retour	Code location, nom locataire, prénom locataire, adresse location, prix	9	23	10

Tableau 06 : liste des services web utilisés.

IV.2. Scénario de la découverte :

L'utilisateur doit introduire une requête dont le but de rechercher un ensemble de service web. Le modèle de découverte détermine les services web pertinents à cette requête en calculant la similarité sémantique entre les noms des services web et la requête introduite par l'utilisateur, ceux qui ont une similarité supérieure à **0.5** ils seront sélectionné.

Pour notre exemple on a introduit la requête suivante : **« Recrutement professeur, diplôme universitaire, état santé, état civil, casier judiciaire, numéro identification nationale, type casier judiciaire»**

Après le calcule de similarité, le programme a retourné les services web suivants :

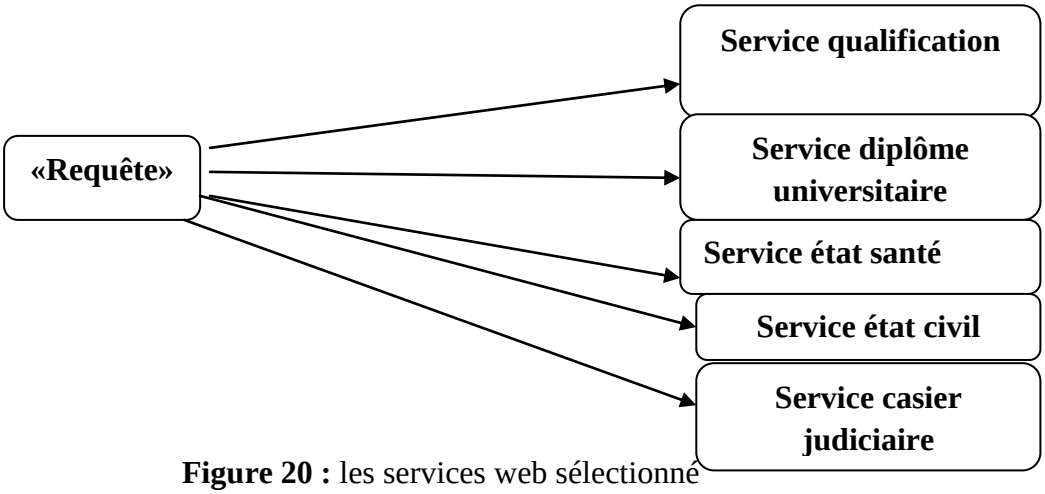


Figure 20 : les services web sélectionné

IV.3. Scénario de la composition :

Après la sélection des services web pertinent selon la requête de l'utilisateur (scénario de la découverte **figure 20)**, on va faire la composition de ces services, tout d'abord on calcule la similarité entre les entrés des services web avec les données entrés par l'utilisateur dans la requête afin de sélectionner le premier service qui va être exécuté, ensuite on calcule la similarité entre les sorties et les entrés des services web sélectionnés pour enfin obtenir le schéma de composition.(**figure 21**)

