

République algérienne démocratique et populaire
Ministère de l'enseignement supérieur et de la recherche scientifique

Université -Ain-Temouchent- Belhadj Bouchaib
Faculté des Sciences et de la Technologie
Département de Mathématiques et Informatique



Projet de fin d'études

Pour l'obtention du diplôme de Master en Informatique

Spécialité : Réseaux et Ingénierie des Données

Thème

**Systeme Collaboratif sémantique pour la sélection et la
recommandation de services Web : Approche basée Filtrage
collaboratif**

Présenté Par :

1. BELHAFIANE Sabrina Wissam .
2. RAHMANI Houria .

Devant le jury composé de :

Président :	Mr. BENARIBI Fethi Imad	MAA	UAT.B.B (Ain Temouchent)
Examinatrice :	Melle. BOUHALOUAN Djamila	MAA	UAT.B.B (Ain Temouchent)
Encadrante :	Mme. ABDERRAHIM Naziha	MCB	UAT.B.B (Ain Temouchent)

Année universitaire 2020/2021

Remerciements

Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce modeste travail.

En second lieu, nous tenons à remercier notre encadrant Mme : Abderrahim Naziha, ses précieux conseils et son aide durant toute la période du travail.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre mémoire en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

Dédicace

À ma famille, mes amis et tous ceux qui m'ont soutenu tout au long de ce travail.

Sabrina Wissam

Dédicace

À mes parents qui m'ont beaucoup aidé et encouragé pour entamer ce long chemin d'étude, mes frères et mes amis.

Houria

Table des matières

Introduction générale	10
I Les services web	12
1.1 Introduction	13
1.2 L'architecture SOA	13
1.2.1 Définition	13
1.2.2 Les acteurs de SOA	13
1.2.3 Avantages de SOA	14
1.2.4 Inconvénients de SOA	15
1.3 Définitions des services web	15
1.3.1 Définition 1	15
1.3.2 Définition 2	15
1.3.3 Définition 3	15
1.3.4 Définition du W3C	15
1.4 Caractéristiques des services web	16
1.5 Architecture en couches des services web	16
1.6 Les technologies des services web	18
1.6.1 XML (Extensible Markup Language)	18
1.6.2 SOAP (Simple Object Access Protocol)	19
1.6.3 WSDL (Web Services Description Language)	22
1.6.4 UDDI (Universal Description Discovery and Integration)	24
1.7 Scénario de fonctionnement des services web	26
1.8 La découverte et la sélection des services web	27
1.8.1 La découverte	27
1.8.2 La découverte sémantique	28
1.8.3 La sélection	28
1.9 La composition des services web	29
1.9.1 Définition	29

1.9.2	Types de composition de services Web	29
1.10	Les inconvénients des services web	31
1.11	Services web sémantiques	31
1.11.1	Définition	31
1.11.2	Langages de description des services web sémantiques	32
1.12	Conclusion	35
II	Les systèmes de recommandation	36
2.1	Introduction	37
2.2	Historique des systèmes de recommandation	37
2.3	Définition	38
2.4	Notions liées aux systèmes de recommandation	38
2.4.1	Notion de Profil	38
2.4.2	Notion de la communauté	39
2.4.3	Notion du vote	39
2.5	Objectifs des systèmes de recommandation	39
2.6	Classification des systèmes de recommandation	40
2.7	Les différentes techniques de recommandation	40
2.7.1	Le filtrage collaboratif (FC)	40
2.7.2	Le filtrage basé sur le contenu (FBC)	43
2.7.3	Le filtrage démographique	44
2.7.4	Le filtrage basé utilité	44
2.7.5	Le filtrage basé connaissance	44
2.7.6	Le filtrage hybride	45
2.8	Avantages et inconvénients des systèmes de recommandation	46
2.9	Systèmes de recommandation sémantiques	48
2.9.1	Systèmes de filtrage d'information à base d'ontologies	48
2.9.2	Systèmes d'annotations collaboratives	48
2.10	Conclusion	49
III	Etat de l'art : les systèmes de recommandation de services web	50
3.1	Introduction	51
3.2	Les systèmes de recommandation de services web	51
3.3	Tableau comparatif	54
3.4	Synthèse	56
3.5	Conclusion	56

IV Système collaboratif sémantique pour la sélection et la recommandation	
de services Web	57
4.1 Introduction	58
4.2 Architecture de l'approche	58
4.2.1 Filtrage collaboratif basé item	59
4.2.2 Filtrage sémantique	61
4.2.3 Hybridation	63
4.3 Conception	64
4.3.1 Diagramme de séquence	64
4.3.2 Diagramme de classes	64
4.4 Outils d'implémentation	65
4.4.1 Java	65
4.4.2 Netbeans	66
4.4.3 JDom	66
4.4.4 Collection de test	67
4.5 Implémentation de l'application	68
4.6 Évaluation expérimentale	71
4.7 Conclusion	75
Conclusion générale et perspectives	76

Table des figures

1.1	Principaux acteurs de SOA.	14
1.2	Architecture en couches des services web.	17
1.3	Structure d'un message XML.	19
1.4	Messages SOAP dans une application client/serveur.	20
1.5	Structure d'un message SOAP.	20
1.6	Exemple de SOAP Header.	21
1.7	Exemple de demande SOAP.	21
1.8	Exemple de réponse SOAP.	22
1.9	Exemple de SOAP Fault.	22
1.10	Structure d'un document WSDL.	23
1.11	Exemple de description WSDL.	24
1.12	Structure des données UDDI.	25
1.13	Scénario de fonctionnement des services web.	26
1.14	La découverte des services web.	28
1.15	La sélection des services web.	29
1.16	Vue générale de l'orchestration.	30
1.17	Vue générale de la chorégraphie.	31
1.18	Ontologie OWL-S.	32
1.19	Les variantes de WSML.	33
1.20	Architecture de SAWSDL proposée par le W3C.	34
1.21	Annotation sémantique d'une description WSDL.	34
2.1	La matrice Utilisateur/Document.	41
2.2	Extension de la matrice utilisateur/document par les annotations.	49
4.1	L'architecture de l'approche proposée.	58
4.2	Le processus de recommandation basé sur le filtrage collaboratif.	60
4.3	Le processus de recommandation basé sur le filtrage sémantique.	62
4.4	Le processus de recommandation basé sur le filtrage collaboratif sémantique.	63

4.5	Diagramme de séquence.	64
4.6	Diagramme de classes.	65
4.7	Interface principale de SCRS.	68
4.8	Fichier OWLS.	69
4.9	Fichier XML des votes.	70
4.10	Valeurs de DCG_{sim} en se basant sur le filtrage collaboratif et/ou le filtrage sémantique.	71
4.11	Evaluation basée sur MAE.	74
4.12	Evaluation basée sur RMSE.	75

Liste des tableaux

2.1	Avantages et inconvénients des techniques de recommandation.	47
3.1	Les approches qui traitent les systèmes de recommandation de services Web.	55
4.1	Évaluation de la performance de prédiction du filtrage collaboratif sémantique.	73
4.2	Évaluation de la performance de prédiction du filtrage collaboratif.	73
4.3	Évaluation de la performance de prédiction du filtrage sémantique.	74

Introduction générale

Les systèmes d'information actuels se caractérisent par leur volume important de données, leur hétérogénéité et leur incapacité à s'adapter pleinement aux besoins des utilisateurs. Compte tenu de l'état actuel de ces systèmes en termes d'hétérogénéité des domaines et des sources et de la surcharge d'informations. Il est difficile pour les utilisateurs de trouver les données et les informations qu'ils souhaitent.

Afin de résoudre ce problème, les systèmes de recommandation ont vu le jour, leurs objectifs est de filtrer et adapter les informations pour chaque utilisateur afin de minimiser le temps que les utilisateurs passent à rechercher et de suggérer des ressources connexes qu'ils n'auraient pas consultées.

Les méthodes généralement utilisées pour la recommandation sont soit basées sur la similarité de l'utilisateur avec les autres utilisateurs (les approches collaboratives), soit sur le contenu, ou bien sur les deux combinées.

Les nouveaux systèmes de recommandation qui sont apparus ont intégré les technologies du web sémantique pour améliorer la qualité de la recommandation en appliquant une méthode d'inférence basée sur le domaine et/ou en appliquant une méthode de similarité sémantique.

Dans ce mémoire nous proposons un système collaboratif sémantique pour la sélection et la recommandation de services Web qui sont de plus en plus utilisés lors du développement des applications web. Ce système calcule la similarité entre les services web afin de générer des recommandations de services web à un utilisateur cible et cela en utilisant une technique de filtrage hybride : filtrage collaboratif basé sur la mémoire et filtrage sémantique.

Ce document comporte quatre chapitres qui se répartissent comme suit :

Le premier chapitre, présente la notion de services web, leurs caractéristiques, leurs fonctionnements, ainsi que les principaux standards qu'ils supportent tel que XML, SOAP, WSDL.

Le deuxième chapitre présente une vue générale sur les systèmes de recommandation, leurs histoire, leurs objectifs, les différentes techniques de recommandation qui existent,

ainsi que leurs avantages et inconvénients.

Le troisième chapitre est consacré à l'état de l'art des travaux de la littérature utilisant les principales techniques de recommandation de services web.

Le quatrième et dernier chapitre est consacré pour la présentation de l'application réalisée, ainsi que les principaux outils utilisés. Et enfin, une conclusion générale qui résume notre travail avec quelques perspectives.

Chapitre I

Les services web

1.1 Introduction

La technologie des services web est un moyen qui permet la communication et l'échange de données entre applications et systèmes de différentes plates-formes via le réseau, ils introduisent un nouveau modèle de développement basé sur ce que l'on appelle l'architecture orientée services SOA qui leur permet une meilleure interaction les uns avec les autres.

Dans ce chapitre, nous présenterons tout d'abord l'architecture SOA, puis nous détaillons la notion de services web ainsi que leur fonctionnement, l'architecture en couche et les technologies utilisées. Ensuite nous introduisons le concept de services web sémantiques et les différentes approches qui ont été développées pour automatiser les tâches liées au cycle de vie des services web.

1.2 L'architecture SOA

1.2.1 Définition

L'architecture orientée service (SOA) est un mode de conception de logiciel dans lequel les services sont fournis aux autres composants par des composants d'application, par un protocole de communication sur un réseau. Ses principes sont indépendants des fournisseurs et d'autres techniques. Dans l'architecture orientée service, un certain nombre de services communiquent entre eux, de l'une des deux façons suivantes : En passant des données ou via deux ou plusieurs services pour coordonner l'activité [1].

1.2.2 Les acteurs de SOA

SOA implique trois entités d'interaction :

- **Service fournisseur (service provider)** : On peut l'appeler fournisseur il met le service web en application et il le rend disponible pour tout le monde sur internet.
- **Service consommateur (service requestor)** : C'est un client demandeur ou un consommateur qui fait la demande d'un service web bien précis pour répondre à ses besoins.
- **Service annuaire** : Le registre fournit un endroit où le consommateur peut trouver des nouveaux services web et le fournisseur dispose une description pour des nouveaux services web [2].

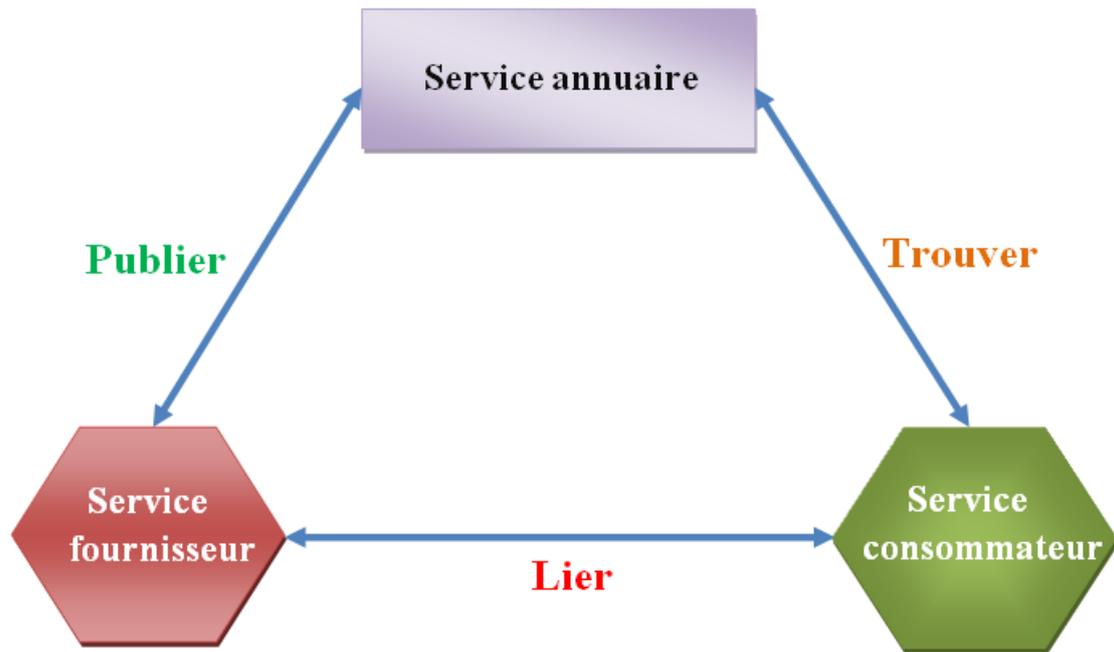


FIGURE 1.1: Principaux acteurs de SOA.

1.2.3 Avantages de SOA

- **Réutilisabilité du service** : Dans SOA, les applications sont faites à partir de services existants. Ainsi, les services peuvent être réutilisés pour faire de nombreuses applications.
- **Maintenance facile** : Comme les services sont indépendants les uns des autres, ils peuvent être mis à jour et modifiés facilement sans affecter d'autres services.
- **Plateforme indépendante** : SOA permet de réaliser une application complexe en combinant des services choisis à partir de différentes sources, indépendantes de la plateforme.
- **Disponibilité** : Les installations SOA sont facilement accessibles à tous sur demande.
- **Fiabilité** : Les applications SOA sont plus fiables car il est facile de déboguer de petits services plutôt que d'énormes codes.
- **Évolutivité** : Les services peuvent s'exécuter sur différents serveurs dans un environnement, ce qui augmente l'évolutivité [3].

1.2.4 Inconvénients de SOA

- **Frais généraux élevés** : Une validation des paramètres d'entrée des services est effectuée chaque fois que les services interagissent, ce qui diminue les performances car cela augmente la charge et le temps de réponse.
- **Investissement élevé** : Un investissement initial énorme est requis pour la SOA.
- **Gestion de services complexes** : Lorsque les services interagissent, ils échangent des messages avec des tâches. le nombre de messages peut aller en millions. Gérer un grand nombre de messages devient une tâche fastidieuse [3].

1.3 Définitions des services web

1.3.1 Définition 1

Un service Web est une application disponible sur Internet et qui utilise un système de messagerie XML normalisé.

1.3.2 Définition 2

Les services Web sont des systèmes d'échange d'informations basés sur XML qui utilisent Internet pour une interaction directe entre les applications. Ces systèmes peuvent inclure des programmes, des objets, des messages ou des documents [4].

1.3.3 Définition 3

Un service Web est un programme informatique permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués. Il s'agit donc d'un ensemble de fonctionnalités exposées sur internet ou sur un intranet, par et pour des applications ou machines, sans intervention humaine, et en temps réel [5].

1.3.4 Définition du W3C

« Un service Web est un composant logiciel identifié par une URI, dont les interfaces publiques sont définies et appelées en 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. » [6].

1.4 Caractéristiques des services web

Un service Web a les caractéristiques suivantes :

- Il est accessible via le réseau.
- Réutilisable et composable : les web services sont des composants logiciel avec possibilité de réutilisation et de composition pour construire de nouveaux services, de manière qu'un service peut agir dans plusieurs compositions.
- Indépendant de la plate-forme.
- Modulaire : Les web services fonctionnent de manière modulaire, Cela implique une flexibilité d'intégration et d'interopération entre ces modules.
- Il dispose d'une interface publique décrite en XML.
- Ses descriptions sont stockées dans un annuaire.
- Conçus pour résoudre des problèmes bien définis et complexes.
- Il communique en utilisant des messages XML, ces messages sont transportés par des protocoles Internet (HTTP, SMTP, FTP, BEEP).
- l'intégration d'application en implémentant des services Web produit des systèmes faiblement couplés, le demandeur du service ne connaît pas forcément le fournisseur [7].

1.5 Architecture en couches des services web

Les services Web emploient un ensemble de technologies qui ont été conçues afin de respecter une structure en couches appelée la pile protocolaire des services Web qui aide à définir, découvrir, implémenter et faire interagir les services Web les uns avec les autres [8].

La figure 1.2 est une représentation simplifiée de l'architecture en couches des services web :

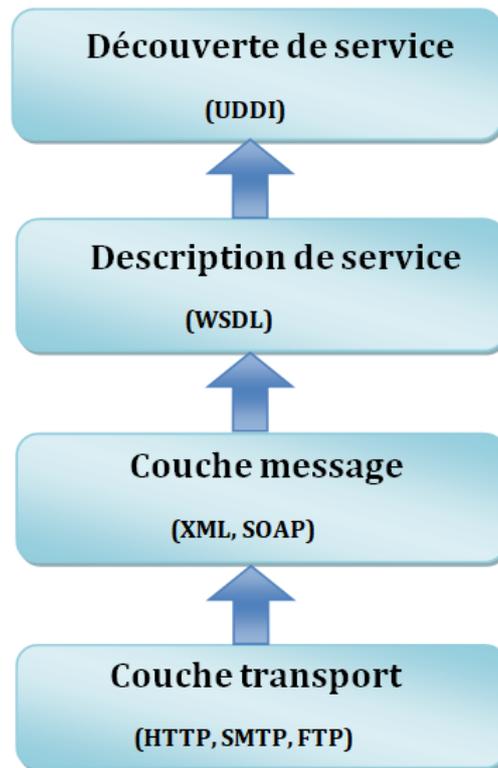


FIGURE 1.2: Architecture en couches des services web.

- **Couche transport** : Transporte les messages XML entre les applications réseau et utilise les protocoles de transport courants tels que HTTP , SMTP , FTP , etc.
- **Couche message** : Cette couche est responsable du formatage des données échangées de sorte que les messages peuvent être compris à chaque extrémité. Actuellement, deux styles architecturaux totalement différents sont utilisés pour ces échanges de données. Nous avons d'un côté l'architecture orientée opérations distribuées (protocoles RPC) basée sur XML et qui comprend XML-RPC et SOAP et de l'autre côté une architecture orientée ressources Web, REST (Representational State Transfer) qui se base uniquement sur le bon usage des principes du Web (en particulier, le protocole HTTP) [9].
- **Couche de description de service** : Aide à décrire l'interface publique d'un service Web. Une description de service représente les informations nécessaires pour utiliser un service et facilite la visibilité et l'interaction entre les consommateurs et les fournisseurs de services, le format le plus utilisé est le WSDL (Web Service Description Language) qui aide le client à identifier les offres d'un service Web [8].
- **Couche de découverte de service** : Centralise les services avec un registre commun et aide les clients à rechercher les services disponibles sur le réseau. Actuel-

lement, la découverte des services est assurée par un annuaire UDDI (Universal Description, Discovery and Integration) [8].

1.6 Les technologies des services web

Un service Web rend les ressources des applications logicielles disponibles sur les réseaux à l'aide de technologies standard. Les services Web étant basés sur des interfaces standard, ils peuvent communiquer même s'ils s'exécutent sur des systèmes d'exploitation différents et sont écrits dans des langues différentes. Pour cette raison, ils constituent une excellente approche pour créer des applications distribuées qui doivent intégrer divers systèmes sur un réseau [10].

1.6.1 XML (Extensible Markup Language)

Définition

C'est un métalangage informatique de balisage, l'objectif majeur de XML est d'étendre les fonctionnalités de HTML afin de faciliter les échanges d'informations.

Principe

Les messages XML fournissent un langage commun par lequel différentes applications peuvent communiquer entre elles sur un réseau. La plupart des services Web communiquent via XML. Un client envoie un message XML contenant une demande au service Web et le service Web répond avec un message XML contenant les résultats de l'opération. Dans la plupart des cas, ces messages XML sont formatés selon la syntaxe SOAP [10].

