

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Ain Temouchent Belhadj Bouchaib



Faculté des sciences et technologies
Département des Mathématiques et de l'Informatique

Mémoire

En vue de l'obtention du Diplôme de Master en Informatique

Option :

Réseaux et Ingénierie des Données (RID)

Thème

Extension du WSDL par SQOS Trust en utilisant
le Social Computing.

Présenté par :

M^{lle} ZAHAF Kheira

M^{lle} ZAOUI Chourouk Halima

Devant le jury composé de :

Président : *M^r* BENOMAR Mohammed Lamine MCA U.A.T.B.B.

Examineur : *M^r* MEDEDJEL Mansour MCA U.A.T.B.B.

Encadrante : *M^{me}* ABDERRAHIM Naziha MCB U.A.T.B.B.

Année universitaire 2021/2022

Dédicace

À ma famille et mes cheres amis Nour eldjihen, Wahiba , Fatima et Djamel eddine.

ZAHAF Kheira

Dédicace

Remerciements et louanges à Dieu, de m'avoir donné la foi et la force pour accomplir ce travail, Prière et salut sur notre prophète « Mohamed » et sur sa famille et ces compagnons.

Je dédie ce mémoire aux personnes qui me sont chers et qui m'ont soutenu tout au long des épreuves et qui ont donné lumière à ce travail. Je cite particulièrement :

Mon Père, mon bonheur perdu, qui est parti trop tôt avant de voir ce que je suis devenue. Que Dieu l'accueille dans son vaste paradis.

Ma Mère, en vous, je vois la maman parfaite, toujours prête à se sacrifier pour le bonheur de ses enfants. Merci pour tout.

Mon frère et ma sœur, que je ne cesserai jamais d'aimer, pour qui ma réussite est très importante.

Que Dieu vous paye pour tous vos bienfaits.

Mes chères copines, Saadia, Sihem, Soltana, Abir.

A tous ceux qui me sont chers, si mes vœux pouvaient avoir quelques pouvoirs j'en serai profondément heureuse car je veux pour vous et vos familles toutes les réussites et satisfactions de ce monde.

ZAOUI Chourouk Halima

Remerciement

Avant tout nous remercions *ALLAH* le Tout Puissant de nous avoir donné le courage, la volonté et la patience d'entamer et de mener ce présent travail à terme.

Nos plus sincères remerciements vont à notre encadrant *M^{me} ABDERRAHIM Naziha* qui a accepté de diriger ce travail et qui nous a fait bénéficier de son expérience et de ses connaissances. Qu'elle trouve ici notre respect et notre gratitude pour ses conseils, ses encouragements, ses qualités humaines et surtout la confiance qu'il nous a témoignée pour réaliser ce travail.

Nous profitons de cette occasion aussi pour exprimer nos plus sincères remerciements aux Membres du jury *M^r BENOMAR Mohammed Lamine* et *M^r MEDEDJEL Mansour* qui ont accepté d'évaluer et d'examiner ce travail.

Enfin, nous remercions tous les enseignants du département Mathématique et informatique pour leur soutien tous le long du cursus.

Résumé

Avec l'avènement de la technologie des services Web, la conception d'applications informatiques repose, de plus en plus, sur l'utilisation des services Web existants. Ainsi choisir les services Web adéquats pour l'application en cours de conception n'est pas une chose aisée. Ceci nécessite en effet de trouver les services Web qui satisfont les besoins fonctionnels de l'application en cours de développement et qui, autant que possible, respectent les autres besoins (besoins non fonctionnels) et en particulier le niveau de qualifié requis. L'objectif de notre approche est d'améliorer la découverte et la composition des services Web en se basant sur leur comportement social et en déterminant s'ils sont dignes de confiance (SWW : Service Web Trustworthy) pour une éventuelle composition ou non, et ce en utilisant l'analyse des réseaux sociaux des services Web. A partir de l'analyse des réseaux sociaux, la valeur de Trust peut être déterminé et ajouté comme extension au fichier WSDL.

Mots clés : services web, social computing, qualité social des services Web, extension wsdl.

Abstract

With the advent of Web services technology, the design of computer applications increasingly relies on the use of existing Web services. So choosing the right web services for the application being designed is not easy. This requires finding Web services that meet the functional needs of the application being developed and that, as far as possible, meet other needs (non-functional needs) and in particular the level of qualification required. The goal of our approach is to improve the discovery and composition of Web services based on their social behavior and whether they are trustworthy (SWW : Trustworthy Web Service) for possible composition or not, using social network analysis of Web services. From the social networks analysis, the value of Trust can be determined and added as an extension to the WSDL file.

Keywords : web services, social computing, social quality of web services, wsdl extension.

ملخص

مع ظهور تكنولوجيا خدمات الويب ، يعتمد تصميم تطبيقات الكمبيوتر بشكل متزايد على استخدام خدمات الويب الحالية. لذا فإن اختيار خدمات الويب المناسبة للتطبيق الجاري تصميمه ليس بالأمر السهل. يتطلب هذا بالفعل العثور على خدمات الويب التي تلبى الاحتياجات الوظيفية للتطبيق قيد التطوير والتي تحترم ، قدر الإمكان ، الاحتياجات الأخرى (الاحتياجات غير الوظيفية) وعلى وجه الخصوص مستوى التأهيل المطلوب الهدف من نهجنا هو تحسين اكتشاف وتكوين خدمات الويب بناءً على سلوكهم الاجتماعي وتحديد ما إذا كانت جديرة بالثقة (SWW: خدمة ويب جديرة بالثقة) للتكوين المحتمل أم لا ، باستخدام تحليل الشبكة الاجتماعية لخدمات الويب.

كلمات مفتاحية: خدمات الويب، الحوسبة الاجتماعية، الجودة الاجتماعية لخدمات الويب، ملحق wsdl

Table des matières

Table des figures	8
Liste des tableaux	9
Introduction générale	10
1 Services Web	12
1.1 Introduction	13
1.2 Architecture orientée services(SOA)	13
1.2.1 Les composants de la SOA.	14
1.2.2 Le fonctionnement de la SOA	14
1.3 Qu'est-ce qu'un service Web?	15
1.4 Modèle d'implémentation de services web	15
1.5 Le concept des services web	16
1.5.1 WSDL	16
1.5.2 UDDI	18
1.6 Architecture d'un service web	19
1.6.1 SOAP	19
1.6.2 REST	21
1.7 Fonctionnement des services web	22
1.8 Composition des services web	23
1.8.1 Les principales approches de composition de services Web	23
1.8.2 Langages de composition de services Web	25
1.9 Avantages et inconvénients des services web	25
1.10 Service web et qualité de service	26
1.11 Le courtier de Qos-SW	27
1.12 Conclusion	27
2 Analyse des réseaux sociaux des services Web (Social Computing)	28
2.1 Introduction	29
2.2 social computing	29