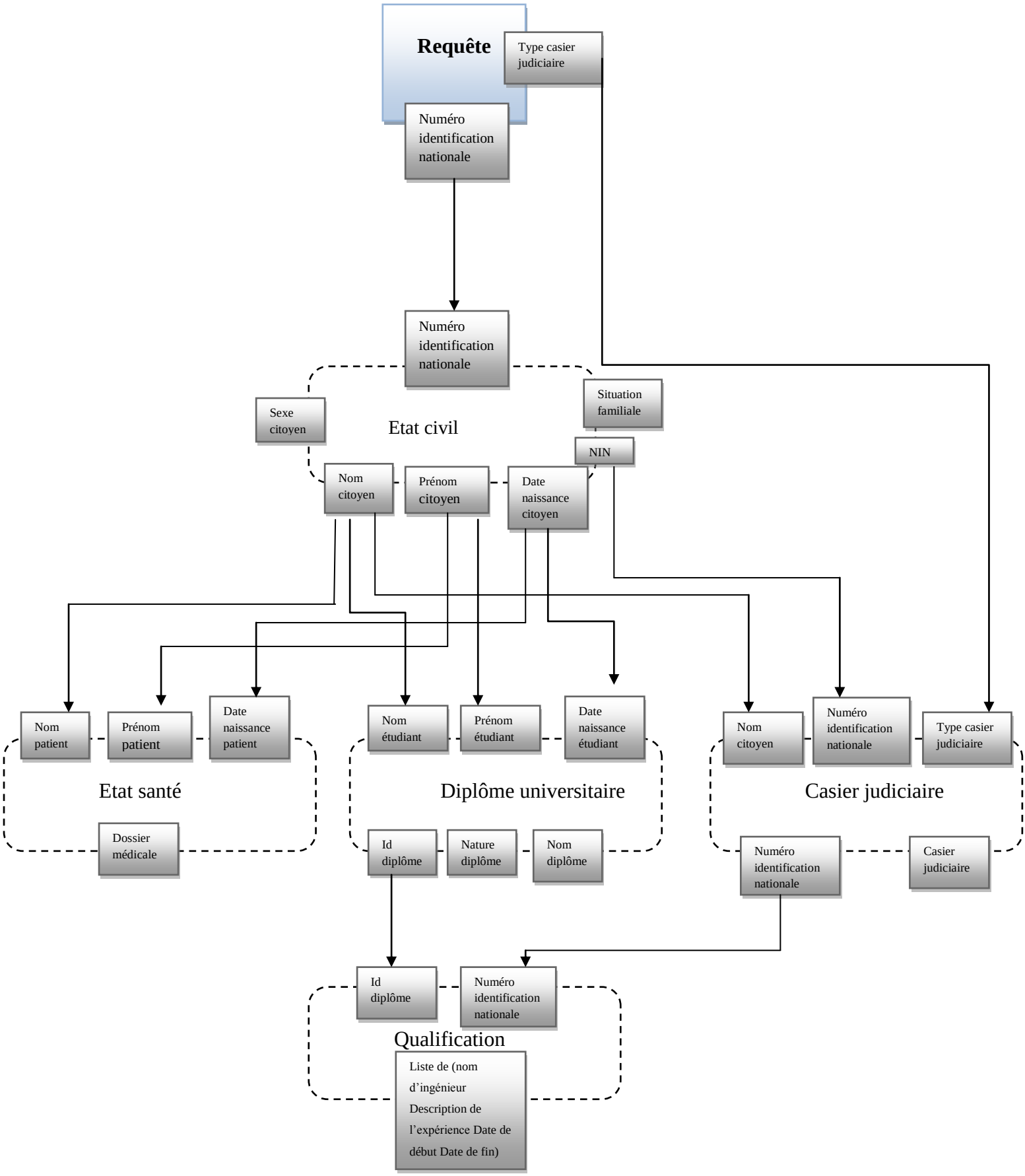


Figure 21 : schéma de composition

V. Résultat et discussions :

Les donnés textuels peuvent être mesuré en comparant les chaines de caractères cette mesure fournit une valeur obtenue selon un algorithme.

Similarité sémantique :

Consiste à calculer la ressemblance de deux chaines de caractères, deux concepts en se basant sur une base de connaissance. Nous avons utilisé le WordNet qui est une base de donné lexicographie.