Structure d'un message XML

Un document XML est un ensemble d'éléments structurés sous forme de Balises comme s'est montré dans la figure 1.3 :

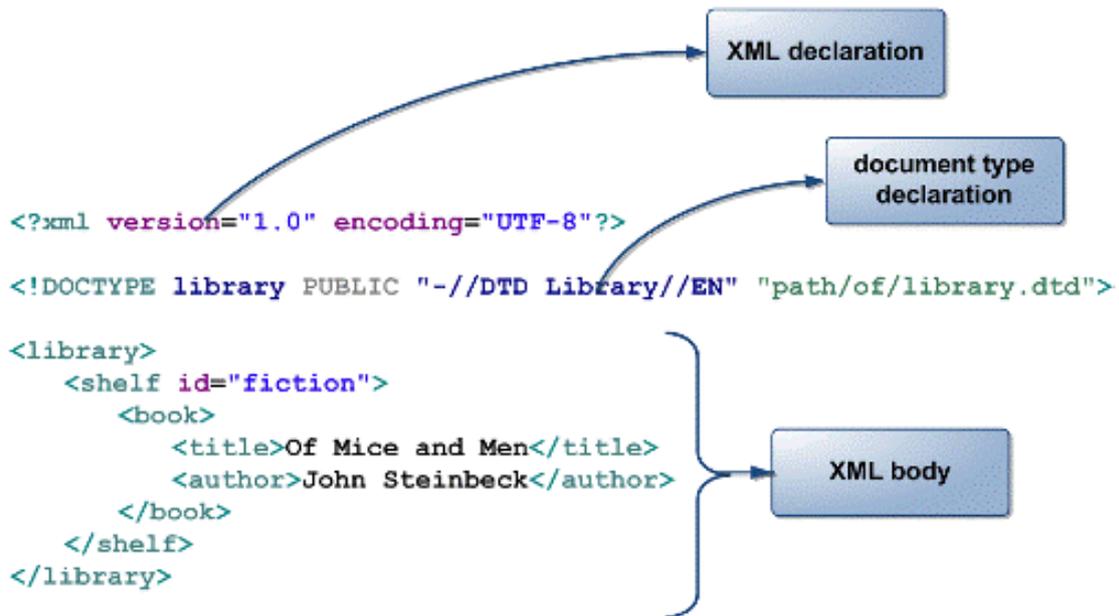


FIGURE 1.3: Structure d'un message XML.

1.6.2 SOAP (Simple Object Access Protocol)

Définition

Protocole basé sur le standard XML pour l'échange de données structurées entre des applications réseaux, Il permet ainsi l'accès aux services web et l'interopérabilité des applications à travers le web.

SOAP assure la partie messaging dans l'architecture des services web il est utilisé pour normaliser le format des messages échangés entre le consommateur et le fournisseur de services web, il est aussi indépendant de tout système d'exploitation et de tout langage de programmation car il utilise XML. Ceci permet une exposition et une consommation de services web avec des outils et des OS différents [11].

SOAP peut être utilisé pour :

- Appeler une méthode d'un service (SOAP RPC).
- échanger un message avec un service (SOAP Messaging).
- Recevoir un message d'un service [11].

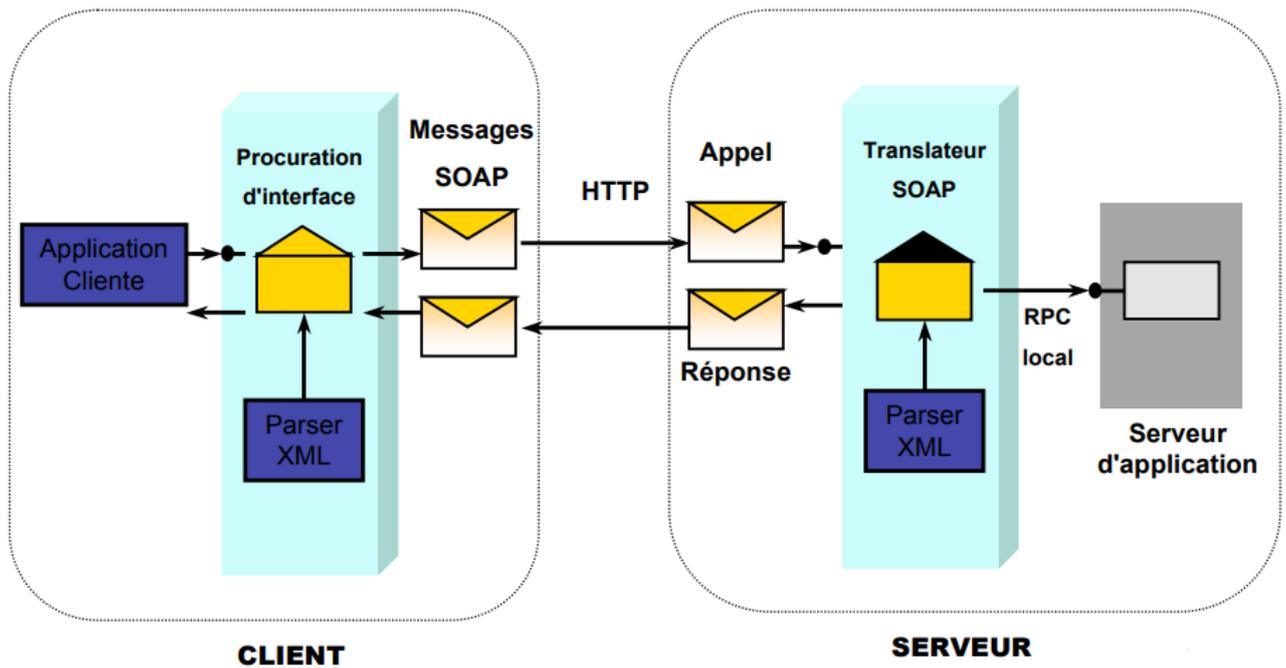


FIGURE 1.4: Messages SOAP dans une application client/serveur.

Structure d'un message SOAP

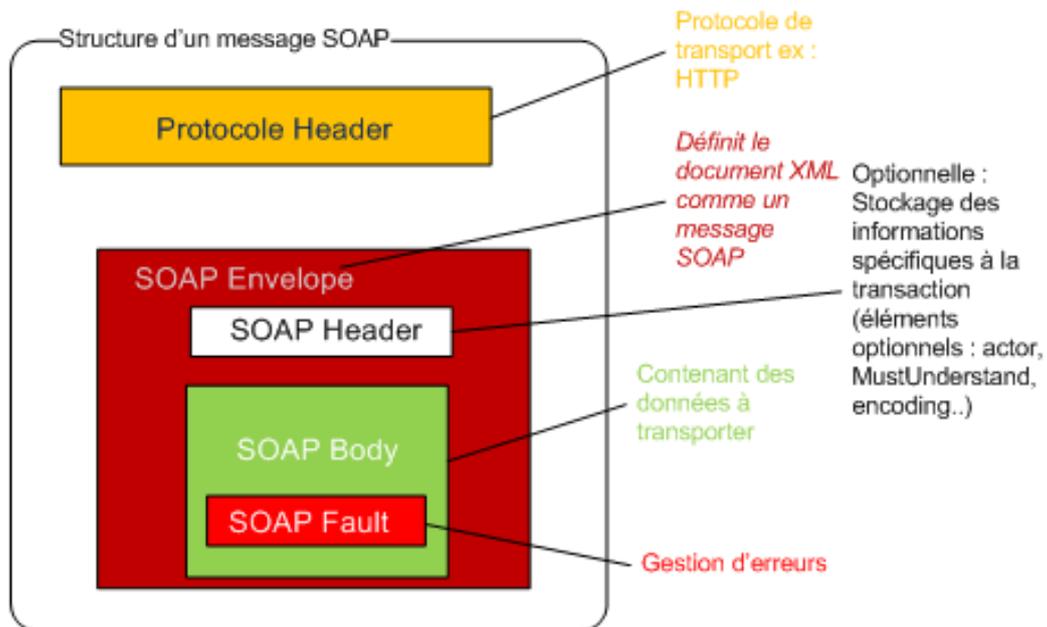


FIGURE 1.5: Structure d'un message SOAP.

- **SOAP Envelope** : C'est lui qui contient le message et ses différents sous-blocs. Il s'agit du bloc racine XML. Il peut contenir un attribut `encodingStyle` dont la valeur

est une URL vers un fichier de typage XML qui décrira les types applicables au message SOAP.

- **SOAP Header** : C'est un bloc optionnel qui contient des informations d'en-têtes sur le message. S'il est présent, ce bloc doit toujours se trouver avant le bloc Body à l'intérieur du bloc Envelope.

```
<soap:Header>
  <m:User xmlns:m="http://www.exemple.com/rights/"
    soap:actor="http://www.exemple.com/rights/RightsManager">
    Charles
  </m:User>
  <m:Session xmlns:m="http://www.exemple.com/session/"
    soap:mustUnderstand="1">12AE3C</m:Session>
  <m:Lang xmlns:m="http://www.exemple.com/lang/"
    soap:actor="http://schemas.xmlsoap.org/soap/next"
    soap:mustUnderstand="0">
    FR
  </m:Lang>
</soap:Header>
```

FIGURE 1.6: Exemple de SOAP Header.

- **SOAP Body** : C'est le bloc qui contient le corps du message. Il doit absolument être présent de manière unique dans chaque message et être contenu dans le bloc Envelope. SOAP ne définit pas comment est structuré le contenu de ce bloc. Cependant, il définit le bloc Fault qui peut s'y trouver.

```
<!-- exemple de demande -->
<soap:Body>
  <m:GetPrice xmlns:m="http://www.exemple.com/prices">
    <m:Item>Pomme</m:Item>
  </m:GetPrice>
</soap:Body>
```

FIGURE 1.7: Exemple de demande SOAP.

```

<!-- exemple de réponse -->
<soap:Body>
  <m:GetPriceResponse xmlns:m="http://www.exemple.com/prices">
    <m:Price>1.90</m:Price>
  </m:GetPriceResponse>
</soap:Body>

```

FIGURE 1.8: Exemple de réponse SOAP.

- **SOAP Fault** : Ce bloc est la seule structure définie par SOAP dans le bloc Body. Il sert à reporter des erreurs lors du traitement du message, ou lors de son transport. Il ne peut apparaître qu'une seule fois par message. Sa présence n'est pas obligatoire [12].

```

<soap:Body>
  <soap:Fault>
    <faultcode>soap:Server</faultcode>
    <faultstring>Impossible de router le message.</faultstring>
    <faultactor>http://www.exemple.com/messageDispatcher</faultactor>
    <detail>
      <m:error xmlns:m="http://www.exemple.com/errors">
        E_NO_ROUTE
      </m:error>
    </detail>
  </soap:Fault>
</soap:Body>

```

FIGURE 1.9: Exemple de SOAP Fault.

1.6.3 WSDL (Web Services Description Language)

Définition

WSDL fournit une description d'un service web afin de permettre son utilisation. Il joue un rôle important dans l'architecture des services en assurant la partie description, il contient toutes les informations nécessaires à l'invocation du service qu'il décrit.

Pour permettre à un client de consommer un service web, ce dernier a besoin d'une description détaillée du service avant de pouvoir interagir avec lui, cette description est fournie dans un document XML.

En résumé WSDL c'est un contrat entre un client et un serveur qui fait état :

- des spécifications d'interfaces qui décrivent toutes les méthodes publiques.

- des spécifications relatives aux types de donnée de messages mis en œuvre dans les questions-réponses.
- des informations liées au protocole de transport utilisé.
- des informations d'adresse permettant de localiser le service décrit [13].

Structure d'un document WSDL

Un document WSDL définit plusieurs éléments :

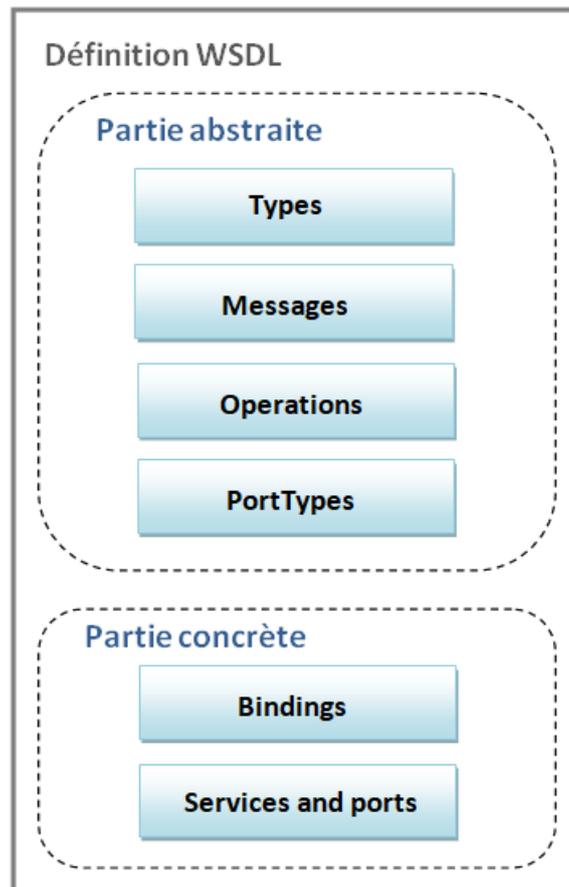


FIGURE 1.10: Structure d'un document WSDL.

Partie abstraite : Contient les éléments qui définissent l'interface du service :

- **Type :** la définition des types de données utilisés.
- **Message :** la définition de la structure d'un message en lui attribuant un nom et en décrivant les éléments qui le composent avec un nom et un type.
- **Operation :** la description d'une action proposée par le service web notamment en précisant les messages en entrée et en sortie.

- **PortType** : la description de toutes les opérations proposées par le service web (interface du service) et identification de cet ensemble avec un nom.

Partie concrète : Permet l'invocation du service :

- **Binding** : la description du protocole de transport et d'encodage utilisé par un PortType afin de pouvoir invoquer un service web.
- **Service** : c'est un ensemble de ports.
- **Port** : référence un Binding [11].

```

<definitions name="CarPrice"
<message name="get_PRICERequest">
  <part name="Car" type="CarType">
  </part>
</message>
<message name="get_PRICEResponse">
  <part name="PRICE" type="PriceType"/>
</message>
<portType name="CarPriceSoap">
  <operation name="get_PRICE">
    <input message="get_PRICERequest"/>
    <output message="get_PRICEResponse"/>
  </operation>
</portType>
<binding name="CarPriceSoapBinding" type="CarPriceSoap">
  <wsdlsoap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="get_PRICE">
    <input> <wsdlsoap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://127.0.0.1/services/CarPrice"/>
    </input>
    <output> <wsdlsoap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://127.0.0.1/services/CarPrice"/>
    </output>
  </operation> </binding>
<service name="CarPriceService">
  <port name="CarPriceSoap" binding="CarPriceSoapBinding">
    <wsdlsoap:address location="http://127.0.0.1/services/CarPrice"/>
  </port> </service>
</definitions>

```

FIGURE 1.11: Exemple de description WSDL.

1.6.4 UDDI (Universal Description Discovery and Integration)

Définition

UDDI est introduit en 2000 par Ariba, Microsoft et IBM. Il a été créé pour publier, découvrir et rechercher des services web. UDDI est un annuaire qui stocke les informations

concernant les services afin de permettre à un client de les retrouver facilement et il permet ainsi aux entreprises de se découvrir et de définir comment elles interagissent sur Internet.

Les données incluses dans un annuaire UDDI sont classées dans trois catégories :

- **Les pages blanches** : elles contiennent la liste des entreprises ainsi que des informations associées à ces dernières. Nous y retrouvons des informations comme le nom de l'entreprise, ses coordonnées, la description de l'entreprise mais également l'ensemble de ses identifiants.
- **Les pages jaunes** : elles permettent une catégorisation des entreprises, elles contiennent la liste des services web de chaque entreprise en format WSDL.
- **Les pages vertes** : elles contiennent des informations techniques sur les services web. Ces informations concernent les descriptions de services et d'information de liaison ou encore les processus métiers associés.

La structure des données UDDI

Un registre UDDI est composé de quatre instances de structures de données persistantes appelées entités exprimées en XML : `businessEntity`, `BusinessService`, `BindingTemplate` et `TModel` :

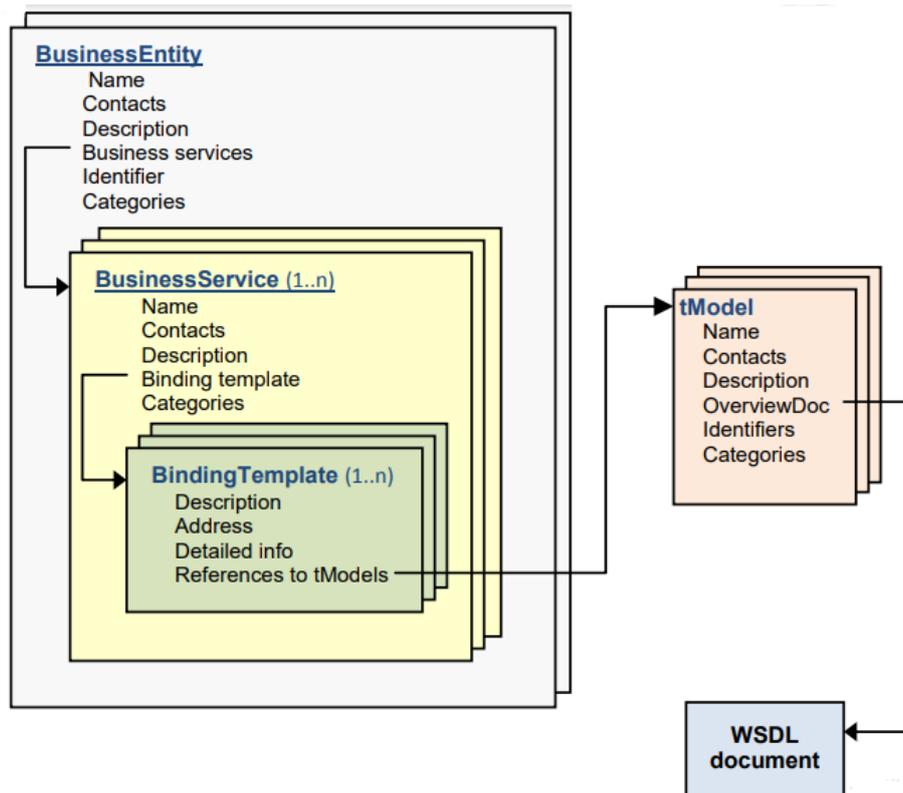


FIGURE 1.12: Structure des données UDDI.

- **L'entité Business (Business Entity)** : Il s'agit des informations concernant l'entreprise qui publie ses services web dans l'annuaire UDDI. On y trouve son nom, sa description, l'adresse de son site internet... (l'équivalent des pages blanches).
- **L'entité Service (Business Service)** : Décrit les services Web associés à l'entité business correspondante, il contient le nom, la description ainsi qu'une liste (optionnelle) de BindingTemplates (l'équivalent des pages jaunes).
- **L'entité modèle de liaison (Binding Template)** : Décrit les points d'accès aux services web et le moyen d'y accéder afin d'invoquer les services (l'équivalent des pages vertes).
- **L'entité modèle technique (tModel)** : Le tModel a pour rôle de décrire les spécifications des services web à enregistrer. Il permet d'associer un service à sa description WSDL [14].