2.3	Du réseau social traditionnel au réseau social numérique	30
2.4	Réseau social sur le web	30
2.4.1	Principes de fonctionnement	31
2.4.2	Classification des réseaux sociaux	31
2.4.3	Réseaux sociaux et apprentissage	32
2.5	La confiance sur les réseaux sociaux	33
2.6	Les réseaux sociaux des services Web	34
2.7	Les qualités sociales de services Web	35
2.8	Conclusion	36
3	Etat de l'art : Composition des services web et extension wsdl	38
3.1	Introduction	39
3.2	La composition des services web et extension WSDL	39
3.3	Tableau comparatif	44
3.4	Synthèse	44
3.5	Conclusion	45
4	Conception et réalisation	46
4.1	Introduction	47
4.2	Environnement de développement	47
4.3	Langage de programmation	47
4.4	Implémentation	47
4.4.1	Conception	48
4.4.2	Les fonctions	49
4.4.3	Collection de données	51
4.5	Expérimentation	54
4.6	Conclusion	55
	Bibliographie	57

Table des figures

1.1	Les interactions entre les composants de la SOA [19].	14
1.2	Fonctionnement de la SOA [3].	14
1.3	Structure d'un document WSDL.	17
1.4	Le contenu du WSDL d'un service nommé MonService [16].	18
1.5	Message SOAP [10].	20
1.6	RESTful architecture.	21
1.7	Fonctionnement des services web [15].	22
1.8	Orchestration de services Web [30]	24
1.9	Chorégraphie Web Service [30].	24
2.1	Architecture de social computing [23].	30
2.2	La pyramide des besoins selon Maslow [24].	32
2.3	Réseau social de collaboration [37].	34
2.4	Réseau social de compétition [37]	35
4.1	Organigramme.	48
4.2	La liste des services.	49
4.3	Résultat de la fonction <code>calculate_egoism()</code>	49
4.4	Résultat de la fonction <code>calculate_honnetete()</code>	50
4.5	résultat de la fonction <code>calculate_cooperation()</code>	50
4.6	Taux de confiance pour chaque service.	50
4.7	Description WSDL du service « S32 ».	51
4.8	Comparaison des services.	54
4.9	Taux des trois qualités avec la valeur de confiance « Trust ».	55

Liste des tableaux

1.1	Comparaison REST et SOAP.	22
3.1	Tableau comparatif.	44
4.1	Table des transactions.	52
4.2	Exemples de transactions.	53

Introduction générale

Le Service-Oriented Computing (SOC) est un paradigme émergent pour l'informatique distribuée visant à changer la façon dont les applications logicielles sont conçues, livrées et consommées. Le principe fondamental du SOC est la possibilité de construire des applications distribuées sur le Web en utilisant les services Web (WS) comme éléments de base. La prolifération des services Web est considérée comme la deuxième vague d'évolution de l'ère Internet, déplaçant le Web d'une collection de pages à une collection de services utilisés par les agents de logiciels [17]. Cependant, plusieurs questions importantes restent non résolues, à savoir que les services Web :

- Ne connaissent qu'eux-mêmes, pas leurs utilisateurs ou leurs pairs.
- Limiter considérablement l'intervention des utilisateurs et fonctionner comme des boîtes noires.
- Ne tiennent compte que de leurs propres détails fonctionnels et non fonctionnels internes pendant l'exécution et ignorer d'autres détails externes, tels que les interactions passées avec les utilisateurs .
- Ne peuvent déléguer leurs invocations.
- Ne coopérer pas instantanément et volontairement les uns avec les autres. [35].

Le concept du social computing et, en particulier, la confiance sociale (Trust) a été introduit par la communauté des services Web pour aborder ces problèmes.

Le social computing est un nouveau paradigme informatique émergent qui implique une approche multidisciplinaire dans l'analyse et la modélisation des comportements sociaux sur différents médias et plateformes pour produire des applications et des résultats intelligents et interactifs.

Le social computing fait référence généralement à un groupe de services Web qui sont réalisés par des groupes de personnes à travers, par exemple, les systèmes de recommandation, les systèmes de confiance/réputation, les réseaux sociaux, les réseaux peer-to-peer, les wikis, et les enchères en ligne [44].

Par conséquent, la confiance dans SOC devient un enjeu critique compte tenu du fait que la confiance joue un rôle important dans chaque phase du déploiement des services.

L'objectif de notre approche est de déterminer la valeur de confiance « trust » entre les services Web en se basant sur leur interaction passé et en utilisant l'analyse des réseaux sociaux des services-Web, afin d'améliorer la sélection, la découverte et éventuellement la composition des services Web. A la fin, la valeur de Trust a été ajoutée comme extension au fichier WSDL, dans le but de déceler les services Web digne de confiance.

Notre mémoire est organisé comme suit : Le premier chapitre décrit la notion de services web et leurs technologies ainsi que sa composition, et les principaux standards qu'ils supportent (WSDL, SOAP, UDDI).

Le deuxième chapitre présente le concept de "Social computing", ainsi que l'évolution des réseaux sociaux sur le web et la qualité des services web.

Le troisième chapitre présente l'état de l'art des recherches existants.

Le quatrième chapitre est consacré pour la conception ,la réalisation ainsi que les principaux outils utilisés pour la création de notre application. Enfin, une conclusion générale avec quelques perspectives.

Chapitre 1

Services Web

1.1 Introduction

Depuis le début des années 1990, Internet et surtout le Web ont révolutionné les échanges tant au niveau des individus dans la sphère privée qu'au niveau professionnel. Les modes de communication ont complètement changé. Le transport de l'information se fait, de nos jours, principalement sous forme électronique à la vitesse de la «lumière». Les informations sont de plus disponibles quasiment en tout lieu et à toute heure [6].

Pour de nombreuses personnes, utiliser des services sur le Web va de soi. Acheter en ligne, lire son journal, réserver une table dans un restaurant, regarder un film sont autant de services que nous utilisons chaque jour. Alors que la personne devant son écran navigue sur Internet et prend des décisions, de nombreuses choses se passent entre l'homme et la machine. Mais il s'en passe tout autant entre les machines. Des requêtes et des réponses sont constamment envoyées entre le client et le serveur. Cet échange est rendu possible par le service web [26].

Dans ce chapitre, nous décrivons les technologies liées aux services web et sa composition, en se basant aussi sur trois méthodes (WSDL, SOAP, UDDI) des services web qui ne sont pas à mettre en opposition ni en comparaison. Chacun d'entre eux peut être adapté à une situation, l'idéal étant d'implémenter les trois lors de la création d'un "Web service".