Le tableau suivant (tableau 07) montre les résultats des similarités entre les mots clé de la requête **« recrutement professeur, diplôme universitaire, état santé, état civil, casier judiciaire »** et les noms des services web :

SW / Mot Clé	Recrutement professeur	Diplôme universitaire	Etat santé	Etat civil	Casier judiciaire
Etat civil01	0.30	0.41	0.36	**0.56**	0.16
Etat civil02	0.53	0.41	0.36	**0.56**	0.16
Diplôme professionnel	0.38	**0.50**	0.35	0.42	0.44
diplôme universitaire	0.48	**0.60**	0.26	0.41	0.24
Recherche étudiant	0.36	0.35	0.24	0.32	0.25
Recherche livre	0.41	0.35	0.32	0.28	0.46
Qualification	**0.53**	0.20	0.38	0.19	0.43
Réserve vol	0.40	0.32	0.4	0.32	0.46
Réserve vol1	0.40	0.32	0.4	0.32	0.46
Réserve hôtel	0.37	0.45	0.31	0.42	0.2
Location	0.34	0.33	0.26	0.28	0.28

voiture					
Etat santé	0.16	0.26	**0.59**	0.36	0.31
Diplôme éducation	0.49	0.35	0.36	0.28	**0.5**
Casier judiciaire	**0.5**	0.24	0.31	0.16	**1**

Tableau 07: résultat de similarité entre la requête et les services web

La requête donné est testé avec tous les noms des services web en calculant la similarité sémantique le résultat est varié entre (0 et 1). Le seuil minimum choisi est 0.5, donc on a pris que les services web ayant une similarité >0.5.

A partir du tableau on a remarqué que le programme a sélectionné pour chaque mot clé un ensemble de service web ayant des similarités >=0.5.

Le tableau suivant (**tableau 08**) représente le nombre de services web choisi pour chaque mot clé :

Mots clé	Nombre de service sélectionné
Recrutement professeur	2
Diplôme universitaire	2
Etat santé	1
Etat civil	2
Casier judicaire	2

Tableau 08 : le nombre de service web sélectionné pour chaque mot clé

Pour le mot clé « état civil », on remarque qu'on a obtenus deux services web ayant une similarité supérieure à 0.5 mais avec des mesures de QOS inégal :

Services choisi	QOS		
	Temp de réponse	Cout exécution	Avis utilisateur
Etat civi01	**10**	**9**	**10**
Etat civil02	15	20	2

Tableau 09: choix d'un service web selon QOS.

Dans ce cas on va choisir le service « état civil01 » car il a QOS meilleur que « état civil02 ».

Après cela, on va choisir pour chaque mot clé un seul service web en basant sur les concepts QOS, celui qui a les meilleurs QOS va être choisi.

Le tableau suivant (**tableau 10**) montre le service web choisis selon QOS pour chaque mot clé :

Mot clé	Nom de service web
Recrutement professeur	Service qualification
Diplôme universitaire	Service diplôme universitaire
Eta santé	Service état santé
Etat civil	Service état civil 01
Casier judiciaire	Service casier judiciaire

Tableau 10 : les services web sélectionné selon QOS pour chaque mot clé.

VI. Conclusion :

Ce chapitre a été consacré à la mise en œuvre et l'implémentation de notre approche de créer une ontologie pour la découverte et la composition dynamique. Nous avons commencé par présenter les outils et les langages utiliser durant le développement ainsi que l'environnement de travail. Ensuite, nous avons faites la mise en œuvre de l'ontologie avec des architectures, cette étape est suivie par les scénarios de la découverte et la composition. Enfin, nous avons présenté les résultats retournés avec le test.

Conclusion générale :

Le web représente une grande source d'informations constitué par des données, ces informations peuvent être utiles par l'utilisateur humain mais difficile pour le traitement automatique, les recherches sur les services web ont mené à utiliser les ontologies pour fournir une représentation de l'information sémantique afin de pouvoir les exploiter automatiquement par des machines.

L'ontologie joue un rôle très important dans le web sémantique, elle sert à modéliser un ensemble de connaissances donné dans un domaine et permet de faire un vocabulaire commun et de développer des services web sémantique.

Le modèle qui ont a proposé de la découverte utilise des similarités entre la requête de l'utilisateur et les noms des services web. Un algorithme de composition est proposé pour faire la composition dynamique des services web il est basé sur les similarités sémantique entre les entrées et les sorties des services web.