1.7 Scénario de fonctionnement des services web

Les étapes nécessaires à l'accès et à l'invocation d'un service Web selon l'architecture SOA sont présentées dans la figure suivante :

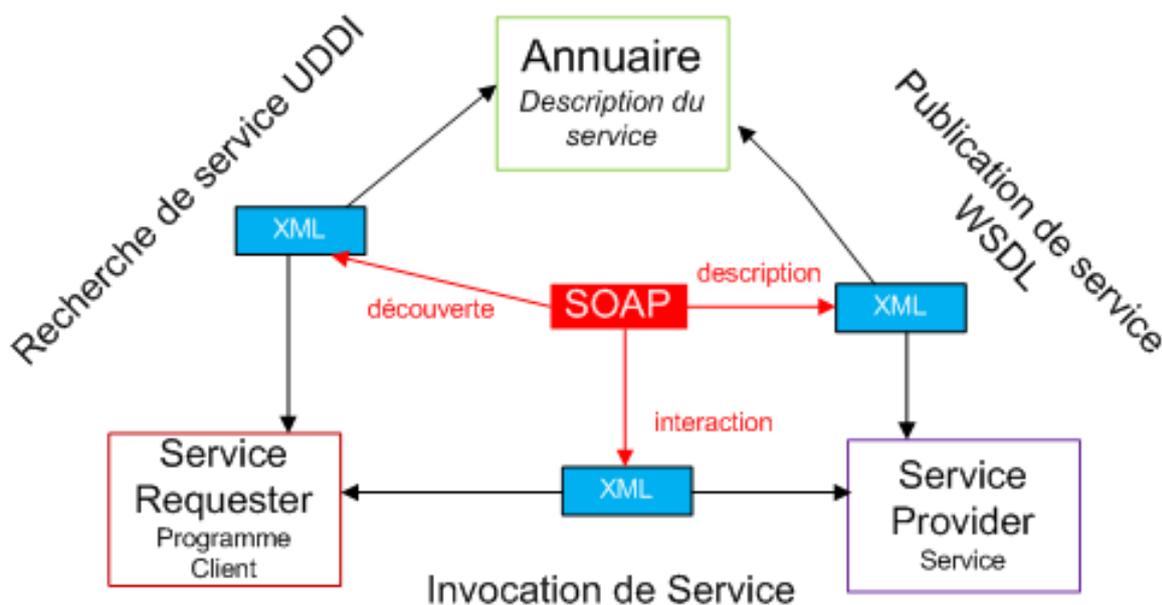


FIGURE 1.13: Scénario de fonctionnement des services web.

1. **Publication de service** : Le fournisseur de service (service provider) définit la description WSDL de son service et la publie dans un annuaire en vue d'être localisé

par des clients.

2. **Recherche de service** : Un client va rechercher un service dont il a besoin dans un annuaire UDDI en spécifiant les critères.
3. **Récupération de la description du service** : Le client reçoit un document WSDL décrivant le service dont il a besoin.
4. **Invocation de service** : Sur la base des informations définies dans la description du service, une communication entre le client et le service Web se fait via le protocole SOAP. Le client appelle le service Web en lui demandant d'exécuter certaines de ses fonctionnalités.
5. **Réception de la réponse** : Le client reçoit une réponse à sa demande.

1.8 La découverte et la sélection des services web

1.8.1 La découverte

Les services sont conçus afin d'être sélectionnés via des mécanismes de recherche. La découverte est permise par la description préalable des services et leur publication au sein d'un registre. Elle est effectuée en deux phases :

- **La recherche** : Elle peut se faire via les registres dans lesquels les fournisseurs ont publiés leurs services Web.
- **La sélection** : Le client doit ensuite sélectionner parmi la collection de services Web issus de l'étape de recherche, le service Web qui convient le mieux à ses attentes.

La recherche et la sélection dans UDDI reposent sur la publication préalablement décrite du service et de son fournisseur. Le futur client peut connaître par l'intermédiaire de l'UDDI : les fournisseurs d'un service, les services proposés par un fournisseur donné et les moyens d'invoquer un service [15].

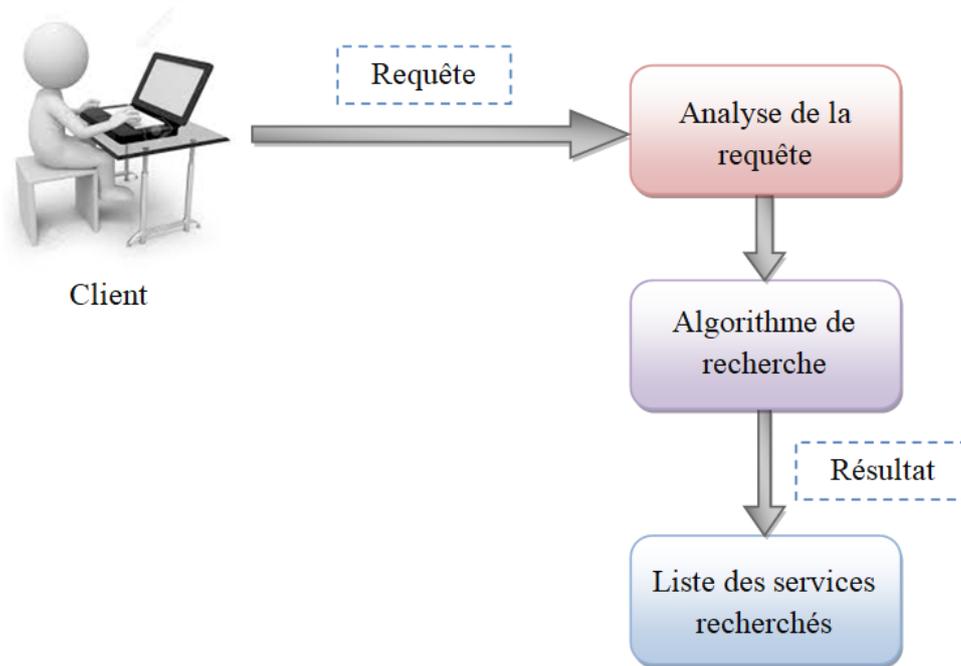


FIGURE 1.14: La découverte des services web.

1.8.2 La découverte sémantique

L'utilisation de la technologie Web sémantique dans la phase de la découverte des services Web donne naissance à la découverte sémantique des services Web. Cette dernière est basée sur le raisonnement sémantique. Elle ajoute à la précision des résultats de recherche traditionnels par rapport aux techniques de découverte de service Web, la précision supplémentaire de « matching » en termes de puissance de calcul [15].

1.8.3 La sélection

La sélection des services Web est l'une des discussions les plus importantes dans la SOA, elle consiste à choisir parmi un groupe de services web découverts ayant des fonctions similaires, ceux qui répondent au mieux aux exigences de l'utilisateur.

Dans une sélection de services web basée sur la Qualité de Service (QoS) deux cas se présentent :

- Si la réponse à la requête d'un client exige la sélection d'un seul service web non composite, alors la sélection est très simple. Le candidat qui présente la meilleure QoS sera désigné pour répondre à la demande du client.
- Si la réponse à la requête d'un client exige la combinaison de plusieurs services existants, alors la sélection dans ce cas sera plus complexe du fait qu'il faut choisir

la combinaison des services composants qui répondent mieux aux besoins des clients [16].

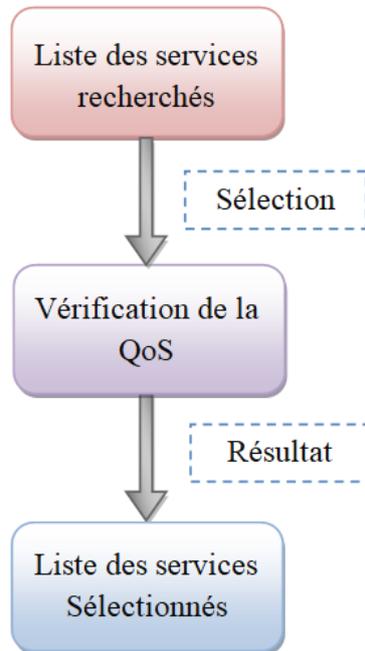


FIGURE 1.15: La sélection des services web.

1.9 La composition des services web

1.9.1 Définition

Composer les services entre eux, c'est entrelacer leurs séquences d'actions, de manière à obtenir des séquences qui correspondent aux séquences d'actions des besoins exprimés par le client dans sa spécification. Nous pouvons définir la composition de services web comme étant l'assemblage, la coordination et la mise en place d'un ensemble de services afin d'obtenir un nouveau service dit composite capable de répondre à des requêtes cliente ne pouvant être satisfaite par l'un des Web services disponible [17].

L'objectif de la composition de service est de créer de nouvelles fonctionnalités en combinant des fonctionnalités offertes par d'autres services existants, composés ou non en vue d'apporter une valeur ajoutée [18].

1.9.2 Types de composition de services Web

- **Orchestration de services Web** : L'orchestration de services Web exige de définir l'enchaînement des services Web selon un canevas prédéfini, et de les exécuter

selon un script d'orchestration. Ces derniers (le canevas et le script) décrivent les interactions entre services Web en identifiant les messages, et en spécifiant la logique et les séquences d'invocation. Le module exécutant le script d'orchestration de services Web est appelé un moteur d'orchestration. Ce dernier est une entité logicielle qui joue le rôle d'intermédiaire entre les services, en les appelants suivant le script d'orchestration [19].

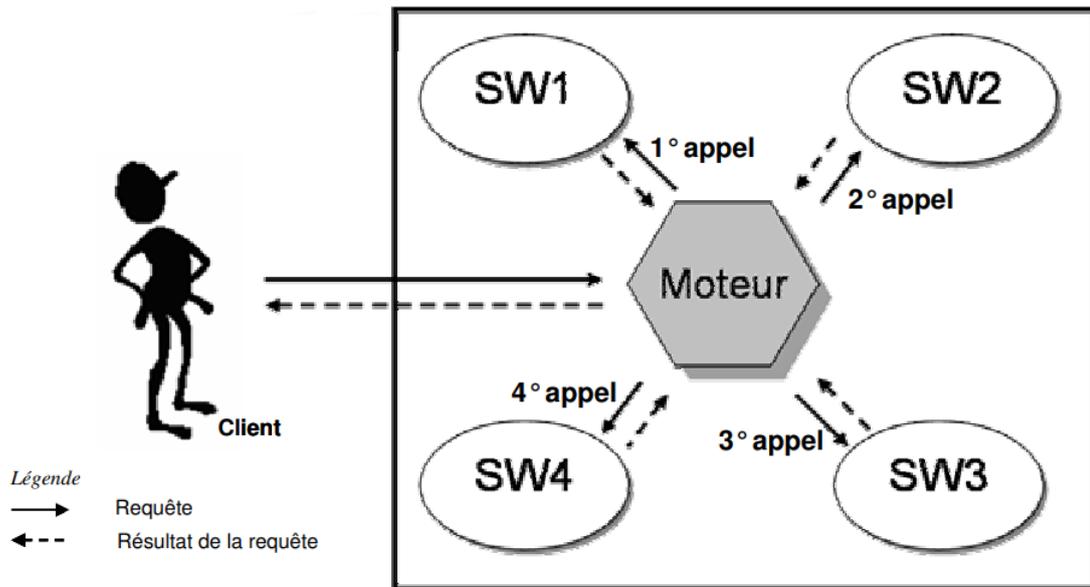


FIGURE 1.16: Vue générale de l'orchestration.

- **Chorégraphie de services Web** : La chorégraphie est aussi appelée composition dynamique. En effet, l'exécution n'est pas régie de manière statique comme dans une composition de type orchestration. Dans une chorégraphie à chaque pas de l'exécution, le service Web choisit le service Web qui lui succède et implémente ainsi une partie de la chorégraphie. La composition de type chorégraphie n'est pas connue, ni décrite à l'avance [19].

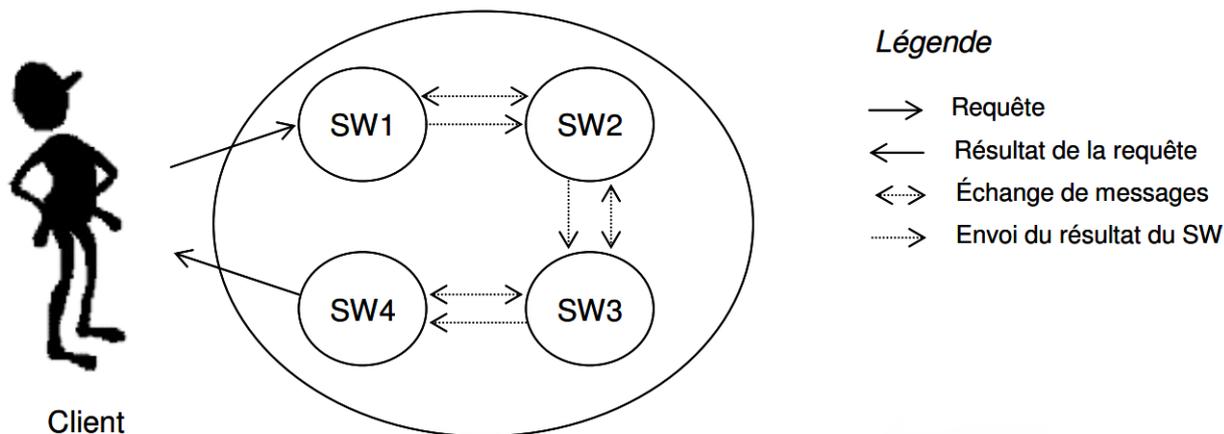


FIGURE 1.17: Vue générale de la chorégraphie.

1.10 Les inconvénients des services web

- Les services web ont de faibles performances par rapport aux autres approches de l'informatique répartie telles que le RMI, CORBA ou DCOM.
- Les transferts reposent sur le XML, ce qui pose un problème sur la taille des fichiers échangés car les fichiers XML sont le plus souvent de très gros fichiers et ceci entraînera une lourdeur considérable.
- En utilisant du protocole HTTP, les services web peuvent contourner les mesures de sécurité mises en place à travers les firewalls.
- La sémantique n'est pas prise en charge de façon efficace car le WSDL décrit les services de manière syntaxique [20].

1.11 Services web sémantiques

1.11.1 Définition

Les services web sémantiques (SWS) se situent à la convergence de deux domaines de recherche importants qui concernent les technologies de l'Internet : le web sémantique et les services web. Le web sémantique s'intéresse principalement aux informations statiques disponibles sur le Web et les moyens de les décrire de manière intelligible pour les machines [21].

Les services web sémantiques étendent les capacités des services web en ajoutant des concepts sémantiques, ces derniers permettent à un agent logiciel d'interpréter les fonctionnalités offertes par un service web et s'il est lui-même capable d'utiliser ce service.

1.11.2 Langages de description des services web sémantiques

L'infrastructure de base autour des standards SOAP, WSDL, UDDI est suffisante pour mettre en place des composants interopérables et intégrables mais insuffisante pour rendre automatique et efficace plusieurs tâches liées au cycle de vie des services web comme la composition et la découverte des services requis. WSDL fournit une description insuffisante pour qu'un agent logiciel puisse interpréter la signification réelle des opérations WSDL [22], pour résoudre ce problème, plusieurs approches ont été développées pour les SWS, parmi eux on cite :

OWL-S

Semantic Markup for Web Services, est un langage permettant de décrire les services Web de façon non ambiguë et interprétable par des programmes. Ce langage est basé sur le langage d'ontologie du Web (OWL) [23]. Il comprend trois parties principales : profil de service, modèle de processus de service et le service grounding comme illustré dans la figure 1.18 :

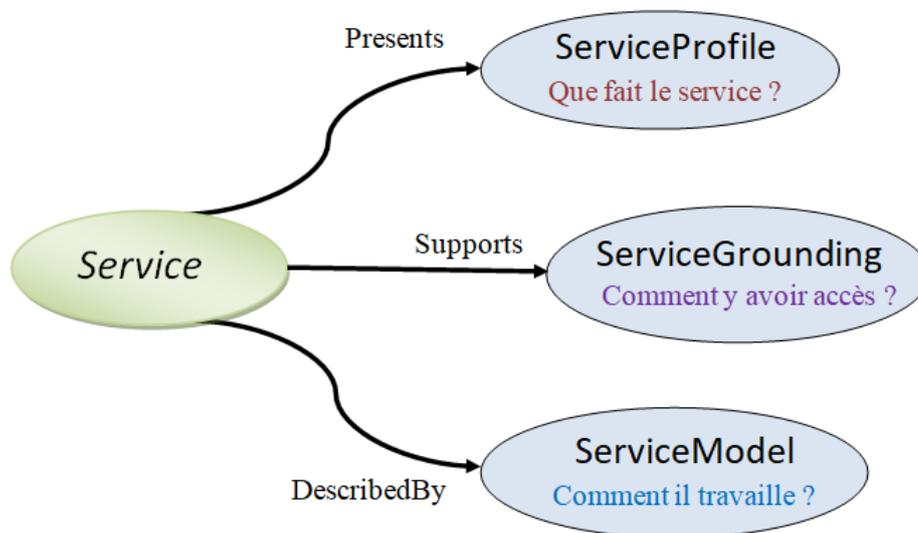


FIGURE 1.18: Ontologie OWL-S.

WSML

Langage de modélisation de service Web (WSML) est un langage pour la spécification de différents aspects des services Web sémantiques. Il fournit un langage formel pour l'ontologie de modélisation de service Web (WSMO) qui est basé sur des formalismes

logiques bien connus, spécifiant un cadre de langage cohérent pour la description sémantique des services Web [24]. Le WSML présente plusieurs variantes telles que : WSML-Core marque le sous-ensemble commun, WSML-DL marque le paradigme de logique de description, WSML-Flight une extension de WSML-Core avec méta-modélisation, WSML-Rule marque le paradigme de règles et WSML-Full marque le sur-ensemble commun.

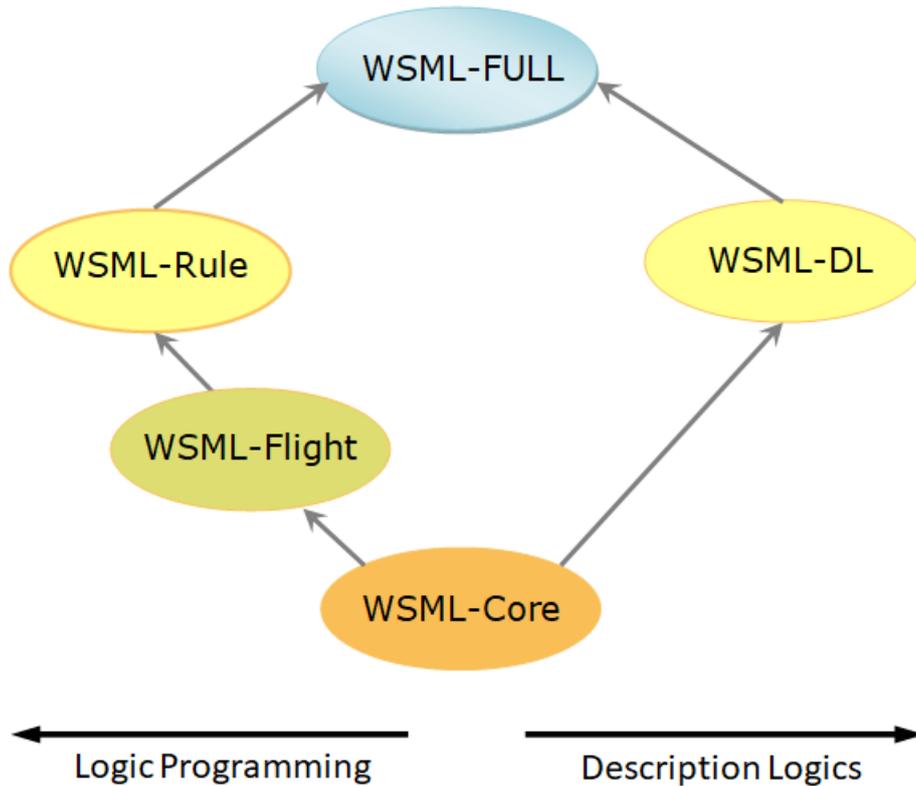


FIGURE 1.19: Les variantes de WSML.

SAWSDL

Semantic Annotations for WSDL and XML Schema, est un langage sémantique de description de service Web. Il est évolutif et compatible avec les standards des services Web existants, et plus spécifiquement avec WSDL, il augmente l'expressivité du langage WSDL avec la sémantique en utilisant des concepts analogues à ceux utilisés dans OWL-S.

SAWSDL fournit un mécanisme permettant d'annoter sémantiquement les types de données, les opérations et les entrées/sorties de WSDL [23].