1.2 Architecture orientée services(SOA)

Il n'y a pas de définition exacte de l'architecture orientée services. Il est vrai que plusieurs définitions ont été proposées, mais toutes s'accordent à dire que la SOA est un paradigme destiné à répondre à l'hétérogénéité et à l'interopérabilité des logiciels qui composent un système d'information.

Définition 1 : La SOA est un paradigme pour l'organisation et la consommation de services distribués. Il permet de fournir des services découvrables, invocables et composables. Selon cette définition, la SOA est une approche de construction d'un système d'information qui ne repose ni sur la représentation de ressources (logiciels, applications, etc.) ni d'objets (commandes, transporteurs, clients, etc.), mais sur la fonctionnalité requis par l'exécution des processus métier. Tous les éléments de l'architecture du système d'information s'articulent autour de la notion de services [19].

Définition 2 : SOA est un ensemble de composants qui peuvent être appelés, et dont les descriptions d'interface peuvent être publiées et découvertes [19].

1.2.1 Les composants de la SOA.

- Le fournisseur de service : correspond au propriétaire du service qui décrit, publie et garanti la disponibilité du service.
- Le client : correspond au demandeur de service. D'un point de vue technique, il recherche et invoque les services. il peut être lui-même un service.
- L'annuaire des services : Correspond à un registre de description de service qui fournit une fonction de publication de service pour les fournisseurs et une fonction de recherche de service pour les clients [19].

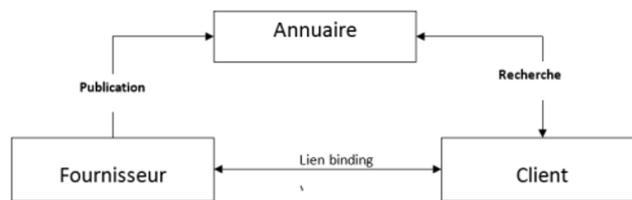


FIGURE 1.1: Les interactions entre les composants de la SOA [19].

1.2.2 Le fonctionnement de la SOA

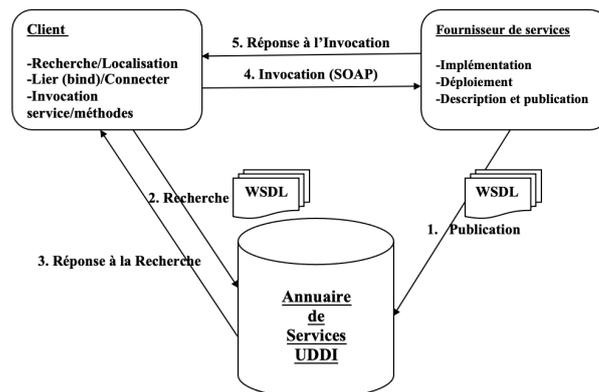


FIGURE 1.2: Fonctionnement de la SOA [3].

1.3 Qu'est-ce qu'un service Web ?

La technologie des services Web est un moyen rapide de distribution de l'information entre clients, fournisseurs, partenaires commerciaux et leurs différentes plates-formes. Selon la définition du W3C (World Wide Web Consortium), 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.

Définition 1 : Un Service Web est une application accessible à partir du Web. Il utilise les protocoles Internet pour communiquer et un langage standard pour décrire son interface [39].

Définition 2 : Papazoglou (2003) a défini les services web comme étant des éléments auto-descriptifs et indépendants des plateformes qui permettent la composition à faible coût des applications distribuées. Les services Web effectuent des fonctions allant de simples requêtes à des processus métiers complexes. Les services Web permettent aux organisations d'exposer leurs programmes résultats sur Internet en utilisant des langages (basés sur XML) et des protocoles standardisés et de les mettre en valeur via une interface auto-descriptive basée sur des formats standardisés et ouverts [42].

Les services Web fournissent des liens entre les applications. Ainsi, les applications utilisant différentes technologies peuvent envoyer et recevoir des données via des protocoles que tout le monde peut comprendre. Les services Web sont normalisés car ils utilisent les standards XML et HTTP pour transférer des données et ils sont compatibles avec de nombreux autres environnements de développement. Ils sont donc indépendants des plates-formes. C'est dans ce contexte qu'un intérêt très particulier a été attribué à la conception des services Web puisqu'ils permettent aux entreprises d'offrir des applications accessibles à distance par d'autres entreprises. Cela s'explique par le fait que les services Web n'imposent pas de modèles de programmation spécifiques.

1.4 Modèle d'implémentation de services web

Les services Web incluent l'exposition de nombreuses fonctions utiles via Un protocole réseau simple. Le protocole est SOAP (Simple Object Access Protocol), basé sur XML. L'accès à ces fonctions sur le réseau est effectué par le programme

dont le client écrit dans n'importe quel langage de programmation et exécute sur divers plate-forme (Windows, Linux, etc.). Mieux encore, la génération de ces clients peut se faire générer automatiquement par des outils basés sur des descripteurs de services. la description du service est assuré par un deuxième protocole de service Web, qui est WSDL (Web Services Description Language). WSDL Représente l'équivalent IDL de COM/DCOM. Il est également basé sur XML, ce qui le rend manipulable par des machines [38].

Un service web peut être publié dans des registres afin qu'il soit retrouvé par des utilisateurs potentiels. La publication de services se fait par UDDI (Universal Discovery Description and Integration) [38].

- Chaque service Web est défini par un fournisseur. Les fournisseurs de services déploient et publient leurs descriptions de services dans un registre que les clients peuvent localiser.
- Les clients trouvent leurs besoins de service en recherchant l'inscription au service Web.
- Après avoir localisé le service, le client extrait sa description du registre.
- Sur la base des informations définies dans la description du service, le client initie l'interaction [38].

1.5 Le concept des services web

Le concept des Services web s'articule actuellement autour des acronymes suivants :

1.5.1 WSDL

Web Services Definition Language pour la description des Services Web. C'est un format de schéma XML permettant de décrire un service Web en précisant les méthodes disponibles, les formats des messages d'entrée et de sortie et comment y accéder. Il définit un espace de travail extensible pour décrire des interfaces de Services Web. Comme d'autres technologies XML, WSDL est tellement extensible et a tellement d'options qu'il peut être difficile d'assurer la compatibilité et l'interopérabilité entre les différentes implémentations. L'interopérabilité est garantie si les expéditeurs et les destinataires des messages peuvent partager et comprendre le même fichier WSDL de la même manière [12].

Le fichier de description WSDL d'un Web Service contient un ensemble de définitions décrivant :

- L'interface du service c'est à dire les opérations que le service fournit, les formats des données et les protocoles utilisés.

- L'implémentation du service c'est à dire l'adresse applicative (URL ou URI) pour accéder au service [20].