A travers ce modeste travail on a pu acquérir des compétences dans différents domaines nous citons :

- En web sémantique
- En ingénierie ontologique
- En fonctionnement des services web

Perspective

Le travail qu'on a réalisé peut être amélioré et enrichi afin d'en faire une ontologie plus performante.

Parmi les perspectives à prendre en compte pour améliorer notre travail, on va citer :

- La construction du fichier BPEL.
- proposer une méthode formelle pour vérifier les résultats de la composition.

Bibliographie :

[1] H. G. Ahcine, “Construction d’une Ontologie pour le WEB Sémantique Table des matières.”

[2] J. Plu, “Introduction au Web Sémantique,” in *Intelligence*, 2011, pp. 1–158.

[3] Q. T. T. PHAN, “Ontologies et Web Services,” 2005.

[4] B. Espinasse, “Introduction au Web Sémantique Bref historique du Web,” 2010.

[5] J. Charlet, P. Laublet, and C. Reynaud, “Web sémantique,” 2003.

[6] S. Boutemedjet, “Web Sémantique et e-Learning,” 2004.

[7] J. Chaumier, *Les ontologies*, A.D.B.S., vol. 12. 2007.

[8] B. Hanane and A. MAZOUZI, “Construction et manipulation d’une ontologie médicale,” Université Abou Bekr Belkaid Tlemcen UABT^2, 2011.

[9] GOUMEIDANE Farida, “Mémoire Utilisation des technologies de services Web sémantiques dans le domaine médical,” 2013.

[10] M. N. Justin, “WEB SERVICES : ETUDE ET CONCEPTION D’UNE PLATEFORME DE SERVICES POUR UN ENVIRONNEMENT NUMÉRIQUE DE TRAVAIL D’UNIVERSITÉ,” 2012.

[11] B. Swartout, R. Patil, K. Knight, and T. Russ, “Ontosaurus: A Tool for Browsing and Editing Ontologie.”

[12] Http://www.w3.org/2001/sw/#owl, “Web Ontology Language (OWL).” https://www.w3.org/OWL/.

[13] C. Pennington, J. Cardoso, J. A. Miller, R. S. Patterson, and I. Vasquez, *Introduction to web services*, no. January. 2007.

[14] T. Zineb and M. Bouhadjar, “La découverte des services web à base de Dominance probabiliste et vote majoritaire,” 2017.

[15] D. Hanane and L. Sihem, “Une approche formelle pour la composition des

services web," 2015.

[16] H. FethAllah, "Composition et interopération des services web sémantiques," Université Abou Bekr Belkaid Tlemcen, 2014.

[17] B. Boubekeur and Y. I. Eddine, "Composition des services Web : Etat de l ' art," 2016.

[18] B. Amina, "Thème Composition des Services Web Sémantiques À base d ' Algorithmes Génétiques," 2012.

[19] D. Thakker, T. Osman, and D. Al-Dabass, "S-CBR: Semantic case based reasoner for web services discovery and matchmaking," *20th Eur. Conf. Model. Simul. Model. Methodol. Simul. Key Technol. Acad. Ind. ECMS 2006*, no. January, pp. 723–729, 2006.

[20] C. Zeginis, "Monitoring the QoS of Web Services using SLAs - Computing Metrics for Composed Services," 2009.

[21] B. Samir, "Sélection de service web basée sur la FOA (fruit Fly optimization algorithm)," 2017.

[22] T. El Bahi, "Modélisation des règles de récitation du coran : l'approche sémantique des ontologies. Dirigé," Alger, 2009.

[23] R. Lawson, *Web scraping with Python*, Packt Publ. 2015.

[24] S. Overflow, *Free ebook, learning beautifulsoup*. 2019.

[25] "codecs." https://pymotw.com/2/codecs/.

[26] "anaconda documentation." https://docs.anaconda.com/#anaconda-documentation.

[27] "spyder documentation." https://docs.spyder-ide.org/current/index.html.