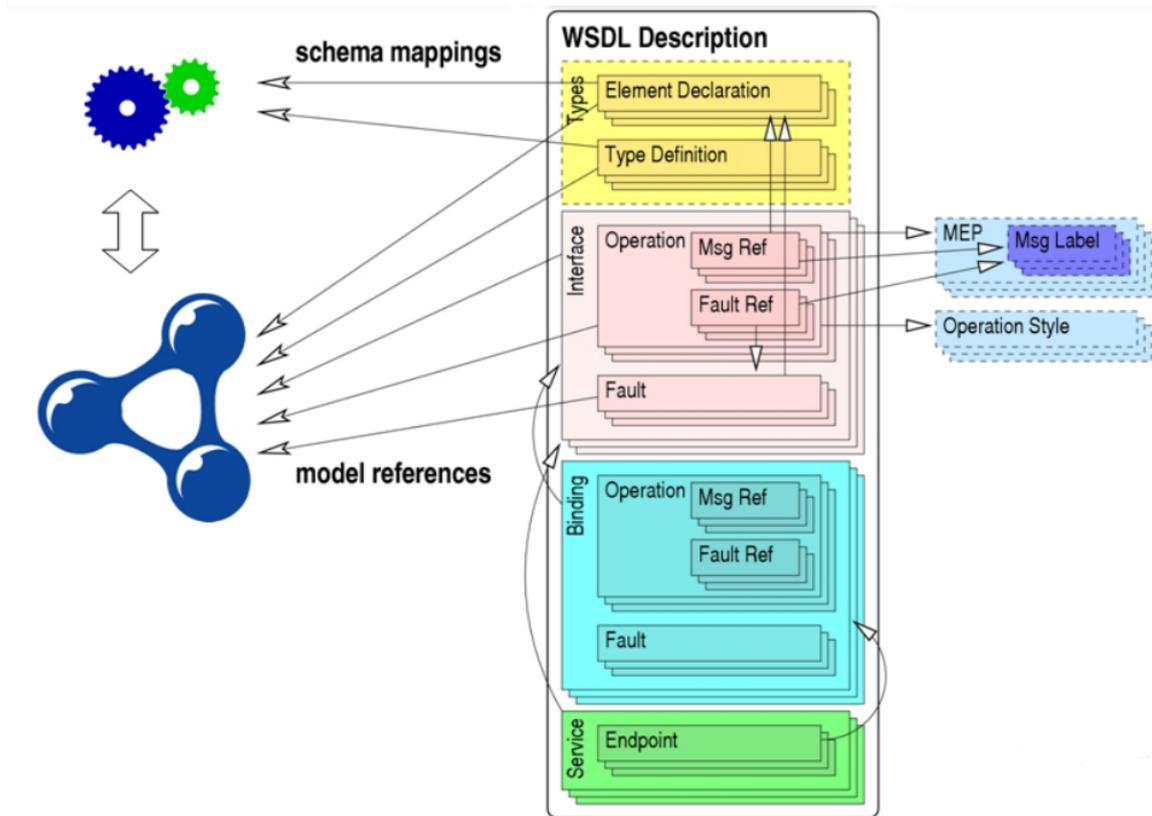


FIGURE 1.20: Architecture de SAWSDL proposée par le W3C.

WSDL-S

Fournit un mécanisme pour associer des annotations sémantiques aux services Web décrits à l'aide du langage WSDL.

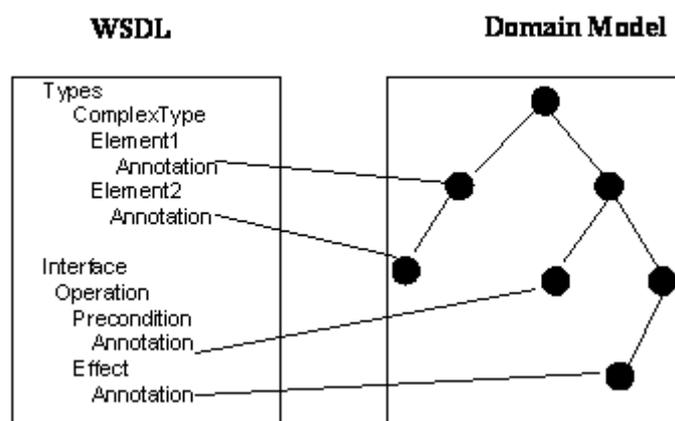


FIGURE 1.21: Annotation sémantique d'une description WSDL.

1.12 Conclusion

Les services web sont basés sur des interfaces standard, ils peuvent communiquer même s'ils fonctionnent sur des systèmes hétérogènes. Pour cette raison, ils constituent une excellente approche pour construire des applications distribuées qui doivent intégrer divers systèmes sur un réseau.

Le prochain chapitre sera consacré aux systèmes de recommandation des services web et les différentes techniques utilisées.

Chapitre II

Les systèmes de recommandation

2.1 Introduction

Vu l'augmentation de la quantité d'informations sur internet, il a été nécessaire de concevoir des mécanismes qui facilitent la recherche et qui permettent aux utilisateurs d'accéder à ce qui les intéresse d'une façon rapide et efficace. Les systèmes de recommandation sont apparus pour résoudre ce problème de surcharge d'information et de choix.

Dans ce chapitre, nous présenterons une vue d'ensemble sur les systèmes de recommandation puis nous détaillons les différentes techniques de filtrages d'information et leur utilité pour guider les utilisateurs dans les grandes masses de données disponibles.

2.2 Historique des systèmes de recommandation

La capacité des ordinateurs pour faire des recommandations à des utilisateurs a été reconnue assez tôt dans l'histoire de l'informatique. Grundy, un système bibliothécaire, était une première étape vers des systèmes de recommandation automatiques. Ce système était assez primitif. Il classait les utilisateurs en "stéréotypes" en se basant sur une courte interview, et utilisait ces stéréotypes pour produire des recommandations de livres. Ce travail constituait une première tentative intéressante dans le domaine des systèmes de recommandation. Cependant, son utilisation est restée très limitée.

Au début des années 1990, le filtrage collaboratif apparaît comme une solution pour faire face à la surcharge d'information. L'année 1992 voit l'apparition du système de recommandation de documents Tapestry, ainsi que la création du laboratoire de recherche GroupLens, qui travaille explicitement sur le problème de la recommandation automatique dans le cadre des forums de news de Usenet. Tapestry avait pour but de recommander à des groupes d'utilisateurs des documents issus des newsgroups susceptibles de les intéresser. L'approche utilisée était de type "plus proches voisins" à partir de l'historique de l'utilisateur. On parle alors de filtrage collaboratif manuel, comme une réponse au besoin d'outils pour le filtrage de l'information énoncé à la même époque. La recommandation résulte d'une action collaborative des utilisateurs qui recommandent à d'autres utilisateurs des documents en leur attribuant des notes d'intérêt selon certains critères. Les systèmes de filtrage collaboratif automatiques apparaissent ensuite. GroupLens utilise cette technique pour identifier les articles de Usenet susceptibles d'être intéressants pour un utilisateur donné. Les utilisateurs doivent seulement attribuer des notes ou effectuer d'autres opérations observables (par exemple, lire un article); le système combine alors ces données avec les notes ou les actions d'autres utilisateurs pour fournir des résultats personnalisés. Avec ces systèmes, les utilisateurs n'ont aucune connaissance directe des

opinions des autres utilisateurs, ni des articles présents dans le système.

Au cours de ces dernières années, les systèmes de recommandation deviennent un sujet d'un intérêt croissant dans les domaines de l'interaction homme-machine, de l'apprentissage automatique ainsi que la recherche d'information. En 1995 apparaissent successivement Ringo, un système de recommandation de musique, basé sur les appréciations des utilisateurs et Bellcore, un système de recommandation de vidéos. La même année, GroupLens crée la société Net Perceptions dont le premier client a été Amazon. De nos jours, les systèmes de recommandation sont devenus des composantes incontournables pour la plupart des sites du e-commerce [25].

2.3 Définition

Les systèmes de recommandation peuvent être définis de plusieurs façons, la définition la plus connue est celle de Robin Burke qui est traduite comme suit :

« Système capable de fournir des recommandations personnalisées ou permettant de guider l'utilisateur vers des ressources intéressantes et utiles au sein d'un espace de données important [26]. ».

2.4 Notions liées aux systèmes de recommandation

Dans tout système de recommandation, il existe deux entités importantes qui sont l'item et l'utilisateur :

- L'item est le terme général utilisé pour dénoter ce que le système recommande aux utilisateurs.
- L'utilisateur est une personne qui utilise et accède au système de recommandation et fait l'enregistrement en saisissant ses informations personnelles pour que le système puisse les exploiter.

2.4.1 Notion de Profil

Le profil d'un objet est un ensemble de caractéristiques permettant de le représenter.

- **Le profil utilisateur** : C'est une description des caractéristiques de l'utilisateur (données démographiques, centres d'intérêts, préférences...). Plusieurs approches pour l'acquisition des informations d'un profil utilisateur existent : manuelles, automatiques et semi automatiques. Parmi les approches automatiques et semi-automatiques, nous distinguons le profiling et les méthodes par stéréotypes. Le profiling consiste à

analyser les activités d'un utilisateur durant ses sessions de travail. L'approche par stéréotype se base sur l'identification de groupes d'utilisateurs pour déterminer les caractéristiques clés de chaque groupe.

- **Le profil item** : C'est la description du contenu d'une ressource par le biais d'un ensemble de mots clés, généralement pondérés. La ressource peut être : document, livre, film, image, page web, etc. [27].

2.4.2 Notion de la communauté

Une communauté est un groupe d'utilisateurs qui se rapprochent relativement à un critère donné. La formation de communautés en système de recommandation peut s'appuyer sur plusieurs critères : Les évaluations attribuées aux items, le contenu des items notés, les intérêts, et les données démographiques. En fonction de chacun de ces critères, la communauté et la position d'un utilisateur dans cette communauté peut varier [27].

2.4.3 Notion du vote

Le vote est la note attribuée à un item i par un utilisateur u . Cette note reflète la satisfaction ou non d'un utilisateur pour un item i . Un vote peut se présenter sous forme :

- **Numérique** : Défini sous une échelle de valeurs discrètes (par exemple 1 à 5 étoiles) ;
- **Ordinal** : L'utilisateur est appelé à choisir parmi une liste de valeurs ordonnées et figées (par exemple : très bien, bien, moyen, mauvais) ;
- **Binaire** : L'utilisateur doit juste mentionner s'il aime ou n'aime pas l'item ;
- **Unaire** : Indique que l'utilisateur a consulté, acheté ou noté l'item [27].

2.5 Objectifs des systèmes de recommandation

Un système de recommandation a pour objectif de fournir à un utilisateur des ressources pertinentes en fonction de ses préférences. Ce dernier voit ainsi réduit son temps de recherche mais reçoit également des suggestions de la part du système auxquelles il n'aurait pas spontanément prêté attention. L'essor du Web et sa popularité ont notamment contribué à la mise en place de tels systèmes comme dans le domaine du e-commerce. Les systèmes de recommandation peuvent être vus initialement comme une réponse donnée aux utilisateurs ayant des difficultés à prendre une décision dans le cadre d'utilisation d'un système de recherche d'informations « classique » [28].

2.6 Classification des systèmes de recommandation

Il est possible de classer les systèmes de recommandation de différentes manières. Parfois plusieurs termes sont utilisés pour désigner une même méthode ou approche. La classification la plus utilisée est une classification selon deux approches : les recommandations basées sur le contenu et le filtrage collaboratif. En plus de ces deux approches, Robin Burke propose de considérer trois autres approches : la recommandation basée sur les données démographiques, la recommandation basée sur la connaissance (knowledge-based) et la recommandation basée sur l'utilité (utility-based). Mais il note que ces trois approches sont des cas particuliers des approches classiques [25].

2.7 Les différentes techniques de recommandation

2.7.1 Le filtrage collaboratif (FC)

Le filtrage collaboratif (FC) est un algorithme de recommandation qui consiste à prédire les articles que des utilisateurs apprécieront dans le futur. Le filtrage collaboratif repose sur l'exploitation de « la sagesse de la foule » (traduction de l'expression anglaise consacrée : « wisdom of the crowd »). L'idée est que pour prédire ce qu'une personne va aimer dans le futur, et qu'il est donc pertinent de lui recommander, l'algorithme n'utilise pas uniquement l'historique de cet utilisateur, mais toutes les informations existantes dans le système concernant les autres utilisateurs. Pour faire simple, ces algorithmes permettent de détecter les utilisateurs qui ont des goûts similaires pour exploiter ces informations à des fins de recommandations [29].

Les systèmes de filtrage collaboratif ont pour principe d'exploiter les « évaluations » que des utilisateurs ont faites de certains documents, afin de recommander ces mêmes documents à d'autres utilisateurs. De manière plus formelle, l'utilité d'un document i pour un utilisateur u , $v(u,i)$ sera calculée en fonction des $v_j(u_j,i)$ qui lui sont similaires [30].

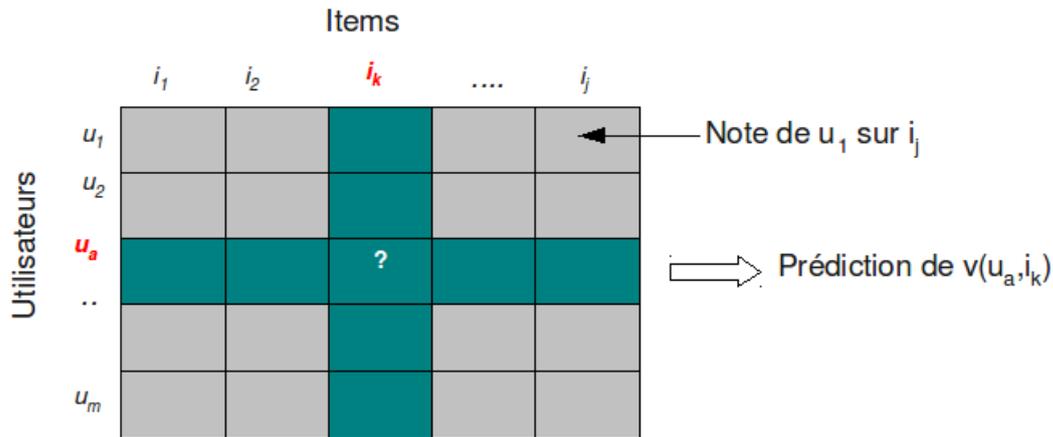


FIGURE 2.1: La matrice Utilisateur/Document.

Il existe deux grandes catégories d’algorithmes du filtrage collaboratif : Les algorithmes basés mémoire et les algorithmes basés modèle :

Filtrage basé mémoire

Les algorithmes basés mémoire, utilisent la totalité ou une partie des profils utilisateurs afin de générer une nouvelle prédiction. Ainsi la note potentielle que donnera un utilisateur à un document est calculée en fonction des notes données par les autres utilisateurs (généralement les N plus proches) au même document [30]. Le processus de calcul de la recommandation pour un usager u_i passe par deux étapes successives [31] :

- **Phase de calcul du voisinage** : En se basant sur le profil de cet usager u_i , le système recherche les usagers u_j (j diffère de i) qui lui sont les plus similaires. Deux mesures de similarité qui sont très utilisées sont : la similarité vectorielle et la corrélation de Pearson [31] :
 - La similarité vectorielle : Dans cette méthode les usagers A et B sont considérés comme deux vecteurs de même origine dans un espace de m dimensions, m est égale au nombre d’items évalués par les deux usagers. Plus deux usagers sont similaires, plus l’angle entre leur vecteur est plus petit. Empiriquement, la similarité entre ces deux usagers est calculée par la formule du cosinus suivante [31] :

$$\text{cosinus}(A, B) = \sum_{j=1}^{l1} \frac{v_{A,j}}{\sqrt{\sum_{j=1}^{l1} v_{A,j}^2}} \frac{v_{B,j}}{\sqrt{\sum_{j=1}^{l1} v_{B,j}^2}} \quad (\text{II.1})$$

$l1$: nombre d'items communs entre A et B votés par v .

$v_{A,j}$: vote de A pour l'item j .

$v_{B,j}$: vote de B pour l'item j .

- La corrélation de Pearson : C'est une méthode issue des statistiques. Elle est aussi très utilisée dans le domaine des systèmes de recommandation pour mesurer la similarité entre deux usagers. La formule ci-dessous, nous donne cette valeur pour deux usagers A et B [31] :

$$w(A, B) = \frac{\sum_j (v_{A,j} - \bar{v}_A)(v_{B,j} - \bar{v}_B)}{\sqrt{\sum_j (v_{A,j} - \bar{v}_A)^2 \sum_j (v_{B,j} - \bar{v}_B)^2}} \quad (\text{II.2})$$

j : nombre d'objets ayant été voté à la fois par A et B .

$v_{A,j}$: vote de A pour l'item j .

\bar{v}_A : moyenne des votes de A .

- **Phase de prédiction** : Une fois que toutes les similarités de l'utilisateur cible A par rapport aux autres usagers sont calculées et que les n usagers les plus similaires qui constituent le voisinage de cet usager cible sont définis, la prédiction de la valeur d'un item j évaluée par l'utilisateur A est calculée à l'aide de la formule suivante [31] :

$$P_{A,j} = \bar{v}_A + \frac{\sum_{i=1}^n w(A, i)(v_{i,j} - \bar{v}_i)}{\sum_{i=1}^n |w(A, i)|} \quad (\text{II.3})$$

n : nombre d'utilisateurs présents dans le voisinage de A , ayant déjà voté sur l'objet j .

$v_{i,j}$: vote de l'utilisateur i pour l'objet j .

\bar{v}_i : moyenne des votes de l'utilisateur i .

Filtrage basé modèle

L'approche basée modèle utilise les informations disponibles sur les utilisateurs et sur les items (les évaluations) pour élaborer un modèle qui génère les recommandations. L'avantage pour ces méthodes par rapport à celles basées sur la mémoire, et qu'elles peuvent être conçues dans la phase hors ligne, en utilisant les données d'apprentissage, pour être utilisées rapidement et facilement en ligne.

Dans le cadre des approches basées modèles, la prédiction peut être faite de deux façons différentes :

- A partir de la prédiction fournie par le modèle lui-même, en construisant par exemple un modèle probabiliste pour l'estimation des valeurs de prédiction ou directement à partir du modèle.
- Ou bien, en regroupant les utilisateurs/ items par les méthodes de clustering et par la suite, les méthodes basées mémoires seront utilisées pour prédire les évaluations pour les items [32].

2.7.2 Le filtrage basé sur le contenu (FBC)

Les systèmes de filtrage à base de contenu recommandent des documents similaires à ceux que l'utilisateur a déjà appréciés. Ceci est calculé en rapprochant les centres d'intérêt des utilisateurs (introduits de manière explicite à travers un questionnaire par exemple ou de manière implicite à travers la surveillance de son comportement) avec la métadonnée ou les caractéristiques des documents, sans prendre en compte les avis des autres utilisateurs [30].

2.7.3 Le filtrage démographique

Ce type de filtrage utilise les données démographiques de l'utilisateur (âge, profession, ville d'origine, etc.) pour lui générer des recommandations. Dans ce cas, le filtrage se base sur une catégorisation des informations en fonction des données démographiques des individus. Ce qui conduit à limiter la performance de ce type du filtrage du fait que ces systèmes ne fonctionnent pas bien que pour un utilisateur avec beaucoup de proches voisins (similaires). De plus, les informations démographiques doivent être accumulées [32].

2.7.4 Le filtrage basé utilité

Ce système ne construit pas une généralisation à long terme pour ses utilisateurs, mais il base sa recommandation sur une évaluation de la correspondance entre les besoins des utilisateurs et l'ensemble des items disponibles. Dans ce type de filtrage, l'utilité de chaque item pour l'utilisateur est calculée afin de recommander les items les plus utiles.

Dans ce cas, le profil de l'utilisateur est la fonction d'utilité que le système dérive pour lui, et le système emploie des techniques de contraintes de satisfaction afin de localiser la meilleure correspondance. Certaines limites sont associées à ce type du filtrage du fait que l'utilisateur doit introduire une fonction d'utilité au système. De plus, ces systèmes n'apprennent pas les préférences des utilisateurs ce qui conduit à une capacité de suggestion statique [32].