WSDL permet de décrire :

- Types des données.
- Messages.
- Opérations.
- Binding

Ces descriptions doivent être exhaustives et précises, par exemple, l'API Amazon nécessite 1150 lignes WSDL pour 23 opérations, et Google nécessite 200 lignes pour 3 opérations.

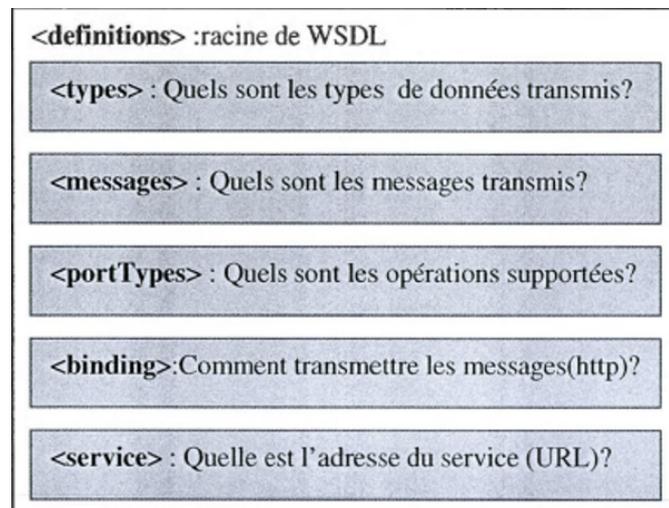


FIGURE 1.3: Structure d'un document WSDL.

Chacun des éléments principaux peut être spécifié dans un document XML et importé dans différentes combinaisons pour créer la description finale du service Web, ou ils peuvent tous être définis ensemble dans un seul document. La définition du type de données détermine la structure et le contenu du message [12].

Les informations contenues dans WSDL constituent la description du profil fonctionnel du service. Avec WSDL, le client peut invoquer le service par le biais de sa description abstraite (méthodes disponibles, paramètres d'entrée/sortie) et concrète (description des protocoles de communication et des points d'accès du service) [32].

```

<!-- Structure d'un WSDL -->
<definitions name="MonService"
targetNamespace="http://com.jmdoudoux.test.ws.monservice/"
xmlns="http://schemas.xmlsoap.org/wsdl/">
  <!-- Définitions abstraites -->
  <types> ... </types>
  <message> ... </message>
  <portType> ... </portType>

  <!-- Définitions concrètes -->
  <binding> ... </binding>
  <service> ... </service>
</definitions>

```

FIGURE 1.4: Le contenu du WSDL d'un service nommé MonService [16].

1.5.2 UDDI

Universal Description, Discovery and Integration, pour le référencement des services. D'une manière générale, l'UDDI est une interface XML regroupant d'autres services en ligne. C'est une norme d'annuaire de services Web appelée via le protocole SOAP. Il contient les registres Internet globaux regroupant les services et descriptions business depuis septembre 2000. D'autres services sont attendus ainsi que de nombreuses compagnies. Cette technologie doit être utilisée avec un pare-feu. Pour publier un nouveau service Web, il faut générer un document appelé Business Registry, il sera enregistré sur un UDDI Business Registry Node. Les nodes sont répliqués entre eux suivant un mécanisme analogue aux DNS [12].

Données du registre UDDI [12] :

- (a) Pages blanches :
 - Nom du business
 - Texte de Description (Liste de chaînes de caractères de plusieurs langues).
 - Informations (Noms, Numéros de téléphone, fax, sites Web, etc..).
- (b) Pages jaunes :
 - Services organisés en catégories de business.
 - La classification est implémentée comme des paires Non-valeur pour faire correspondre un identificateur à une page blanche.
- (c) Pages Vertes :
 - Ensemble d'informations décrivant comment faire du e-commerce avec ces compagnies.

- Processus marketing.
- Descriptions des services rendus.
- Programmation, Plateforme, Implémentation.
- Catégorie de services.

À l'origine, il existait des registres dits publics UDDI (comme ceux de Microsoft ou d'IBM) où n'importe qui pouvait devenir fournisseur ou client de services Web. L'omniprésence de ces registres a conduit UDDI à devenir la norme pour l'édition de services Web. En 2006, le nombre de services Web publiés a atteint 50 000. Néanmoins, UDDI n'a pas atteint son objectif de devenir un registre standard pour les services Web. En conséquence, la maintenance des registres UDDI publics tels que ceux de Microsoft et IBM a été suspendue. Le vrai succès d'UDDI se situe au niveau du registre privé. En fait, de nombreuses organisations utilisent la spécification UDDI pour implémenter leur propre registre de services Web [32].

1.6 Architecture d'un service web

Les services Web reprennent la plupart des idées et des principes du Web (HTTP, XML) et les appliquent aux interactions entre machines. Comme le World Wide Web, les services Web communiquent via un ensemble de technologies sous-jacentes qui partagent une architecture commune. Ils sont conçus pour fonctionner sur de nombreux systèmes développés et déployés indépendamment. Les technologies utilisées par les services Web sont **HTTP**, **WSDL**, **REST**, **SOAP** et **UDDI**.

1.6.1 SOAP

Simple Object Access Protocol pour les appels de services à distance par échange de messages XML. SOAP est un protocole de transmission de messages basé sur XML. C'est un standard recommandé par le World Wide Web Consortium qui définit un ensemble de règles pour structurer des messages qui peuvent être utilisés dans de simples transmissions unidirectionnelles, mais il est particulièrement utile pour exécuter des dialogues requête-réponse RPC (Remote Procedure Call). SOAP permet une communication en univers hétérogène, il est assez léger, simple et facile à déployer, extensible et ouvert mais ne garde pas la trace des requêtes envoyées des différents clients vers les services Web invoqués [4].

Structure d'un message SOAP : Un message SOAP est un document XML constitué d'une enveloppe, contenant un entête et le corps du message (voir figure ci dessous) :

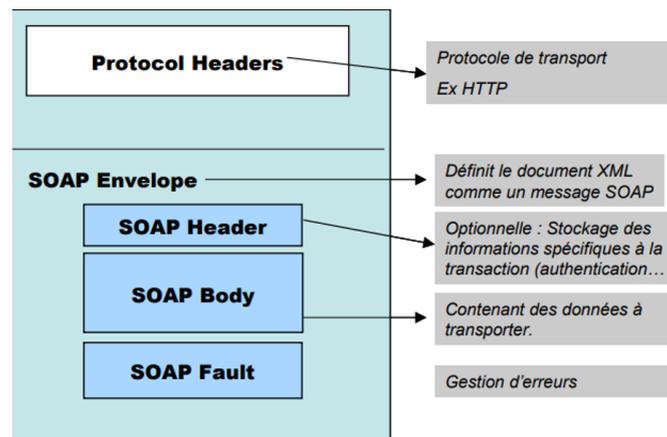


FIGURE 1.5: Message SOAP [10].