2.7.5 Le filtrage basé connaissance

Les systèmes de recommandation basés sur la connaissance font des suggestions en se basant sur des inférences des besoins et des préférences d'un utilisateur. Dans un certain sens, toutes les techniques de recommandation peuvent être décrites comme faisant un certain type d'inférence. Cette classe des systèmes s'inspire de la recherche en raisonnement à base de cas.

Contrairement aux trois premières méthodes qui utilisent des algorithmes d'apprentissage, le filtrage basé connaissance exploite des connaissances du domaine et fait des inférences sur les besoins et préférences des utilisateurs, ce qui rend cette méthode plus appropriée dans certain cas que les autres surtout quand le système souffre d'un manque d'informations sur les utilisateurs ou sur les items (le problème de démarrage à froid) du fait qu'elle n'est pas sensible au manque d'informations.

Cependant, les approches dans ces systèmes se distinguent par le fait d'avoir une connais-

sance fonctionnelle, i.e. elles possèdent une connaissance décrivant comment un item particulier satisfait le besoin d'un utilisateur. Par exemple, l'utilisateur peut recommander tout simplement un thé, et le système va connaître que le thé fait partie de la cuisine asiatique [32].

2.7.6 Le filtrage hybride

Un système de recommandation est dit hybride quand il combine deux ou plusieurs approches de recommandation différentes. La recommandation basée sur le contenu et la recommandation collaborative ont souvent été considérées comme complémentaires. L'hybridation de ces deux techniques, afin de traiter les insuffisances de chaque technique utilisée seule et profiter de leurs points forts, a fait l'objet de plusieurs travaux de recherche. Le système FAB est un des premiers systèmes de recommandation hybrides. Il combine le filtrage collaboratif et une approche basée sur le contenu afin de traiter à la fois le problème du démarrage à froid pour les items et la sur-spécialisation. Dans ce système, deux critères doivent être satisfaits pour recommander un item : son contenu doit être similaire au profil de l'utilisateur, et il doit être apprécié par les voisins les plus proches.

Il existe plusieurs manières de faire de l'hybridation et aucun consensus n'a été défini par la communauté des chercheurs. Toutefois, Burke a identifié sept manières différentes de faire l'hybridation :

- **Pondérée (Weighted)** : Le score ou la prédiction obtenu par chacune des deux techniques est combiné en un seul résultat.
- **Par sélection (Switching)** : Le système bascule entre les deux techniques de recommandation en fonction de la situation.
- **Mixte (Mixed)** : Les listes des recommandations issues des deux techniques sont fusionnées en une seule liste.
- **Par combinaison des propriétés (Feature combination)** : Les données issues des deux techniques sont combinées et transmises à un seul algorithme de recommandation.
- **Par augmentation de propriétés (Feature augmentation)** : Le résultat d'une technique est utilisé comme entrée de l'autre technique.
- **En cascade** : Dans ce type d'hybridation, une technique de recommandation est utilisée pour produire un premier classement des items candidats et une deuxième technique affine ensuite la liste des recommandations.

- **En définissant un niveau méta** : Cette méthode est analogue à la méthode par augmentation de propriétés mais c'est le modèle appris qui est utilisé en entrée de la deuxième technique et non la liste résultat des recommandations [25].

2.8 Avantages et inconvénients des systèmes de recommandation

- **Adaptabilité** : Au fur et à mesure que la base de données des évaluations augmente, la recommandation devient plus précise.
- **Nouvel utilisateur** : Un nouvel utilisateur qui n'a pas encore accumulé suffisamment d'évaluations ne peut pas avoir de recommandations pertinentes.
- **Nouvel item** : Un item doit avoir suffisamment d'évaluations pour qu'il soit pris en considération dans le processus de recommandation.
- **Démarrage à froid** : Souvent, on se retrouve confronté au problème qu'un utilisateur ne soit comparable avec aucun autre. Ce problème est dû au fait que peu ou pas d'utilisateurs ont évalué un article donné, ou qu'un utilisateur donné a évalué très peu ou pas d'articles. Généralement, ce problème survient quand un nouvel utilisateur ou une nouvelle ressource est ajouté à la base de recommandation.
- **Le cas du système débutant** : provient lors du lancement d'un nouveau service de recommandation. Le système ne possède alors aucune information sur les utilisateurs et sur les items. Les méthodes de filtrage collaboratif ne peuvent pas fonctionner sur une matrice d'usages vide. La solution consiste en général à trouver des informations descriptives des items afin d'organiser le catalogue et inciter les utilisateurs à le parcourir jusqu'à ce que la matrice d'usages soit assez remplie et permette de passer en mode collaboratif [33].

Le tableau ci-dessous résume les avantages et inconvénients des méthodes traditionnelles utilisées par les systèmes de recommandation [33] :

Approches	Avantages	Inconvénients
Filtrage à base de contenu	<ul style="list-style-type: none"> • Pas besoin d'une large communauté d'utilisateurs pour effectuer des recommandations. • Une liste de recommandations peut être générée même s'il n'y qu'un seul utilisateur. • La qualité croît avec le temps. • Pas besoin d'information sur les autres utilisateurs. • Prendre en considération les goûts uniques des utilisateurs. 	<ul style="list-style-type: none"> • L'analyse du contenu est nécessaire pour faire une recommandation. • Problème de recommandation des images et de vidéos en absence de Métadonnées. • Nécessité du profil d'utilisateur.
Filtrage collaboratif	<ul style="list-style-type: none"> • Ne demande aucune connaissance sur le contenu de l'item ni sa sémantique. • La qualité de la recommandation peut être évaluée. • Plus le nombre d'utilisateurs est grand plus la recommandation est meilleure. 	<ul style="list-style-type: none"> • Démarrage à froid. • Nouvel Item. • Nouvel utilisateur. • Problème de confidentialité. • La complexité : dans les systèmes avec un grand nombre d'items et d'utilisateurs le calcul croît linéairement.
Filtrage basé connaissance	<ul style="list-style-type: none"> • Elle ne souffre pas du problème de démarrage à froid avec un nouvel utilisateur ou un nouvel item. • Sensibilité aux changements des préférences. • Possibilité de relier les besoins des utilisateurs et les items [32]. 	<ul style="list-style-type: none"> • Capacité de suggestion statique. • Une gestion des connaissances est requise [32].

TABLE 2.1: Avantages et inconvénients des techniques de recommandation.

2.9 Systèmes de recommandation sémantiques

La plupart des techniques de filtrage actuelles ne prennent pas en compte un bon nombre d'informations utiles pour expliquer un jugement qu'effectue un utilisateur. Ceci requière la prise en compte de différentes sources d'informations sur utilisateur (ses préférences précédemment annoncées, son entourage social) et leurs relations avec le contenu des documents qui lui sont proposés. Par exemple, dans les systèmes purement collaboratifs, l'utilisateur et le document sont représentés par un ensemble de votes. Par conséquent, le groupement de ces utilisateurs dans des communautés traduit une similarité globale souvent insuffisante pour refléter les relations pouvant lier ces utilisateurs et encore plus les documents [30].

2.9.1 Systèmes de filtrage d'information à base d'ontologies

L'utilisation de ressources conceptuelles pour indexer les documents a fait émerger de nouvelles approches tentant d'intégrer la sémantique dans la recherche d'informations. Ces méthodes ont inspiré le passage d'une description des documents par mots-clés ou à base d'attributs à une description sémantique à base de concepts. Dans les systèmes de filtrage d'informations, ces ressources sémantiques sont surtout utilisées pour classifier les documents et inférer les profils utilisateurs [30].

2.9.2 Systèmes d'annotations collaboratives

Les systèmes d'annotations collaboratives sont un autre aspect très prometteur du web sémantique. Ces systèmes offrent à leurs utilisateurs la possibilité d'héberger leurs photos, vidéos, documents ou toute autre ressource et surtout leur assigner un ensemble de mots décrivant leurs contenus : c'est ce qu'on appelle les annotations (tags).

Ces annotations décrivent le contenu de la ressource ou bien fournit une information contextuelle et sémantique en plus. Parmi les services web les plus populaires on retrouve Flickr et del.icio.us. En général, les annotations sont souvent associées au web sémantique du fait qu'ils permettent aux utilisateurs d'ajouter une métadonnée aux documents d'une manière très simple. Cette connaissance en plus d'offrir les annotations sur les documents peut être exploitée dans le processus de filtrage d'informations.

D'après [34], La matrice des votes est étendue avec les annotations comme troisième dimension. Cette matrice 3-dimensionnel est ensuite décomposée en trois sous matrices : « utilisateurs, documents », « utilisateurs, annotations » et « documents, annotations ».

Pour le calcul de la prédiction, les auteurs utilisent la combinaison de la similarité entre utilisateurs et celle entre documents mais cette fois sur la matrice étendue. Les résultats ont montré que l'introduction des annotations a été surtout bénéfique pour la combinaison des deux similarités ce qui n'était pas du tout le cas si on l'appliquait à chaque méthode a part (similarité utilisateur/utilisateur et similarité document/document) [30].

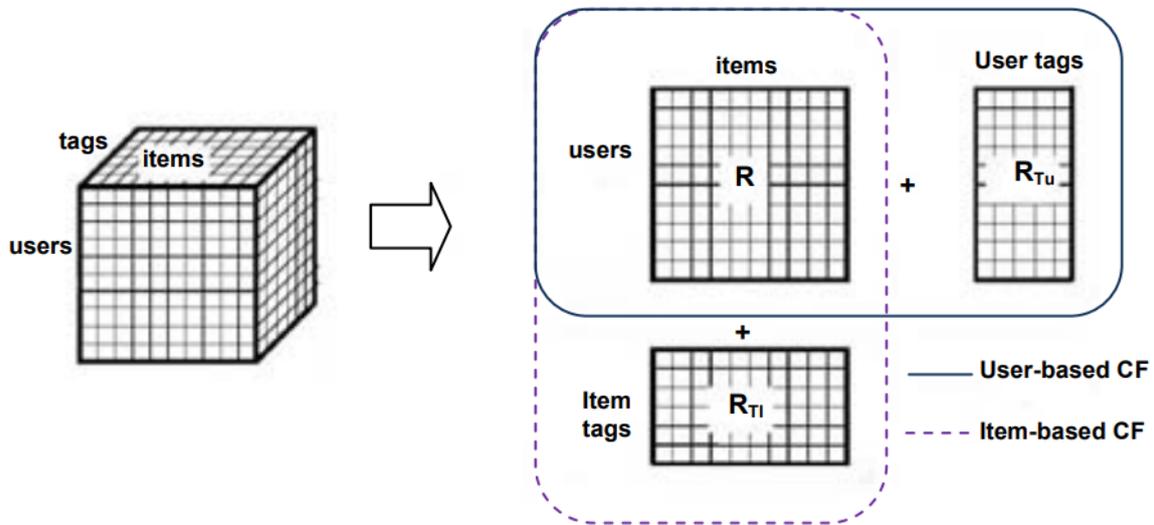


FIGURE 2.2: Extension de la matrice utilisateur/document par les annotations.

2.10 Conclusion

Dans ce chapitre, nous avons présenté les systèmes de recommandation qui sont devenus omniprésents dans de nombreux domaines, ainsi que les différentes approches de filtrages qui aident les utilisateurs à trouver les ressources qui sont adaptées à leurs préférences parmi la grande masse d'informations et de choix qui s'offrent à eux.

Nous présenterons dans le chapitre suivant un état de l'art des articles les plus récents dans le domaine des systèmes de recommandation ainsi que les services web.

Chapitre III

Etat de l'art : les systèmes de recommandation de services web

3.1 Introduction

Dans ce chapitre, nous présentons les différents travaux de la littérature utilisant les principales techniques de recommandation de services web : filtrage basé sur le contenu, filtrage collaboratif et le filtrage hybride.

3.2 Les systèmes de recommandation de services web

Blake et al. [35] proposent une approche de recommandation de services Web à base de contenu aux utilisateurs d'une façon proactive (sans que les utilisateurs demandent). Les auteurs appliquent une méthode d'appariement syntaxique vu que l'annotation sémantique n'est pas disponible pour tous les services Web. Le système proposé acquiert les préférences utilisateur implicitement en observant son comportement. Il détermine les mots clés marquant le plus les messages de l'utilisateur. Ces mots clés sont appariés avec les descriptions de services pour identifier les services à recommander à l'utilisateur.

Cependant, l'approche syntaxique présente des limites telles que l'imprécision dans la comparaison des mots clés et l'incomplétude de la description des services qui peuvent apparaître dans le mécanisme d'évaluation de la similarité. Pour remédier à ces limites, des approches basées sur des mesures sémantiques combinées aux mesures syntaxiques ont été proposées.

En effet, les approches sémantiques utilisent des ontologies et des corpus pour prendre en considération la sémantique lors du processus de comparaison. Par exemple, l'approche proposée dans [36] se base sur l'appariement logique entre les paramètres de services (à savoir les entrées et les sorties) pour créer des relations de causalité entre les services. Si les sorties d'un service s_1 correspondent aux entrées de services s_2 , une relation de causalité est créée entre les deux. Un appariement sémantique entre la requête utilisateur et les services est effectué pour la génération automatique des services composés afin de satisfaire le besoin de l'utilisateur.

Dans l'article [37], les auteurs ont proposé une approche de recommandation de services web basée sur une ontologie décrite en OWL-S et un appariement sémantique. L'idée de base de cette approche est de calculer la similarité entre toutes les entrées/sorties du service web avec celles exprimées dans la requête en utilisant une ontologie de domaine. Le degré de similarité dépend des relations ontologiques entre les concepts des entrées/sorties. Ensuite les services Web sont classés en fonction de leurs degrés d'appa-

riement (Exact, Plug-in, Subsume, Sibling et échec) avec la requête.

En cas d'échec total (aucun service n'est retourné), les auteurs proposent l'utilisation d'une mise en correspondance structurelle qui prend en considération tous les paramètres de la requête ainsi que du service web (nom de service, commentaires). Cette mise en correspondance structurelle consiste à l'utilisation des relations de voisinage structural dans l'ontologie de domaine, ce qui permet d'améliorer les chances d'avoir de meilleurs services [38].

Zheng et al. [39] proposent un système de recommandation de services Web (WSRec) pour résoudre le problème de sélection de services Web. WSRec inclut un mécanisme de contribution utilisateur pour la collection d'information de QoS de service Web et un nouvel algorithme de filtrage collaboratif hybride, pour la prévision de la valeur de QoS des services Web, en employant l'historique de QoS d'autres utilisateurs similaires de services.

Dans l'article [40] les auteurs ont proposé un système de recommandation de service Web basé sur le filtrage collaboratif, afin d'aider les utilisateurs à choisir les services Web avec une Qualité de Service optimale. Basé sur les valeurs de QoS prédites des services Web, des recommandations personnalisées basées sur les QoS de service Web peuvent être produites pour aider des utilisateurs à choisir le service Web optimal parmi ceux qui lui sont fonctionnellement équivalent.

Pour augmenter l'exactitude de prévision, ils ont proposé un système de recommandation de service Web basé localisation (appelé LoRec), qui utilise des valeurs de QoS de service Web et la localisation d'utilisateur pour fournir des prédictions personnalisées de QoS.

Dans [41] les auteurs tiennent compte de l'influence personnalisée des services en calculant la mesure de similarité entre les utilisateurs et l'influence personnalisée des services. Basé sur le modèle de mesure de similarité des services Web, ils ont développé une technique (PHCF) de filtrage collaboratif hybride personnalisée efficace en intégrant l'algorithme basé utilisateur personnalisé et l'algorithme basé item personnalisé.

Deng et al. [42] ont utilisé l'analyse des réseaux des utilisateurs et des services web pour étudier les rapports de confiance entre les utilisateurs et les services web, ils proposent un algorithme de filtrage collaboratif nommé « Trust-based Service Recommendation (TSR) » qui fournit aux utilisateurs des recommandations personnalisées de services Web basées

sur leur confiance et sur la QoS désirée [43].

Plusieurs approches de recommandation de services Web hybrides ont été proposées dans la littérature. Dans [44], les auteurs proposent un système de découverte de services Web combinant les techniques d'appariement sémantique avec un système de filtrage collaboratif. Dans ce système, les fournisseurs de services exposent la description WSDL du service proposé ainsi qu'une description sémantique des profils et contrat de QoS du service écrit en DAMLS.

Un processus de découverte de services dans ce système est initié par la réception d'une requête de la part d'un demandeur de services. Cette requête exprime son besoin fonctionnel et éventuellement ses besoins de QoS. Un appariement sémantique des entrées/sorties est fait entre la requête et les descriptions de services. Cette étape fournit une liste de services répondant aux besoins exprimés par la requête. Cette liste est transmise à un système de filtrage qui la trie sur la base des évaluations d'utilisateurs précédents et la transmet au demandeur de services. Quand un demandeur de services exécute un service, le système de recommandation lui demande de l'évaluer afin de proposer des recommandations futures.

Dans [45], les auteurs exploitent les informations de région (la localisation des utilisateurs) pour la recommandation de services Web. L'approche proposée utilise les caractéristiques de QoS et regroupe les utilisateurs selon leurs localisations. Elle exploite les historiques des voisins les plus proches pour prédire les évaluations de l'utilisateur courant. Les recommandations finales sont visualisées sur une carte par l'utilisateur. Dans [46], les auteurs proposent une approche hybride qui recommande dynamiquement des services Web selon les intérêts des utilisateurs. Le filtrage collaboratif exploite les données d'évaluations de QoS pour trouver les utilisateurs similaires et identifier la liste de services à recommander. Cette liste est filtrée selon les préférences de l'utilisateur [38].

Yao et al. [47] proposent une nouvelle approche qui unifie le filtrage collaboratif et les recommandations basées sur le contenu. En particulier, cette approche considère simultanément des données d'estimation et des données sémantiques des services Web en utilisant un modèle génératif probabiliste. Dans ce modèle, les préférences inobservables d'utilisateur sont représentées par un ensemble de variables latentes, qui peuvent être statistiquement estimées, en utilisant des algorithmes comme maximisation d'espérance (EM) [43].

3.3 Tableau comparatif

Le tableau suivant compare les approches qui traitent les systèmes de recommandation des services web selon plusieurs critères d'évaluations : L'objectif de l'approche, l'algorithme et les techniques de recommandations utilisés, la sémantique et la technique utilisée pour la sémantique (annotation ou l'ontologie) :

Approche	Objectif	Algorithme	Technique de recommandation	Sémantique	Technique utilisée pour la sémantique
[35]	Recommandation	Filtrage basé contenu	Appariement syntaxique	-	-
[36]	Composition	Filtrage basé contenu	Appariement sémantique	+	Annotation
[37]	Recommandation	Filtrage basé contenu	basée sur une ontologie OWL-S et un appariement sémantique	+	Ontologie et annotation
[39]	Sélection	Filtrage collaboratif	Hybride	-	-
[40]	Sélection	Filtrage collaboratif	Basée service web	-	-
[41]	Recommandation	Filtrage collaboratif	Hybride	-	-
[42]	Recommandation	Filtrage collaboratif	Hybride	-	-
[44]	Sélection	Filtrage collaboratif et basé contenu	Hybride	+	Annotation
[45]	Recommandation	Filtrage collaboratif et basé contenu	Hybride	-	-
[46]	Recommandation	Filtrage collaboratif et basé contenu	Hybride	+	Annotation
[47]	Sélection	Filtrage collaboratif et basé contenu	Hybride	+	Annotation

TABLE 3.1: Les approches qui traitent les systèmes de recommandation de services Web.

3.4 Synthèse

L'avantage des systèmes basés contenu est qu'ils permettent d'associer des items à un profil utilisateur. L'utilisateur est indépendant des autres ce qui lui permet d'avoir des recommandations même s'il est le seul utilisateur du système [38]. De nombreux travaux de recherche [44] [45] [46] [47] ont utilisé le filtrage hybride (filtrage collaboratif et basé contenu) afin de traiter les insuffisances de chaque technique utilisée seule.

Les systèmes de filtrage collaboratif sont les plus utilisés dans les travaux existants car ils fournissent des recommandations considérées comme pertinentes. Cependant, le problème majeur de ces approches est l'absence de données pour les nouveaux utilisateurs [38]. Dans notre travail, nous étendons les méthodes de filtrage collaboratif en considérant la sémantique des services web, notre approche peut traiter efficacement les limites de ces systèmes de filtrage collaboratif.

3.5 Conclusion

Dans ce chapitre, nous avons présenté un état de l'art sur les systèmes de recommandation de services web dans le quel nous avons cité quelques travaux de recherche utilisant les différentes techniques de recommandation.

Le chapitre suivant sera consacré pour l'étude et le développement de notre l'application à savoir les outils utilisés, les bases de données, les programmes etc.

Chapitre IV

Systeme collaboratif sémantique pour la sélection et la recommandation de services Web

4.1 Introduction

Dans ce chapitre, nous allons décrire les axes principaux de la réalisation de notre application logicielle qui consiste à proposer un système collaboratif sémantique pour la sélection et la recommandation de services web.

Pour atteindre notre objectif, nous avons utilisé la technique de filtrage hybride : collaboratif basé sur la mémoire et filtrage sémantique, ce dernier se base sur le contenu des services web sémantiques afin de recommander à un utilisateur des services similaires à ceux qu'il a précédemment apprécié, et pour avoir une base de données de services web diversifiée, nous avons rajouté la sémantique des services en utilisant le langage ontologique OWL-S.

4.2 Architecture de l'approche

L'architecture globale de notre système de recommandation est comme suit :

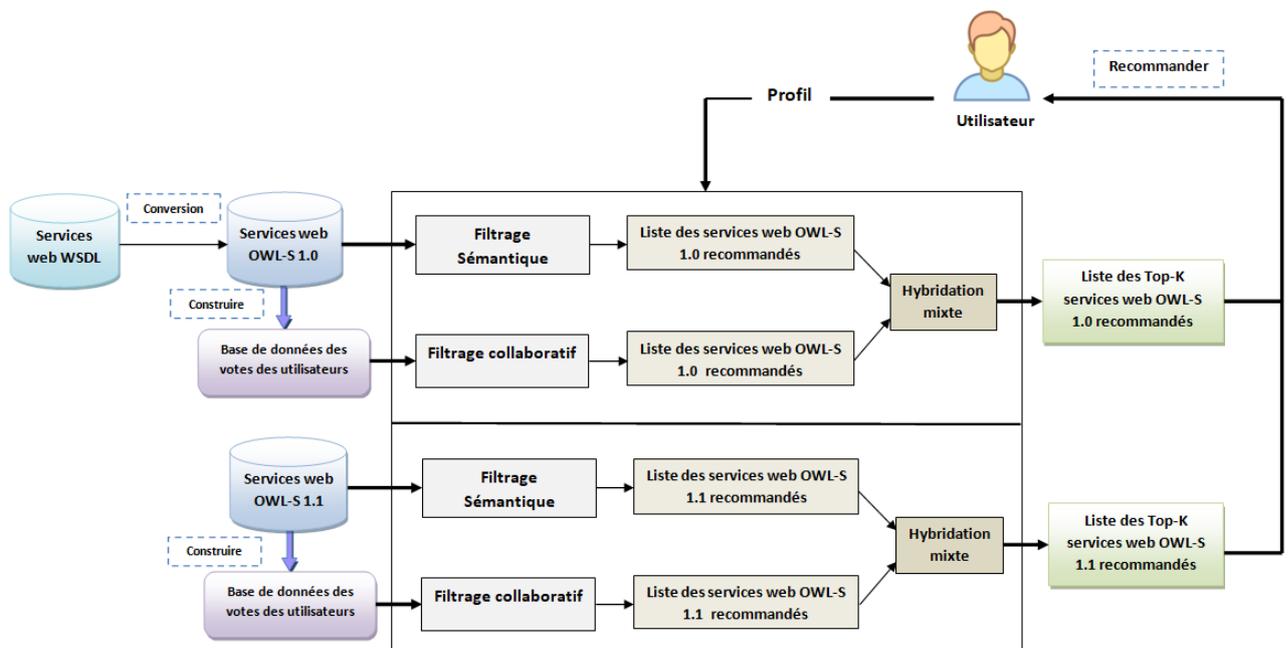


FIGURE 4.1: L'architecture de l'approche proposée.

Pour avoir une liste des services web recommandés, l'utilisateur doit sélectionner un service parmi ceux qu'il a précédemment utilisé (profil utilisateur), et puisque ce système se base sur une technique de filtrage hybride (collaboratif sémantique), la génération de la liste top-k des services les plus pertinents se fait par les étapes suivantes :

4.2.1 Filtrage collaboratif basé item

Il consiste à calculer la valeur de similarité entre le service sélectionné par l'utilisateur et les autres services en se basant sur les votes des utilisateurs (Rating). La méthode que nous avons utilisée c'est la corrélation de Pearson :

$$Sim(a, b) = \frac{\sum_u (v_{u,a} - \bar{v}_u)(v_{u,b} - \bar{v}_u)}{\sqrt{\sum_u (v_{u,a} - \bar{v}_u)^2} \sqrt{\sum_u (v_{u,b} - \bar{v}_u)^2}} \quad (IV.1)$$

u : Représente tous les utilisateurs communs pour les services web a et b .

$v_{u,a}$: Vote de l'utilisateur u pour le service web a .

$v_{u,b}$: Vote de l'utilisateur u pour le service web b .

\bar{v}_u : Moyenne des votes de l'utilisateur u .

La figure 4.2 montre le processus de recommandation basé sur le filtrage collaboratif précédemment expliqué :

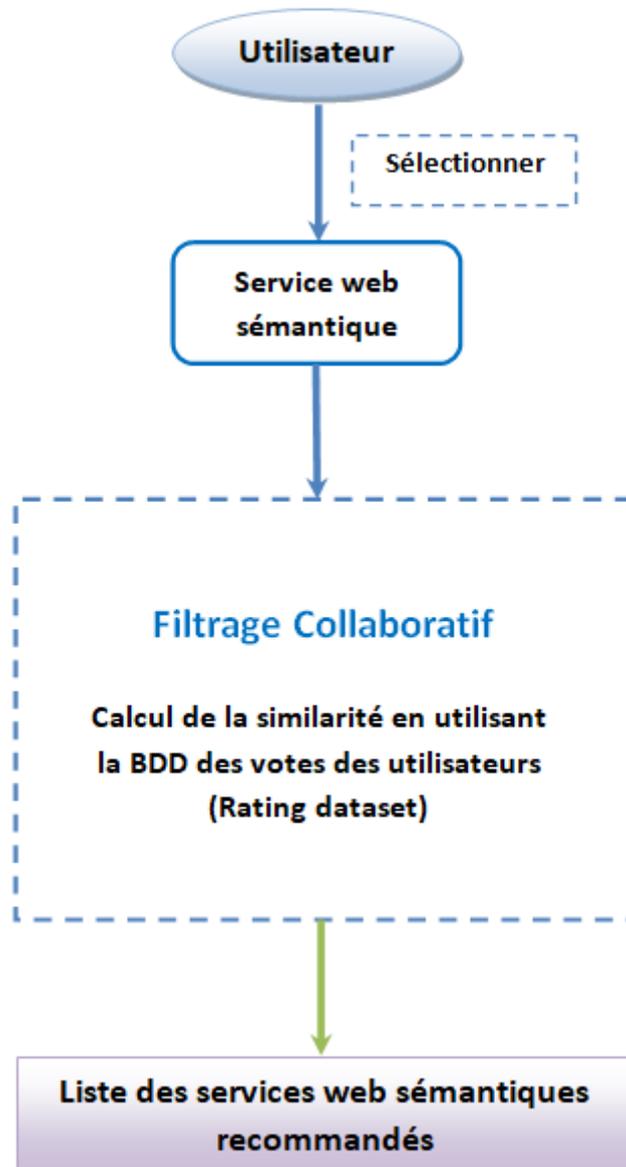


FIGURE 4.2: Le processus de recommandation basé sur le filtrage collaboratif.

4.2.2 Filtrage sémantique

Contrairement au filtrage collaboratif, le filtrage sémantique consiste à recommander des services web sémantiquement similaire au service sélectionné par l'utilisateur, car il se base sur le contenu des services web et non pas sur les votes des utilisateurs.

Pour calculer la valeur de la similarité sémantique nous avons utilisé la similarité cosinus entre les vecteurs des fréquences des mots extraits de la classe ServiceProfile du service web sélectionné par l'utilisateur et des autres services web. La formule cosinus suivante retourne une valeur de 1 lorsque les deux vecteurs sont identiques et la valeur de 0 lorsque les deux vecteurs sont orthogonaux. En d'autres termes, la similarité cosinus est un nombre borné entre 0 et 1 qui nous indique à quel point les deux vecteurs sont similaires :

$$Similarity = \cos(\theta) = \frac{A.B}{\|A\|\|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (IV.2)$$

A_i : La fréquence du mot i dans le vecteur des mots de la description du 1^{er} service OWL-S.

B_i : La fréquence du mot i dans le vecteur des mots de la description du 2^{ème} service OWL-S.

Nous avons choisi la classe ServiceProfile car elle fournit une description complète du service web (le nom du service, la description textuelle, la description fonctionnelle etc.). La figure 4.3 montre le processus de recommandation basé sur le filtrage sémantique précédemment expliqué :

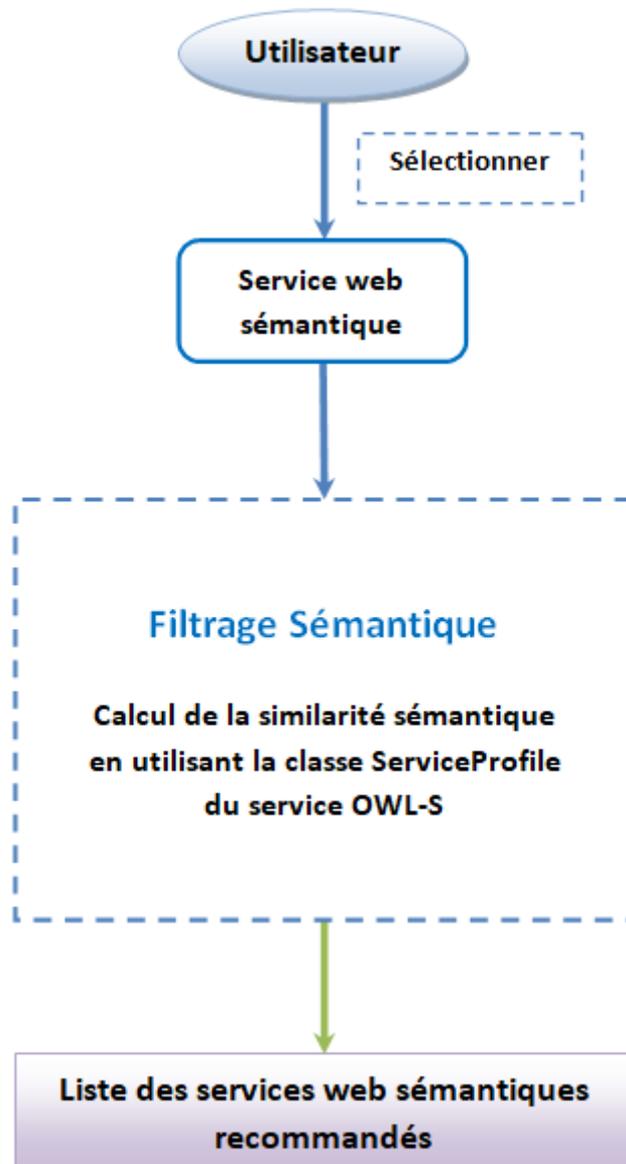


FIGURE 4.3: Le processus de recommandation basé sur le filtrage sémantique.

4.2.3 Hybridation

Dans cette étape, nous avons utilisé la technique d'hybridation mixte qui fusionne les deux listes des services web recommandés par les deux techniques (dont nous avons parlé précédemment) en une seule liste des top-k services web les plus pertinents, classés par ordre décroissant selon le score de la similarité, ce processus est démontré dans la figure suivante :

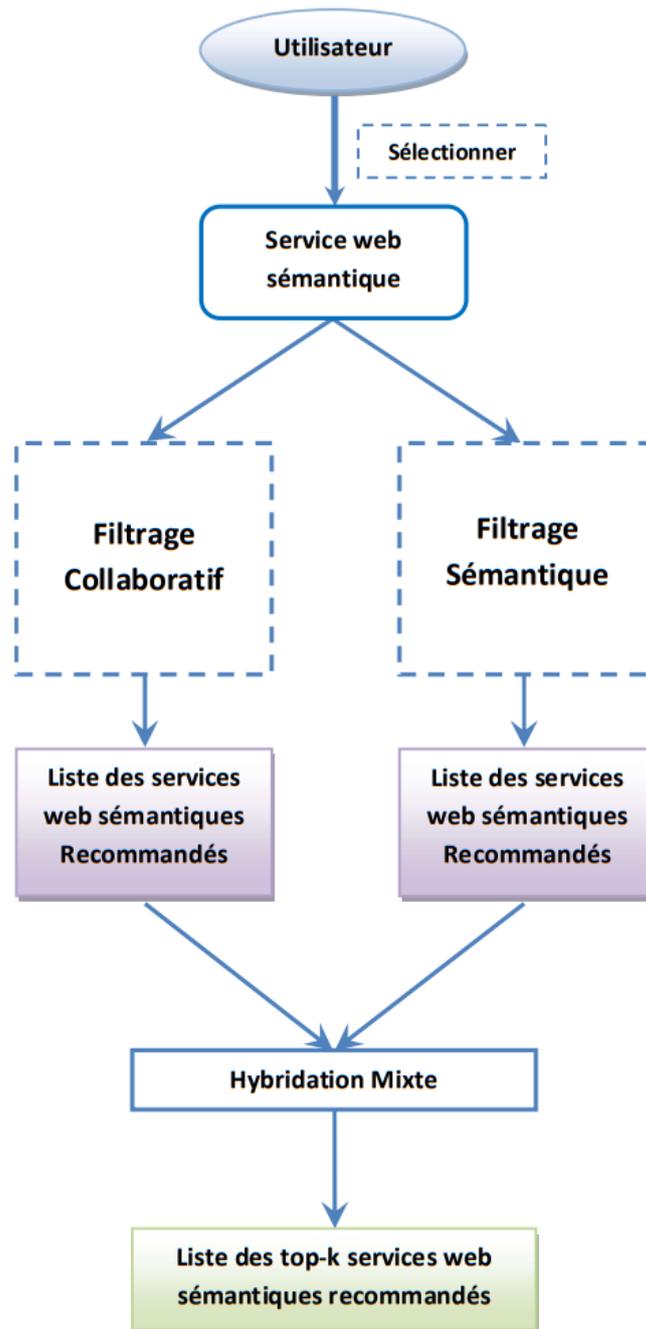


FIGURE 4.4: Le processus de recommandation basé sur le filtrage collaboratif sémantique.

4.3 Conception

Le langage de modélisation que nous avons utilisé est Unified Modeling Language (UML). La fonction d'UML permet de spécifier, construire, visualiser, et documenter un système informatique. Pour modéliser notre application nous avons utilisé deux diagrammes : le diagramme de séquence et le diagramme de classes.

4.3.1 Diagramme de séquence

Le diagramme de séquence (diagramme d'événement) illustre la façon dont des processus interagissent entre eux en montrant les appels entre différents objets dans une séquence [48].

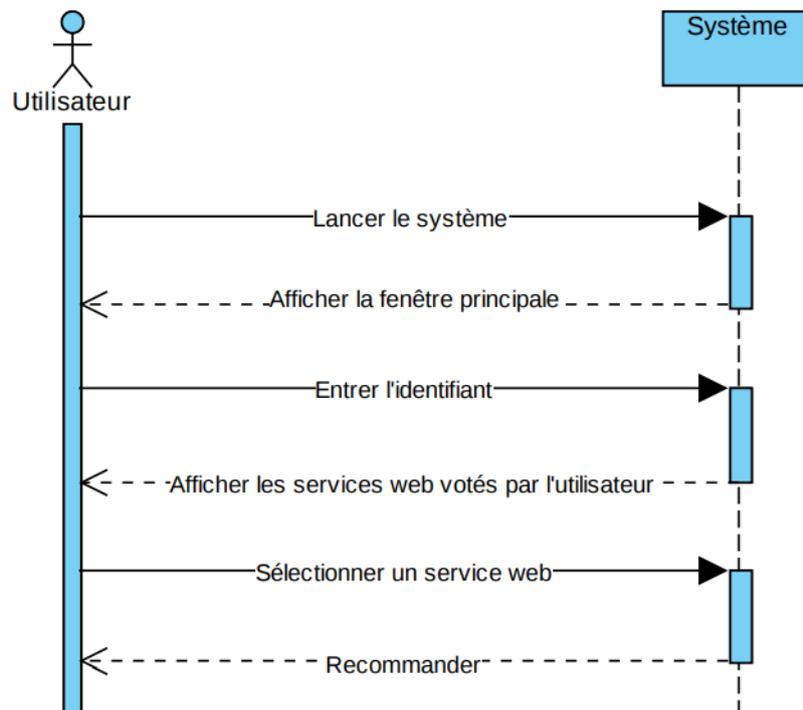


FIGURE 4.5: Diagramme de séquence.

4.3.2 Diagramme de classes

Les diagrammes de classes représentent les structures statiques d'un système, y compris ses classes, attributs, opérations et objets. Un diagramme de classes peut afficher des données statistiques ou des données d'organisation, respectivement sous la forme de classes d'implémentation et de classes logiques [48].

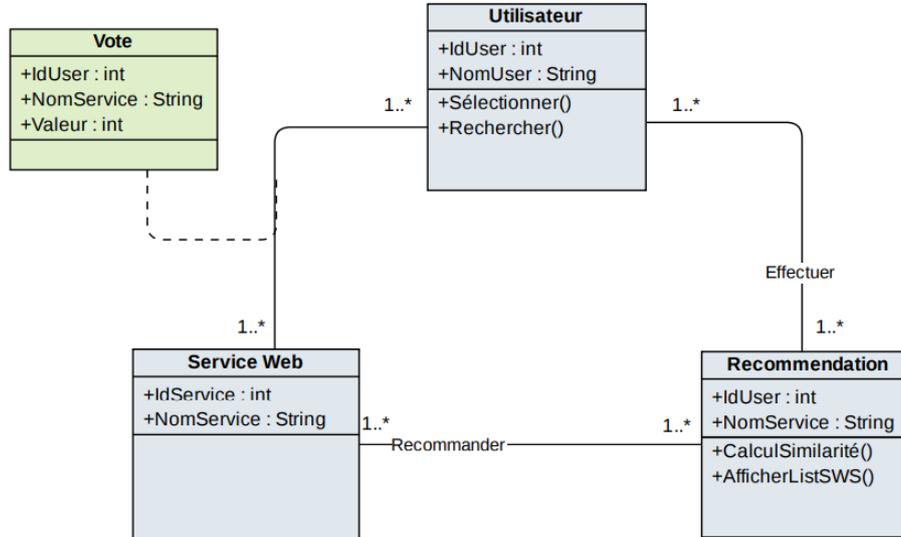


FIGURE 4.6: Diagramme de classes.

4.4 Outils d'implémentation

En plus des langages XML et OWL-S dont nous avons parlé précédemment, nous avons utilisé plusieurs outils et langages pour l'implémentation de l'application tel que java, Netbeans, jDom.

4.4.1 Java

Le langage Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton employés de Sun Microsystems avec le soutien de Bill Joy. Présenté officiellement le 23 mai 1995 au SunWorld.

Le langage Java a la particularité principale que les logiciels écrits avec ce dernier sont très facilement portables sur plusieurs systèmes d'exploitation tels que UNIX, Windows, Mac OS ou GNU/Linux avec peu ou pas de modifications. Le langage reprend en grande partie la syntaxe du langage C++, très utilisé par les informaticiens. Néanmoins, Java a été épuré des concepts les plus subtils du C++ et à la fois les plus déroutants, tels que les pointeurs et références, et l'héritage multiple remplacé par l'implémentation des interfaces. Les concepteurs ont privilégié l'approche orientée objet de sorte qu'en Java, tout est objet à l'exception des types primitifs.