L'enveloppe SOAP : <SOAP-ENV : Envelope>

L'enveloppe est la racine du document XML contenant le message SOAP, marquée par la balise <Envelope>. La spécification impose que tous les attributs de cette balise et celles imbriquées soient explicitement associés à un namespace, de manière à supprimer toute ambiguïté. On peut citer un namespace fréquemment utilisé [10] :

SOAP-ENV associé à l'URI <http://www.w3.org/2001/06/soap-encoding>

L'entête SOAP : <SOAP-Env : Header>

La balise <Header> permet de passer dans le message des informations complémentaires sur ce même message. Cet élément est facultatif mais s'il est présent, il doit être le premier présent dans l'enveloppe SOAP du message. L'entête Header peut servir à plusieurs fins, Par exemple, il peut contenir des informations d'authentification de l'expéditeur, ou Un message n'est que le contexte d'une transaction pour l'une des étapes. Pour les couches En envoyant des virements qui ne fournissent pas d'adresse de retour (par exemple FTP), vous pouvez également utiliser des entêtes pour identifier l'expéditeur d'un message SOAP [10].

Le corps du message SOAP : < SOAP-Env : Body>

Le corps du message est constitué du seul élément Body, contenant un ou plusieurs sous éléments. Ces sous éléments sont les suivants :

- FAULT, indiquant une erreur ou une défaillance en réponse à une requête.
- Des données destinées au destinataire du message, à un format défini par les règles de codage SOAP.

Le corps du message peut effectuer des appels de procédure à distance (au sens OMG : un objet, méthodes et les valeurs de ces paramètres), des résultats

ou des messages d'erreur. Mais Le champ d'utilisation a été élargi, le corps du message est souvent utilisé pour simplement passer des Données structurées entre les applications [10].

1.6.2 REST

REST (Representational State Transfer) n'est pas un protocole, c'est un style d'architecture utilisé pour la conception de services Web. Il a été introduit en 2000 par Roy Fiedling dans sa thèse de doctorat. Cette méthodologie est une alternative au protocole SOAP. REST fournit un ensemble de contraintes architecturales qui permettent d'accéder et de manipuler les données.

L'adjectif anglais « RESTful » qualifie les systèmes qui respectent les contraintes architecturales de REST. Utilisé par les colosses de l'informatique (Google, Amazon, . . .), REST se base uniquement sur le protocole HTTP pour la communication client-serveur [6].

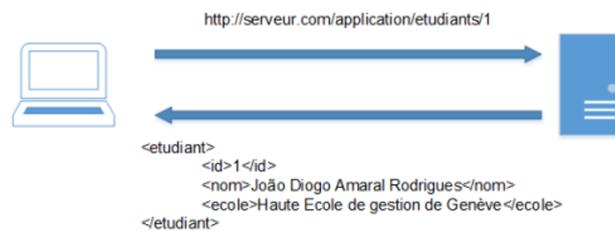


FIGURE 1.6: RESTful architecture.

Une bonne construction des URI est primordiale lors de la conception d'un service Web REST. Les liens (URI) dérivent de la hiérarchie des ressources et sont pour cette raison construits du plus général au plus particulier [6].

REST	SOAP
Style architectural	Protocole
Utilise XML ou JSON pour envoyer et recevoir des informations	Utilise WSDL pour communiquer entre le client et le serveur
Appelle le service via une URL	Utilise des appels de méthodes RPC
Uniquement le protocole HTTP	Principalement HTTP, mais peut utiliser SMTP, FTP, ...
Sécurité s'appuyant sur celle du protocole HTTP	Sécurité intégrée via « WS-Security » en plus de celle du protocole
Facile à mettre en place	Complexité, lourdeur de la mise en place et de la maintenance

TABLE 1.1: Comparaison REST et SOAP.

1.7 Fonctionnement des services web

Le schéma ci-dessous illustre le fonctionnement des services web (comment le client fait une recherche d'un service à partir de l'annuaire UDDI) :

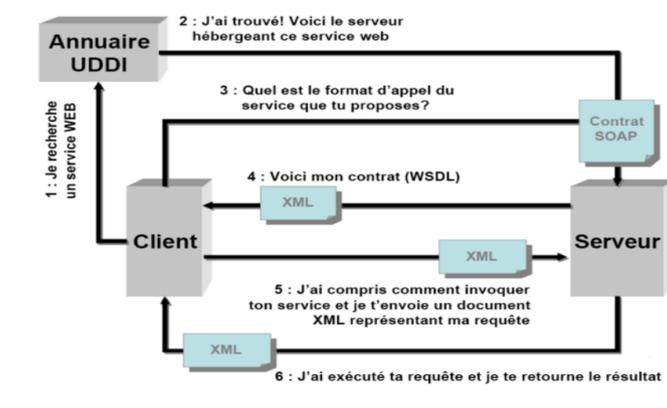


FIGURE 1.7: Fonctionnement des services web [15].

- En premier lieu on a la tâche du client qui fait envoyer une requête à l'annuaire du service (UDDI) pour trouver le service web dont il cherche.
- L'UDDI fait sa recherche et envoie par la suite une réponse au client qui indique quel serveur contient ce qu'il recherche..

- Le client envoie une requête au serveur pour obtenir la normalisation de ses données.
- Le serveur envoie son contrat WSDL en langage XML.
- Enfin, le client peut traiter les données dont il a besoin en rédigeant des requêtes XML.
- En dernier lieu le serveur exécute la requête du client et lui envoie le résultat voulu.

Tout cela est effectué grâce à un modèle en couches [41].

La 1re couche est l'invocation : elle décrit la structure des messages échangés par le client et le serveur. Elle s'effectue à l'aide des standards XML-RPC ou SOAP.

La 2eme est la découverte : C'est la phase de recherche et de localisation des données demandées par le client, le plus souvent via le protocole UDDI.

La 3eme est la description : qui stipule les paramètres des fonctions et les types de données des services web utilisés. Son protocole standard est le WSDL qui repose sur la notation XML.

1.8 Composition des services web

La composition permet de combiner des services pour former un nouveau service dit composé ou composite. L'exécution d'un service composé implique des interactions avec des services partenaires en faisant appel à leurs fonctionnalités [13].

1.8.1 Les principales approches de composition de services Web

- (a) Orchestration de services Web : L'orchestration de services Web résulte un nouveau service Web dit service Web composé, qui peut être défini comme l'agrégation de plusieurs autres services Web atomiques ou composés. 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. Le module exécutant le script d'orchestration de services Web est appelé un moteur d'orchestration [13].