Java a donné naissance à un système d'exploitation (JavaOS), à un environnement de développement (eclipse/JDK), des machines virtuelles (MSJVM, JRE) applicatives multi plates-formes (JVM), une bibliothèque Java (J2ME) avec interface graphique (AWT/Swing),

des applications Java (logiciels, servlet, applet). La portabilité du code Java est assurée par la machine virtuelle. JRE la machine virtuelle qui effectue la traduction et l'exécution du bytecode en code natif supporte plusieurs processus de compilation. La portabilité est dépendante de la qualité de portage des JVM sur chaque OS [49].

4.4.2 Netbeans

NetBeans est un environnement de développement intégré (IDE) open source pour le développement avec Java, PHP, C++ et d'autres langages de programmation, placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License). NetBeans est également appelé plate-forme de composants modulaires utilisés pour développer des applications de bureau Java.

NetBeans est codé en Java et fonctionne sur la plupart des systèmes d'exploitation avec une machine virtuelle Java (JVM), y compris Solaris, Mac OS et Linux. il utilise des composants, également appelés modules, pour permettre le développement de logiciels. NetBeans installe dynamiquement les modules et permet aux utilisateurs de télécharger des fonctionnalités mises à jour et des mises à niveau authentifiées numériquement, les modules IDE NetBeans incluent NetBeans Profiler, un outil de conception d'interface utilisateur graphique (GUI) et l'éditeur JavaScript NetBeans, la réutilisabilité du framework NetBeans simplifie le développement d'applications de bureau Java Swing, qui fournit des capacités d'extension de plate-forme aux développeurs tiers [50].

4.4.3 JDom

JDOM est une API open source Java dont le but est de représenter et manipuler un document XML de manière intuitive. Le but de JDOM est de faciliter la manipulation au sens large de document XML : lecture d'un document, représentation sous forme d'arborescence, manipulation de cet arbre, définition d'un nouveau document, exportation vers plusieurs formats cibles etc.

JDOM propose plusieurs fonctionnalités :

- Création de documents XML.
- Encapsulation d'un document XML sous la forme d'objets Java de l'API.
- Exportation d'un document dans un fichier, un flux SAX ou un arbre DOM.
- Support de XSLT.
- Support de XPath.

JDOM est développée spécifiquement en et pour Java en utilisant les fonctionnalités de Java au niveau syntaxique et sémantique (utilisation des collections de Java 2, de l'opérateur `new` pour instancier des éléments, redéfinition des méthodes `equals()`, `hashCode()`, `toString()`, implémentation des interfaces `Cloneable` et `Serializable`, ...) [51].

4.4.4 Collection de test

Nous avons utilisé la collection `sws-test-collections` qui contient des services web sémantiques écrits en OWL-S 1.0, OWL-S 1.1 et SAWSDL de différents domaines (communication, économie, éducation, alimentation, géographie, médical, voyage, arme). Ainsi que la collection `handon` qui contient 15000 services web dont nous avons converti une sélection de ces services web en services web sémantiques écrits en OWL-S 1.0.

Nous avons construit aussi une base de données des votes (`rating dataset`) de 339 utilisateurs avec 10 évaluations chacun (3390 votes) sur les 1000 services OWL-S 1.1 et 5 évaluations chacun (1695 votes) sur les 500 services OWL-S 1.0 pour avoir une base de données riche et diversifiée.

4.5 Implémentation de l'application

Pour évaluer et expérimenter notre approche, nous avons développé un outil nommé Système de Recommandation Collaboratif Sémantique de services web (SCRS) à base de java et sous l'environnement Netbeans, il dispose d'une interface graphique GUI facile à utiliser afin de mettre en œuvre les principaux composants de notre architecture.

La figure 4.7 représente l'interface principale de notre application qui contient toutes les fonctionnalités. Le premier bouton submit retourne les services web précédemment utilisés et évalués par un utilisateur, ce dernier doit d'abord entrer son Id pour avoir cette liste et sélectionner par la suite un de ces services web retournés. Le deuxième bouton submit retourne la liste des Top-K services web sémantiques dans un ordre décroissant selon la valeur de similarité, plus la valeur est proche de 1 plus la similarité est meilleure.

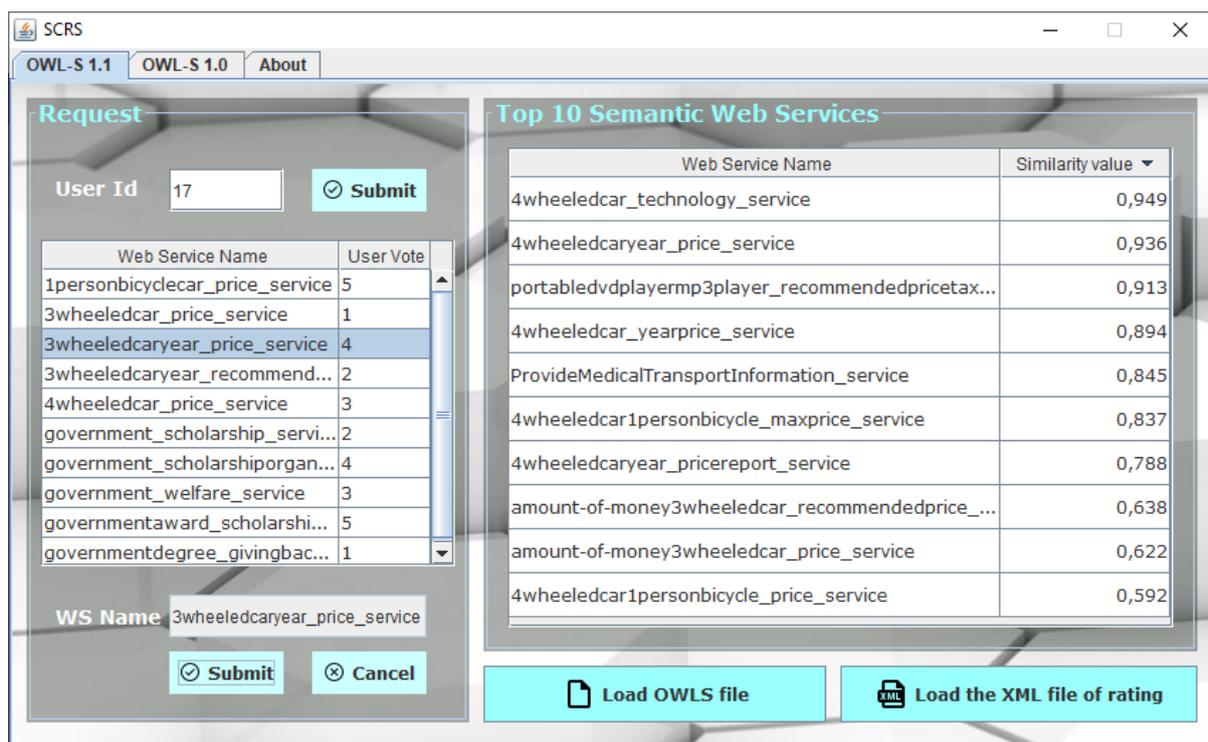


FIGURE 4.7: Interface principale de SCRS.

Le bouton **Load OWLS file** affiche le fichier OWLS d'un service web sémantique (figure 4.8) sélectionné parmi la liste des Top-K.

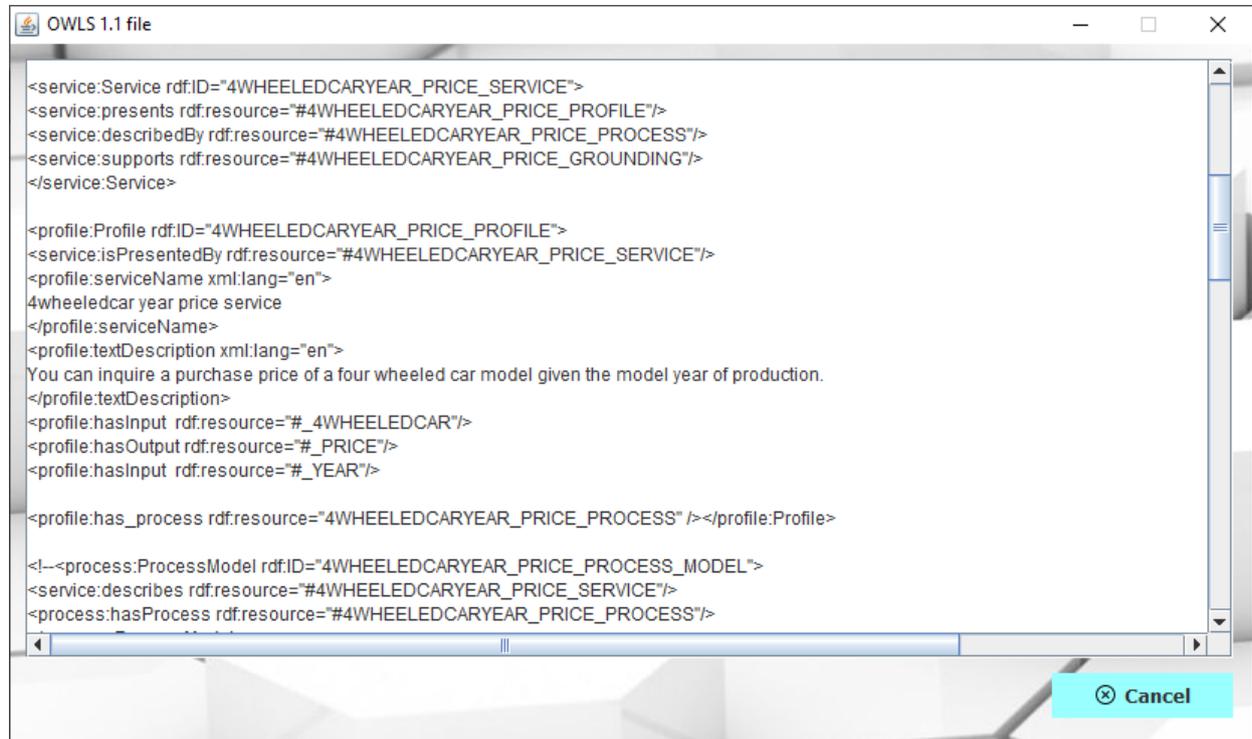
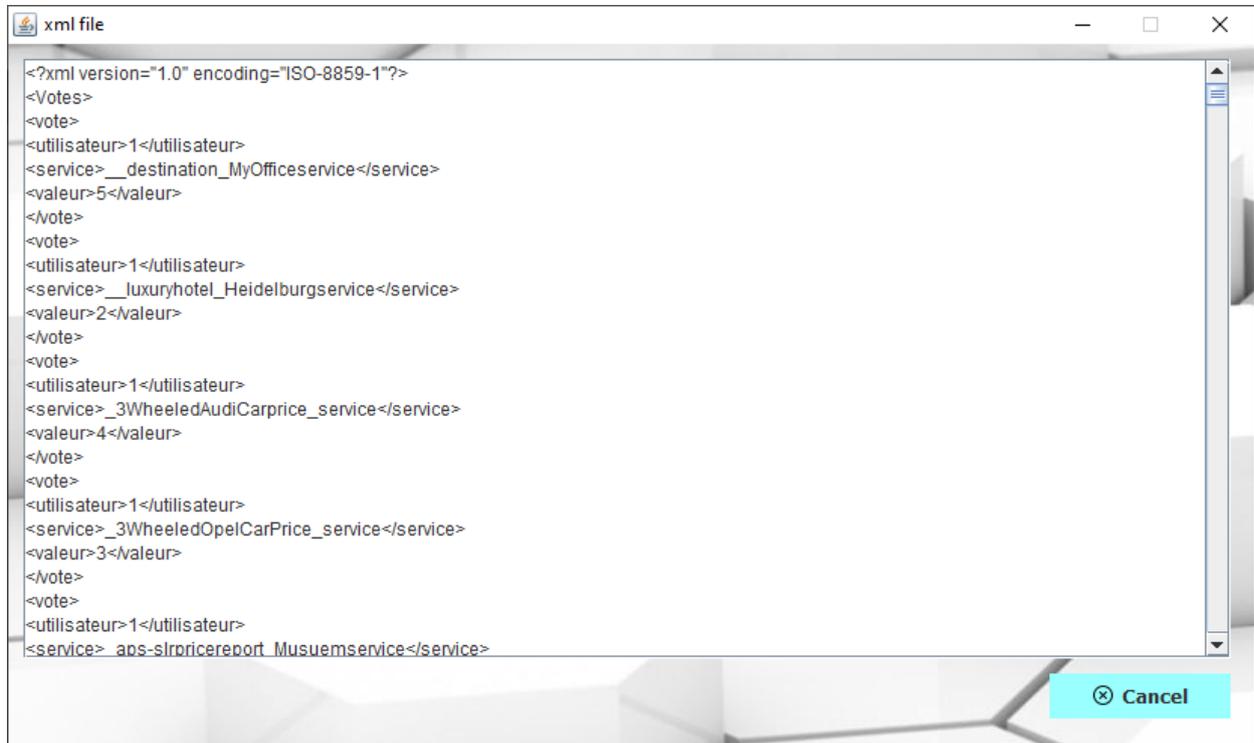


FIGURE 4.8: Fichier OWLS.

La balise `<profile:Profile>` représente la classe `ServiceProfile` que nous avons travaillée avec dans le filtrage sémantique.

Le bouton **Load the XML file of rating** affiche le fichier XML de la base de données des votes des utilisateurs représenté dans la figure 4.9.



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Votes>
<vote>
<utilisateur>1</utilisateur>
<service>__destination_MyOfficeservice</service>
<valeur>5</valeur>
</vote>
<vote>
<utilisateur>1</utilisateur>
<service>__luxuryhotel_Heidelbergservice</service>
<valeur>2</valeur>
</vote>
<vote>
<utilisateur>1</utilisateur>
<service>_3WheeledAudiCarprice_service</service>
<valeur>4</valeur>
</vote>
<vote>
<utilisateur>1</utilisateur>
<service>_3WheeledOpelCarPrice_service</service>
<valeur>3</valeur>
</vote>
<vote>
<utilisateur>1</utilisateur>
<service>_aps-slrpricereport_Musuemservice</service>
```

FIGURE 4.9: Fichier XML des votes.

L'onglet OWL-S 1.0 dispose de la même interface et les mêmes fonctionnalités que l'onglet OWL-S 1.1, c'est seulement la base de données des services web sémantiques qui change (services web OWL-S version 1.0).

4.6 Évaluation expérimentale

Plusieurs métriques d'évaluation expérimentale sont utilisées pour évaluer les performances, la validité et l'efficacité d'un système de recommandation. Pour réaliser cette expérimentation, nous avons utilisé le DCG (Discounted Cumulative Gain) qui permet de mesurer la qualité des top-k services web recommandés pour l'utilisateur cible, ainsi que les métriques prédictives MAE (Mean Absolute Error) et RMSE (Root Mean Squared Error) qui calculent la précision des prédictions par rapport à l'évaluation réelle effectuée par l'utilisateur.

Nous proposons l'équation suivante pour évaluer la qualité de la liste des Top-K services web sémantiques recommandés.

$$DCG_{sim} = \sum \frac{(2^{sim} - 1)}{\log(1 + p_i)} \quad (IV.3)$$

DCG_{sim} : Le gain cumulatif escompté du top-k en termes de similarité.

p_i : La position du service web i dans la liste des top-k.

Plus la valeur de DCG est élevée, plus la qualité de la liste des top-k services web recommandés est meilleure.

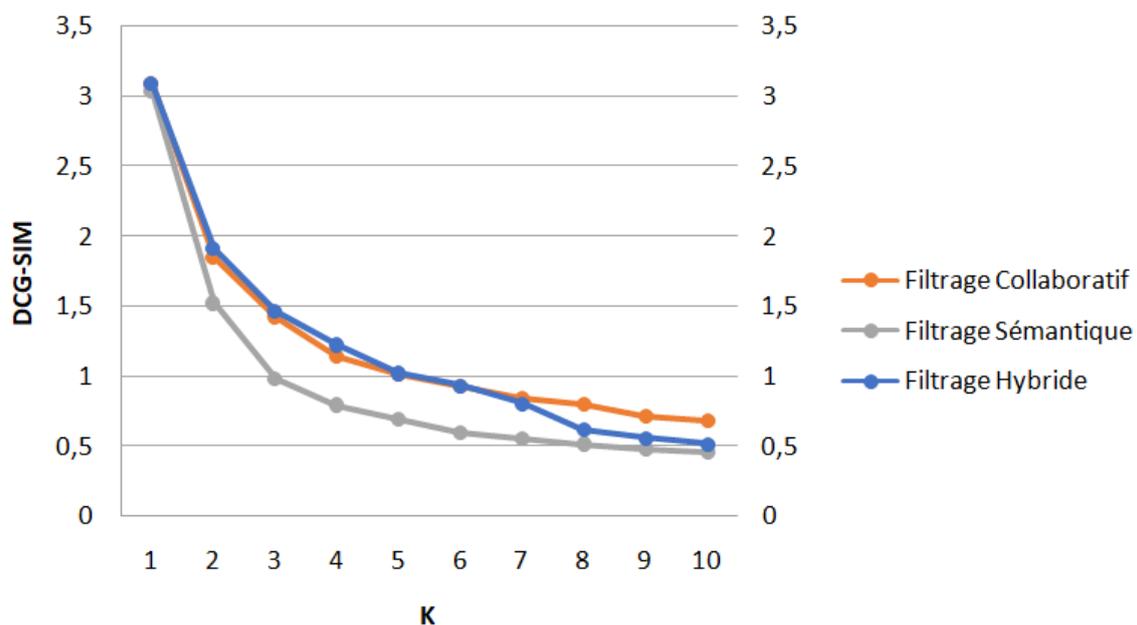


FIGURE 4.10: Valeurs de DCG_{sim} en se basant sur le filtrage collaboratif et/ou le filtrage sémantique.

Dans la figure 4.10, nous avons comparé trois techniques de filtrage : filtrage collaboratif, filtrage sémantique et filtrage hybride (collaboratif sémantique), le filtrage sémantique seul a montré une faible performance par rapport aux deux autres techniques. Notre technique hybride a montré une meilleure performance en termes de similarité plus particulièrement à partir de la 7^{ème} position.

Pour évaluer la précision d'une prédiction, nous avons utilisé la mesure MAE qui correspond à l'erreur absolue moyenne entre l'évaluation réelle et la prédiction. La mesure est calculée par la formule suivante :

$$MAE = \frac{1}{n} \sum_{i=1}^n |\bar{R}_i - R_i| \quad (IV.4)$$

R_i : La valeur de similarité mesurée.

\bar{R}_i : La valeur de similarité prédite.

n : Le nombre des items prédits.

La 2^{ème} métrique prédictive que nous avons utilisée est le RMSE. Cette mesure est utile lorsque que nous voulons donner plus de criticité aux erreurs importantes. Elle est exprimée comme suit :

$$RMSE = \sqrt{\frac{(\bar{R}_i - R_i)^2}{n}} \quad (IV.5)$$

Les résultats retournés par le MAE et le RMSE sont présentés dans les tableaux suivants :

Service Web	Évaluation	Sim	MAE	RMSE
1	0.894	0.894	0.000	0.000
2	0.894	0.894	0.000	0.000
3	0.894	0.816	0.0078	0.002467
4	0.8784	0.816	0.00624	0.001973
5	0.86592	0.815	0.005092	0.00161
6	0.855736	0.766	0.008974	0.002838
7	0.837789	0.766	0.007179	0.00227
8	0.823431	0.756	0.006743	0.002132
9	0.809945	0.672	0.013794	0.004362
10	0.782356	0.581	0.020136	0.006367

TABLE 4.1: Évaluation de la performance de prédiction du filtrage collaboratif sémantique.

Service Web	Évaluation	Sim	MAE	RMSE
1	0.894	0.894	0.000	0.000
2	0.894	0.894	0.000	0.000
3	0.894	0.816	0.0078	0.002467
4	0.8784	0.816	0.00624	0.001973
5	0.86592	0.756	0.010992	0.003476
6	0.843936	0.756	0.008794	0.002781
7	0.826349	0.707	0.011935	0.003774
8	0.802479	0.707	0.009548	0.003019
9	0.783383	0.707	0.007638	0.002415
10	0.768107	0.707	0.006111	0.001932

TABLE 4.2: Évaluation de la performance de prédiction du filtrage collaboratif.

Service Web	Évaluation	Sim	MAE	RMSE
1	0.815	0.815	0.000	0.000
2	0.815	0.766	0.0049	0.00155
3	0.8052	0.766	0.00392	0.00124
4	0.79736	0.672	0.012536	0.003964
5	0.772288	0.581	0.019129	0.006049
6	0.73403	0.498	0.023603	0.007464
7	0.686824	0.498	0.018882	0.005971
8	0.649059	0.479	0.017006	0.005378
9	0.615048	0.479	0.013605	0.004302
10	0.587838	0.475	0.010884	0.003442

TABLE 4.3: Évaluation de la performance de prédiction du filtrage sémantique.

En observant les deux figures 4.11 et 4.12, nous notons que l'utilisation du filtrage hybride (collaboratif sémantique) contribue à une meilleure précision par rapport au filtrage collaboratif seul ou bien le filtrage sémantique seul car le taux d'erreur de notre système est inférieur à celui des deux autres techniques.

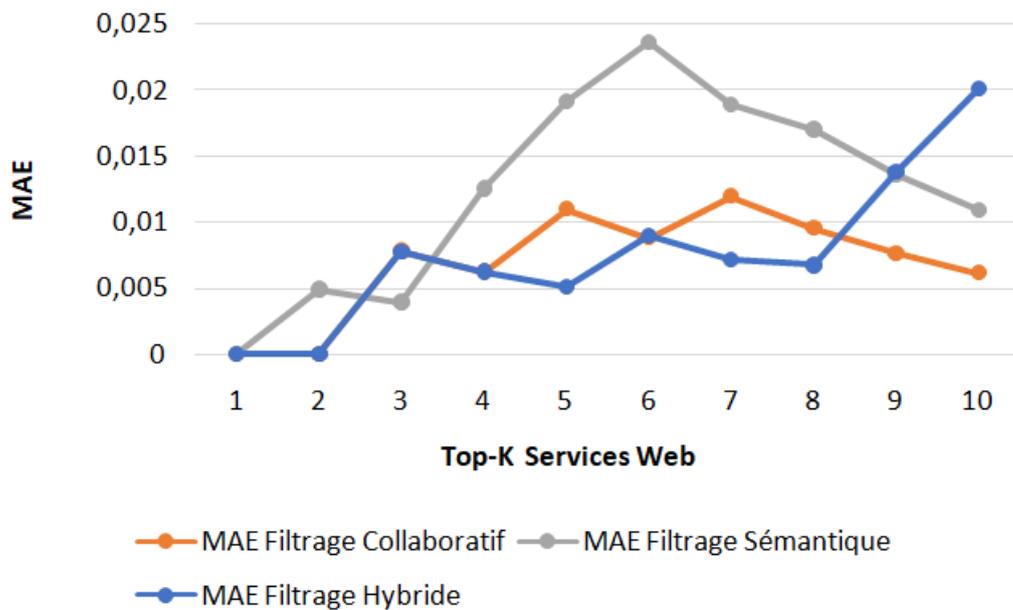


FIGURE 4.11: Evaluation basée sur MAE.

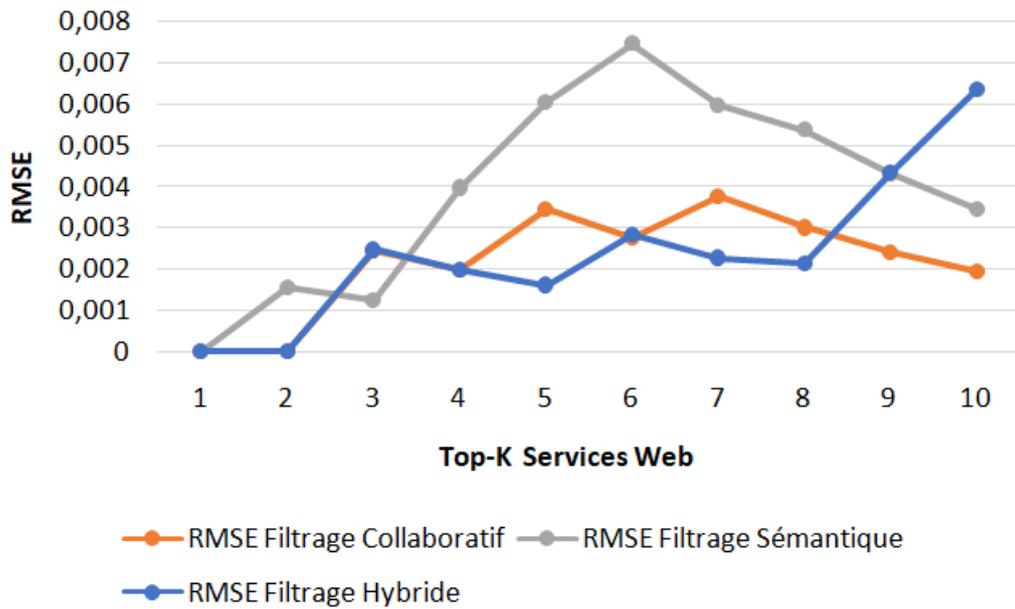


FIGURE 4.12: Evaluation basée sur RMSE.

4.7 Conclusion

Dans ce chapitre, nous avons présenté l'architecture de notre système de recommandation, les outils logiciels utilisés et le prototype que nous avons développé comme support à notre approche. Nous avons ainsi discuté l'expérimentation mise en place afin d'évaluer les performances de notre système de recommandation et sa validité et démontrer que le système proposé peut contribuer à l'assistance de l'utilisateur.

Conclusion générale et perspectives

Les systèmes de recommandation sont devenus de plus en plus populaires et constituent actuellement un composant incontournable de beaucoup d'applications dans de différents domaines, ces systèmes jouent un rôle important dans l'orientation des choix des utilisateurs en fournissant des recommandations adaptées à leurs besoins, leurs goûts ou leurs moyens afin de les aider à accéder à des ressources utiles et intéressantes au sein d'un espace de données important.

Les services web quant à eux sont des systèmes d'échange d'informations permettant la communication entre applications et systèmes hétérogènes et qui sont de plus en plus utilisés lors du développement des applications web. Cependant, avec la prolifération des services web, il est nécessaire d'automatiser la sélection et la recommandation de ces derniers.

Dans le cadre de notre projet de fin d'études, nous avons proposé un système collaboratif sémantique pour la sélection et la recommandation de services web, l'intégration des technologies du web sémantique qui a réussi à traiter les limites des systèmes de filtrage collaboratifs traditionnels.

Dans notre travail, nous avons réalisé une application de recommandation qui retourne une liste des Top-K services web OWL-S à un utilisateur cible, et cela en se basant sur la corrélation de Pearson et la similarité sémantique pour établir le taux de similarité entre le service web sélectionné par l'utilisateur et ceux de la collection.

Comme perspectives nous envisageons d'apporter quelques améliorations à savoir :

- Implémenter un outil automatique ou semi-automatique pour convertir les services web WSDL en services web OWL-S, pour avoir une meilleure sélection de services web sémantiques à recommander, car ils sont devenus la technologie la plus développée pour l'interaction de machine à machine.
- Intégrer la confiance dans notre système de recommandation, en se basant non seule-

ment sur la similarité qui existe entre les services web, mais aussi sur la confiance qui existe entre les services web et les utilisateurs.

Bibliographie

- [1] “What is service-oriented architecture ?” [Online]. Available : <https://medium.com/@SoftwareDevelopmentCommunity/what-is-service-oriented-architecture-fa894d11a7ec>
- [2] D. Hanane and L. Siham, “Une approche formelle pour la composition des services web,” Master’s thesis, UNIVERSITÉ ECHAHID HAMMA LAKHDAR - EL-OUED, juin 2015.
- [3] “Service-Oriented Architecture,” Aug. 2018, section : GBlog. [Online]. Available : <https://www.geeksforgeeks.org/service-oriented-architecture/>
- [4] “What are Web Services? - Tutorialspoint.” [Online]. Available : https://www.tutorialspoint.com/webservices/what_are_web_services.htm
- [5] “Service web,” Aug. 2020, page Version ID : 173480689. [Online]. Available : https://fr.wikipedia.org/w/index.php?title=Service_web&oldid=173480689
- [6] “Web Services Architecture Requirements.” [Online]. Available : <https://www.w3.org/TR/wsa-reqs/>
- [7] “Les services Web.” [Online]. Available : <https://pub.phyks.me/sdz/sdz/les-services-web.html>
- [8] “Webservices protocol stack.” [Online]. Available : <https://joyofwisdom.wordpress.com/>
- [9] “Openclassrooms - les services web.” [Online]. Available : <https://www.pfl-cepia.inra.fr/uploads/images/GestionDonneesImages/unites/GdpDoc/OpenClassrooms-servicesWeb.pdf>
- [10] “Introduction to Web Service Technologies.” [Online]. Available : https://docs.oracle.com/cd/E13224_01/wlw/docs103/guide/webservices/conBasicWebServiceTechnologies.html
- [11] “Les services web de type soap.” [Online]. Available : <https://www.jmdoudoux.fr/java/dej/chap-service-web.htm>
- [12] “Service web (soap).” [Online]. Available : [http://lig-membres.imag.fr/plumejeaud/NFE107-fichesLecture/Service%20Web%20\(SOAP\).ppt.pdf](http://lig-membres.imag.fr/plumejeaud/NFE107-fichesLecture/Service%20Web%20(SOAP).ppt.pdf)
- [13] “Les web services.” [Online]. Available : <http://www-igm.univ-mlv.fr/~dr/XPOSE2004/woollams/wsdl.html>

- [14] B. Boudaa, “Les communautés pour une haute disponibilité des services Web maintenant la médiation sémantique dans les compositions,” Ph.D. dissertation, Jul. 2009.
- [15] R. khaled, “Découverte des services dans les systèmes d’information géographiques répartis sous réseau ad-hoc,” Ph.D. dissertation, Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie Département d’Informatique, 18 janvier 2011.
- [16] A. Sofiane and Z. C. Bassim, “La sélection des services web À base d’algorithme de la recherche harmonique,” Master’s thesis, Université Abou Bekr Belkaid– Tlemcen Faculté des Sciences Département d’Informatique, 22 Juin 2015.
- [17] “Semantic-based service discovery for web services,” Master’s thesis, Université de Béjaïa.
- [18] B. Hakim, “Sélection des web services sémantiques,” Ph.D. dissertation, Université A/Mira de Béjaïa Faculté des Sciences Exactes Département d’Informatique, Juin 2012.
- [19] C. LOPEZ-VELASCO, “Sélection et composition de services web pour la génération d’applications adaptées au contexte d’utilisation,” Ph.D. dissertation, Université Joseph-Fourier - Grenoble I, 2008.
- [20] D. Abdelhamid and B. M. Amin, “La selection des services web a base des systemes immunitaires artificiels (clonalg),” Master’s thesis, Université Abou Bakr Belkaid– Tlemcen Faculté des Sciences Département d’Informatique, 03 juillet 2012.
- [21] P. Kellert and F. Toumani, “Les web services sémantiques,” 2004. [Online]. Available : https://www.irit.fr/journal-i3/hors_serie/annee2004/revue_i3_hs2004_01_07.pdf
- [22] N. Temglit, H. Aliane, and M. Ahmed Nacer, “Un modèle de composition des services web sémantiques,” *Revue Africaine de la Recherche en Informatique et Mathématiques Appliquées*, vol. Volume 11, 2009 - Special..., pp. 115 – 133, Sep. 2009. [Online]. Available : <https://arima.episciences.org/1928>
- [23] N. Ould Ahmed M’Barek and S. Tata, “Services Web : revue des approches de description sémantique,” in *SIIE 2008 : Système d’Information et Intelligence Economique*, Hammamet, Tunisia, Feb. 2008, pp. 1 – 7. [Online]. Available : <https://hal.archives-ouvertes.fr/hal-01378724>
- [24] J. de Bruijn, H. Lausen, A. Polleres, and D. Fensel, “The web service modeling language WSML : An overview,” in *The Semantic Web : Research and Applications*, ser. Lecture Notes in Computer Science, Y. Sure and J. Domingue, Eds. Springer, pp. 590–604.
- [25] I. Benouaret, “Un système de recommandation contextuel et composite pour la visite personnalisée de sites culturels.” Ph.D. dissertation, Université de Technologie de Compiègne, 2017.
- [26] R. Burke, “Hybrid Recommender Systems : Survey and Experiments,” *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, Nov. 2002. [Online]. Available : <https://doi.org/10.1023/A:1021240730564>

- [27] M. ZOHRA, “Recommandation de collaborateurs pertinents dans un environnement d’apprentissage collaboratif,” Ph.D. dissertation, UNIVERSITE BADJI MOKHTAR-ANNABA Faculté des Sciences de l’Ingéniorat Département d’Informatique, 23 Juin 2018.
- [28] M. N. AMINA and S. DJIHANE, “Application et exploration de l’approche de découpage des systèmes de recommandations pré-contextuelle,” Master’s thesis, Université Abou Bakr Belkaid– Tlemcen Faculté Des Sciences Département d’Informatique, 27 Juin 2019.
- [29] A.-M. Kermarrec, “L’algorithme de filtrage collaboratif en cinq atouts,” Jan. 2018. [Online]. Available : <https://www.mediago.com/fr/blog/cinq-atouts-du-filtrage-collaboratif/>
- [30] H. Oufaida and O. Nouali, “Le filtrage collaboratif et le web 2.0,” *Document numérique*, vol. Vol. 11, no. 1, pp. 13–35, 2008, publisher : Lavoisier. [Online]. Available : <https://www.cairn.info/revue-document-numerique-2008-1-page-13.htm>
- [31] N. Amine, “Papyrus : Un système de gestion et de recommandation d’articles de recherche,” Master’s thesis, Université de Montréal Faculté des arts et des sciences, Juillet 2009.
- [32] M. MAATALLAH, “Une technique hybride pour les systèmes de recommandation,” Ph.D. dissertation, Université Mokhtar Badji Annaba.
- [33] L. Assia and K. Khadidja, “Un système de recommandation pour l’assistance à la navigation sur internet,” Master’s thesis, UNIVERSITE KASDI MERBAH OUARGLA Faculté des Nouvelles Technologies de l’Information et de la Communication Département d’Informatique et des Technologies de l’Information, 04 juin 2016.
- [34] K. Tso, L. Marinho, and L. Schmidt-Thieme, “Tag-aware recommender systems by fusion of collaborative filtering algorithms,” Jan. 2008, pp. 1995–1999.
- [35] M. B. Blake and M. F. Nowlan, “A web service recommender system using enhanced syntactical matching,” in *IEEE International Conference on Web Services (ICWS 2007)*, pp. 575–582.
- [36] F. Lecue and A. Delteil, “Making the difference in semantic web service composition,” vol. 2, pp. 1383–1388.
- [37] R. Amorim, D. Claro, D. Lopes, P. Albers, and A. Andrade, “Improving web service discovery by a functional and structural approach,” journal Abbreviation : Proceedings - 2011 IEEE 9th International Conference on Web Services, ICWS 2011 Pages : 418 Publication Title : Proceedings - 2011 IEEE 9th International Conference on Web Services, ICWS 2011.
- [38] F. SLAIMI, “Découverte et recommandation de services web,” Ph.D. dissertation, Aix-Marseille Université, 23 mars 2016.
- [39] Z. Zheng, H. Ma, M. R. Lyu, and I. King, “WSRec : A collaborative filtering based web service recommender system,” in *2009 IEEE International*

- Conference on Web Services*. IEEE, pp. 437–444. [Online]. Available : <http://ieeexplore.ieee.org/document/5175854/>
- [40] X. Chen, Z. Zheng, Q. Yu, and M. R. Lyu, “Web service recommendation via exploiting location and QoS information,” vol. 25, no. 7, pp. 1913–1924, conference Name : IEEE Transactions on Parallel and Distributed Systems.
- [41] Y. Jiang, J. Liu, M. Tang, and X. Liu, “An effective web service recommendation method based on personalized collaborative filtering,” in *2011 IEEE International Conference on Web Services*, pp. 211–218.
- [42] S.-G. Deng, L.-T. Huang, J. Wu, and Z.-H. Wu, “Trust-based personalized service recommendation : A network perspective,” vol. 29, no. 1, pp. 69–80. [Online]. Available : <https://doi.org/10.1007/s11390-014-1412-2>
- [43] A. Naziha, “Contribution des réseaux sociaux dans l’ingénierie des services web,” Ph.D. dissertation, Université Djillali Liabes de Sidi Bel Abbes Faculté des Sciences Exactes.
- [44] U. Manikrao and T. Prabhakar, “Dynamic selection of web services with recommendation system,” in *International Conference on Next Generation Web Services Practices (NWeSP’05)*. IEEE, pp. 117–121. [Online]. Available : <http://ieeexplore.ieee.org/document/1592416/>
- [45] Z. Chen, Y. Jiang, and Y. Zhao, “A collaborative filtering recommendation algorithm based on user interest change and trust evaluation.” vol. 4, pp. 106–113.
- [46] L. Yao, Q. Sheng, A. Segev, and J. Yu, “Recommending web services via combining collaborative filtering with content-based features,” pp. 42–49.
- [47] L. Yao, Q. Z. Sheng, A. H. Ngu, J. Yu, and A. Segev, “Unified collaborative and content-based web service recommendation,” vol. 8, no. 3, pp. 453–466. [Online]. Available : <http://ieeexplore.ieee.org/document/6894179/>
- [48] “le langage uml.” [Online]. Available : <https://www.lucidchart.com/pages/fr/langage-uml>
- [49] “Java (langage) - définition et explications.” [Online]. Available : <https://www.techno-science.net/glossaire-definition/Java-langage.html>
- [50] “What is netbeans? - definition from techopedia.” [Online]. Available : <https://www.techopedia.com/definition/24735/netbeans>
- [51] “Développons en java - les modèles de documents.” [Online]. Available : <https://www.jmdoudoux.fr/java/dej/chap-jdom.htm>

Résumé

Le travail présenté dans ce mémoire se situe dans le domaine des systèmes de recommandation qui sont devenus des composantes incontournables dans la plupart des applications web, pour guider les utilisateurs vers les ressources qui correspondent à leurs préférences et intérêts dans les grandes masses de données disponibles. De nos jours, Le nombre de serveurs offrant des services web, est en constante progression. Il est alors important d'automatiser la sélection et la recommandation de ces services Web. Le but de notre travail est de créer une application de recommandation basée sur le filtrage collaboratif et le filtrage sémantique afin de générer une liste de recommandation de services web plus pertinente et adaptée à un utilisateur cible.

Mots-clés : Services web, Système de recommandation, Filtrage collaboratif, Sémantique, Services web Sémantiques.

Abstract

The work presented in this memory is in the field of recommendation systems which have become essential components in most web applications, to guide users to resources that match their preferences and interests in the large masses of available data. Nowadays, the number of servers offering web services is constantly increasing. It is therefore important to automate the selection and recommendation of these web services. The aim of our work is to create a recommendation application based on collaborative filtering and semantic filtering in order to generate a web services recommendation list that is more relevant and adapted to a target user.

Keywords : Web services, Recommendation system, Collaborative filtering, Semantics, Semantic web services.

ملخص

العمل المقدم في هذه الأطروحة هو في مجال أنظمة التوصية التي أصبحت مكونات أساسية في معظم تطبيقات الويب، لتوجيه المستخدمين إلى الموارد التي تتوافق مع تفضيلاتهم واهتماماتهم في الكتل الكبيرة من البيانات المتاحة. في الوقت الحاضر، يتزايد باستمرار عدد الخوادم التي تقدم خدمات الويب. لذلك من المهم أتمتة الاختيار والتوصية بخدمات الويب هذه. الهدف من عملنا هو إنشاء تطبيق توصية يعتمد على التصفية التعاونية والتصفية الدلالية من أجل إنشاء قائمة توصية خدمات الويب أكثر صلة بالمستخدم المستهدف.

الكلمات المفتاحية : خدمات الويب، أنظمة التوصية، التصفية التعاونية، دلالات، خدمات الويب الدلالية.