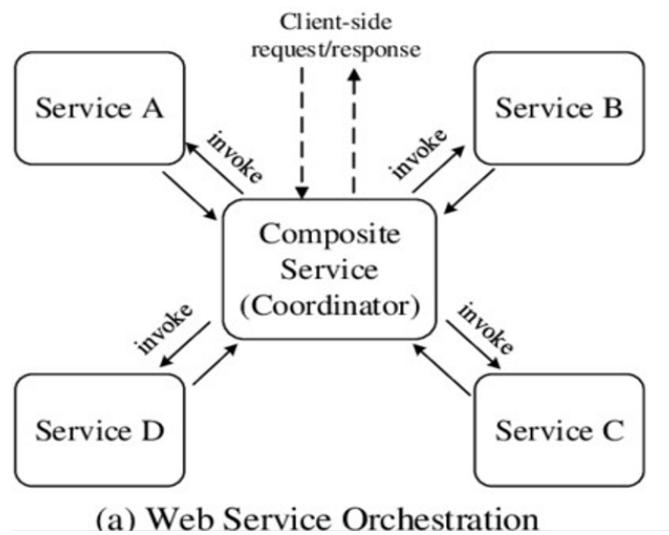


FIGURE 1.8: Orchestration de services Web [30]

- (b) Chorégraphie de services Web : La chorégraphie de services Web est une généralisation de l'orchestration qui consiste à concevoir une coordination décentralisée des services Web. Les services suivent alors le scénario global de composition sans point de contrôle central. En effet, l'exécution n'est pas régie de manière statique comme dans une composition de type orchestration [13].

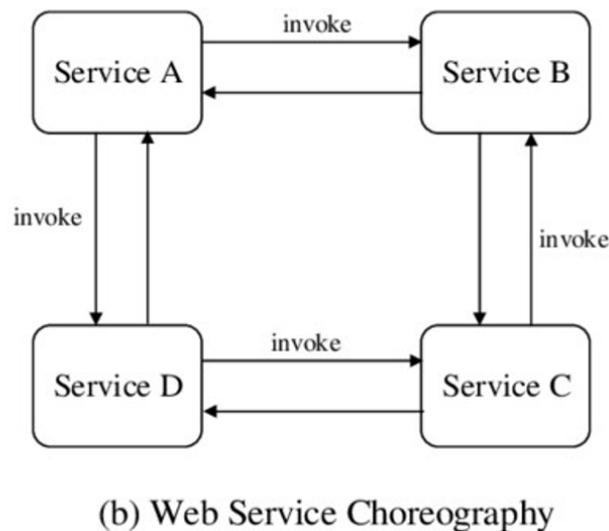


FIGURE 1.9: Chorégraphie Web Service [30].

1.8.2 Langages de composition de services Web

OWL-S : (Web Ontology Language for Web Services) Pour explorer l'approche sémantique de composition de services web, la communauté académique a proposé au début DAML-S (Darpa Agent Markup Language for Services), Aujourd'hui, OWL-S a remplacé DAML-S. On utilise l'ontologie OWL-S pour décrire les flux composés de services web en se basant sur l'utilisation des techniques de planification de l'intelligence artificielle, afin de chercher, orchestrer, composer et exécuter les services [50].

OWL-S étant une ontologie de services web basée sur OWL, qui offre aux fournisseurs des services web un ensemble noyau de balises permettant de décrire les propriétés de ces services web [5].

BPEL4WS : (Business Process Execution Language for Web Services) offre un langage pour les spécifications formelles des processus d'affaires et leurs interactions (BPEL4WS). Il prend en charge la modélisation de deux types de processus : les processus exécutables et les processus abstraits.

Un processus abstrait est un protocole d'affaire qui spécifie le comportement d'échange de messages entre plusieurs composants sans dévoiler leur comportement interne. Un processus exécutable spécifie l'ordre d'exécution entre les activités qui composent le processus (les activités), les partenaires du processus, et Les messages échangés entre ces partenaires et indiquent le comportement en cas d'erreurs ou d'exceptions [5].

WS-CDL : WS-CDL (Web Service Choreography Description Language) est un langage issu des efforts de standardisation du groupe de travail du W3C portant sur la chorégraphie de services Web. L'objectif de ce langage est de décrire les relations entre les services Web lors d'une composition de type chorégraphie [33].

1.9 Avantages et inconvénients des services web

La technologie des services Web est populaire et couramment utilisée car elle offre des avantages intéressants pour les utilisateurs des systèmes distribués [9].

- Grâce aux services Web, les coûts sont réduits par l'automatisation interne et externe des processus commerciaux.
- Les services Web réduisent le temps de mise en marché des services offerts par les diverses entreprises.

L'un des inconvénients provient toutefois de ces formats. XML est un format plutôt encombrant qui entraîne de grands paquets de données. Dans le cas de connexions réseau lentes, cela peut être source de problèmes [26]. Selon [9] la technologie des services web comporte aussi d'autres inconvénients dont :

- **Fiabilité** : Il est difficile de s'assurer de la fiabilité d'un service car on ne peut garantir que ses fournisseurs ainsi que les personnes qui l'invoquent travaillent d'une façon fiable.
- **Disponibilité** : Les services web peuvent bien satisfaire un ou plusieurs besoins du client. Seront-ils pour autant toujours disponibles et utilisables? Ça reste un défi pour les services web.
- **Confiance** : Les relations de confiance entre différentes composantes d'un service web sont difficiles à bâtir, puisque parfois ces mêmes composantes ne se connaissent même pas.
- **Problèmes de sécurité** : L'utilisation du protocole HTTP n'a pas que des avantages, car les normes de sécurité des services web laissent encore à désirer.

1.10 Service web et qualité de service

La qualité de service dans les services Web (QoS-SW) traite des aspects de qualité de service des services Web fournis aux utilisateurs. En SOA, les fournisseurs de services et les utilisateurs (consommateurs) de leurs services doivent pouvoir définir des paramètres et des aspects de la qualité de service, ce qui aura un impact considérable sur la publication, la découverte et la consommation de ces services. Côté client, il est important de les informer au préalable de l'état de la QoS avant son utilisation, quand le service Web est déjà ciblé [31].

Définition de QoS : La qualité de service est souvent assimilée à une distinction entre services, c'est-à-dire à des définitions de différentes classes de services. Mais c'est aussi garantir des services, et réserver des ressources pour cela. L'objectif principal de QoS est de fournir une hiérarchisation avec une bande passante dédiée, une gigue et un retard contrôlés (requis pour certains trafics en temps réel et interactifs) et des caractéristiques de perte améliorées. Le plus important est de s'assurer que la priorité donnée à un ou plusieurs flux n'entraîne

pas la défaillance d'autres flux dans le réseau [31].

[48], ont défini la QoS comme étant un ensemble de caractéristiques quantitatives et qualitatives d'un système multimédia, nécessaires pour atteindre la fonctionnalité requise par l'application.

1.11 Le courtier de QoS-SW

Le courtier QoS-SW joue dans une certaine mesure le rôle du catalogue de services Web, qui est comparable à l'annuaire UDDI mais est plus orienté QoS. Il guide les clients dans la sélection des services Web en fonction d'un ensemble de paramètres QoS. De plus, il permet l'authentification, la validation et le contrôle dynamique de la QoS. Le courtier de QoS-SW est un ensemble de composants qui exécutent des fonctions spécifiques à la QoS pour sélectionner des services Web, conformément aux normes requises par le client et aux paramètres QoS publiés par le fournisseur. Il fournit aux clients des informations pour les aider à prendre des décisions dans le choix des services Web. Le processus de prise en charge est basé sur les résultats de la vérification, de l'authentification et de la validation du service Web cible et de sa QoS[31].

1.12 Conclusion

La communication dans les services web peut s'effectuer sur diverses plateformes. Le client et le serveur n'ont pas besoin d'avoir grand-chose en commun pour que la communication fonctionne. Pour permettre cette dernière, les services Web ont recours à des formats standardisés compris par tous les systèmes.

Le prochain chapitre sera consacré à l'analyse des réseaux sociaux des services web (social computing) et les qualités sociale des services web.

Chapitre 2

Analyse des réseaux sociaux des services Web (Social Computing)

2.1 Introduction

Au 19e siècle, le terme « ingénierie sociale » a été utilisé pour la première fois par l'ingénieur des mines Frédéric Le Play « pour lutter contre la rénovation des ingénieurs, jugeant que le progrès technologique porte atteinte aux valeurs morales de la société et fait peser une menace de destruction sur les institutions nationales »[34].

Dans ce chapitre, nous introduisons le concept de "Social computing", ainsi que l'évolution des réseaux sociaux sur le web et la qualité des services web.

2.2 social computing

Avec l'avènement du Web 2.0, social computing a émergé comme l'un des sujets de recherche les plus importants récemment[28]. La terminologie de social computing a été proposée en 1994. Cependant, il y a diverses définitions pour social computing qui peut être tout type d'application de calcul qui utilise le logiciel comme un moyen de relations sociales[23]. social computing est un paradigme informatique nouveau et émergent qui implique une approche multidisciplinaire dans l'analyse et la modélisation des comportements sociaux sur différents médias et plateformes pour produire des applications intelligentes. De meilleures définitions de social computing par Wang et Fei-Yue ont définie social computing comme, "Computational facilitation of social studies and human social dynamics as well as the design and use of information and communication technologies that consider social context" [29].

La figure 1 montre une architecture de social computing, Le niveau inférieur de cette architecture illustre que le monde réel est composé du monde physique, du monde mental ainsi que du monde artificiel. Du point de vue pratique, il est facile d'obtenir une reconnaissance que la physique sociale, la psychologie sociale et social computing sont les produits certains des trois mondes, respectivement, et il y a un chevauchement significatif entre eux. Il est important de noter que cette reconnaissance nous permet d'utiliser pleinement le modèle de la société artificiel, et d'utiliser les ordinateurs comme des moyens expérimentaux sous un système parallèle pour vérifier et démontrer l'hypothèse de social computing dans le monde artificiel[23].

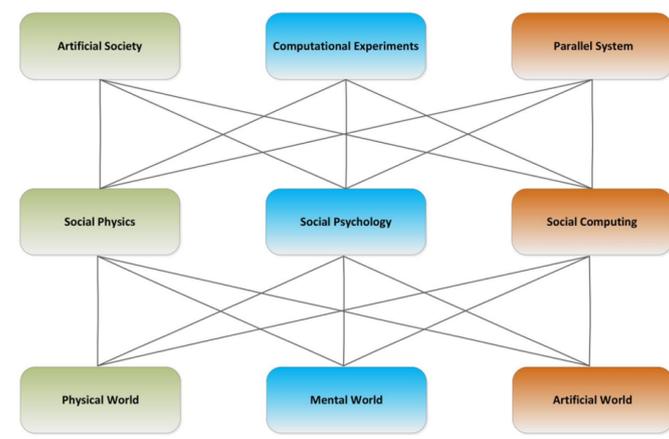


FIGURE 2.1: Architecture de social computing [23].

2.3 Du réseau social traditionnel au réseau social numérique

Auparavant, les individus se regroupaient à partir d'une aire géographique, de la tradition, de la religion ou des classes sociales. Sachant que les réseaux impliquent des communications interpersonnelles entre individus, les contraintes physiques et sociales limitaient le développement de réseaux substantiels. Or, les réseaux sociaux sur le Web, affranchis de ces contraintes, permettent aux gens de sortir des balises préétablies et de former des réseaux hétéroclites. Il est donc possible d'affirmer que les réseaux sociaux sur le Web catalysent le penchant des individus pour la formation de groupes et sous-groupes d'intérêts.

Le réseau social en ligne est donc une reproduction du réseau traditionnel, facilitant les regroupements entre individus partageant des intérêts communs, sans qu'ils soient pour autant entravés par les anciennes contraintes géographiques et sociales. Les individus se regroupent sur la base d'intérêts communs ou de valeurs partagées et le modèle de réseautage traditionnel s'est tout simplement transposé sur le Web[45].

2.4 Réseau social sur le web

Le terme « réseau social » fait référence à une plateforme Web 2.0 interactive et collaborative fournissant des outils pour permettre des rencontres à des fins amicales ou professionnelles, connexion, communication, diversité et interaction multi variée, échange et partage de contenus (vidéos, photos, blogs, messages publics ou

privés, liens hypertextes, etc.) entre tous les internautes s’y inscrivant et capables de favoriser la construction du réseau social de chacun d’eux. Une communauté virtuelle peut dès lors se tisser autour de la dynamique des échanges et de l’implication des membres. Un réseau social est un internaute qui peut créer, définir, Construire une identité numérique et valoriser sa réputation virtuelle. Et comme l’écrit Jean-Hugues Roca : « Si tu ne sais pas : demande, si tu sais : partage ! » telle pourrait être la devise d’un réseau social dédié à la gestion des apprentissages et à la formation [24].

2.4.1 Principes de fonctionnement

Les réseaux sociaux reposent sur des principes similaires [24] :

- L’accès aux fonctionnalités de la plateforme nécessite une inscription préalable des internautes intéressés par le réseau. Ainsi sur cette plateforme il a son propre espace et il pourra :
 - Créer son profil, c’est-à-dire fournir des informations personnelles (nom, prénom, adresse, . . . etc.), une photo qui identifie son centre d’intérêt, et une photo qui le représente dans le réseau.
 - Contenu du magasin à partager avec qui il veut.
 - Inviter des « amis » à le rejoindre, qu’il s’agisse de « vraies connaissances » ou de contacts virtuels dont les centres d’intérêt convergent avec les siens. Chaque contact qui accepte l’invitation s’ajoute au réseau de nouveaux abonnés.
- Un internaute peut filtrer l’accès à chaque contenu ou information qu’il met sur le web en spécifiant à chaque fois des destinataires (privés, semi-publics ou publics).
- folksonomy (un nuage de tags lié aux profils, aux informations et aux publications) est généralement associé à un site.
- Peut soumettre des commentaires sur le contenu.
- Le site Internet intègre un moteur de recherche dans les contenus/informations diffusés et les profils des utilisateurs.

2.4.2 Classification des réseaux sociaux

[7] distingue deux grandes catégories de réseaux sociaux numérique, les RSN de contact/relationnel, où les fonctions de contact et de connexion sont premières, et les RSN de contenu basés sur des activités spécifiques et des réseaux fonctionnels sont secondaires. Cible (sociale, de réseautage ou de navigation) pour différencier

les réseaux sociaux. Les réseaux sociaux sont par essence tournés vers le loisir, notamment en termes de communication et d'interaction entre les utilisateurs. Les sites de réseautage social tels que Facebook ou Myspace représentent ces types de réseaux axés sur les relations. Le but principal des réseaux sociaux est de trouver de nouveaux contacts pour développer votre réseau. Des sites comme LinkedIn ou Viadeo sont représentatifs de cette catégorie.

Enfin, les Réseaux Sociaux De Navigation sont surtout utilisés pour la recherche de contenus. Les individus, par l'intermédiaire de leur liste de contacts, peuvent alors avoir accès aux informations publiées et diffusées au sein de ces réseaux. Les sites de partage de liens sur internet, ou bookmarking, en sont des exemples pertinents[46].

2.4.3 Réseaux sociaux et apprentissage

L'activité des réseaux sociaux peut être une source d'apprentissage informel qui peut se faire à volonté, sans chercher. Les internautes adoptent de nouvelles idées après consultations sur les contenus partagés et lors d'échanges avec les autres membres du réseau. Ses contributions lui permettent de communiquer et de comparer les nouvelles connaissances qu'il génère, qui peuvent être une source d'apprentissage pour les autres. Les internautes ne sont pas toujours motivés par un désir conscient d'apprendre. Elle correspond davantage à la volonté de satisfaire des besoins. Comme nous l'avons vu, les besoins de Maslow (1943) se répartissent en deux catégories dans la représentation de sa pyramide (schéma) [24] :

- Besoins primaires : physiologiques et de sécurité.
- Besoins secondaires : appartenance, respect et réussite.

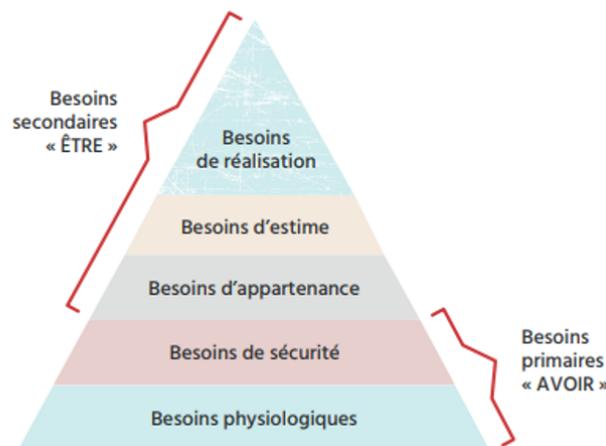


FIGURE 2.2: La pyramide des besoins selon Maslow [24].

2.5 La confiance sur les réseaux sociaux

En anglais, le mot confiance est traduit tantôt par confidence tantôt par trust. Ce dernier « **trust** » s'origine du mot allemand « trost » qui veut dire le « confort ».

En sociologie, l'allemand Luhman (1988) nous dit que la confiance fait référence à une croyance ferme nécessaire à la vie quotidienne, comme la confiance dans la fiabilité des freins de voiture, qui est la confiance dans le fabricant. Alors que "Trust" est un pari volontaire et risqué sur les actions des autres dans une situation donnée, comme croire que quelqu'un a la capacité de faire quelque chose de la façon dont on lui dit. Ainsi, Seligman (1983) affirme que la confiance se trouve dans un système dans lequel la familiarité entre les participants de ce dernier (système) et les sanctions qu'ils reçoivent en cas de violation représentent l'un des éléments clés de l'interaction entre les membres de ce système. D'autre part, Trust est un le sentiment qui surgit lorsqu'un tel système n'existe pas, par exemple en entrant dans un Négociant avec un partenaire pour la première fois[11].

La confiance sur les médias sociaux peut être illustrée par un certain nombre de qualités propres à la nature des informations échangées et à la perception de l'émetteur de l'information par les utilisateurs. Un certain nombre de contributions ont souligné que les types de contenus partagés et affichés par les marques étaient de nature à amener de la confiance dans le sens où l'information obtenue n'est accessible nulle part ailleurs. L'utilisation des médias sociaux par les fédérations pour faire état de leur engagement relatif à l'intérêt général a également un impact positif sur la propension accordée par l'utilisateur à leur faire confiance[25].

Dans le domaine informatique, selon Golbeck la confiance est défini comme étant l'engagement à croire au bon déroulement des prochains actions d'une autre entité [21]. Ainsi que Grandison nous a montré que l'acte de faire confiance se réalise dans un contexte spécifique et se définit par une croyance quantifiée quant aux habilités de l'entité qui est crue. Cette quantification peut être une échelle de valeurs ou une simple classification[22].

Enfin, Olmedilla définit la confiance dans le domaine des services Web comme une croyance mesurable que le fournisseur se comporte d'une manière digne de confiance pendant une période spécifique dans un contexte bien spécifique par rapport à un service Web[40].

2.6 Les réseaux sociaux des services Web

Selon Maamar et al. (2011b), on peut distinguer trois types de réseaux sociaux : réseau social de collaboration, réseau social de substitution et réseau social de compétition[36]. Ces services web sont établis en se basant sur trois relations sociales : compétition, substitution, et collaboration.

(a) **Réseau social de collaboration**

Un réseau social de collaboration est construit lorsqu'au moins la réalisation d'une composition est terminée. L'obtention d'un réseau social de collaboration nécessite un type de nœud d'entrée nommé le service web « Focus » qui est différent des autres, on a besoin aussi des arêtes (liens) ce qui sortent du service web « Focus » sont unidirectionnels allant vers d'autres service web, en combinant leurs capacités respectives, les services Web peuvent travailler ensemble pour gérer les demandes complexes des utilisateurs. Par conséquent, le service Web contrôle son propre réseau de collaborateurs, il peut donc décider s'il aime travailler avec des pairs en fonction de son expérience passée, car il peut également recommander des pairs[37].

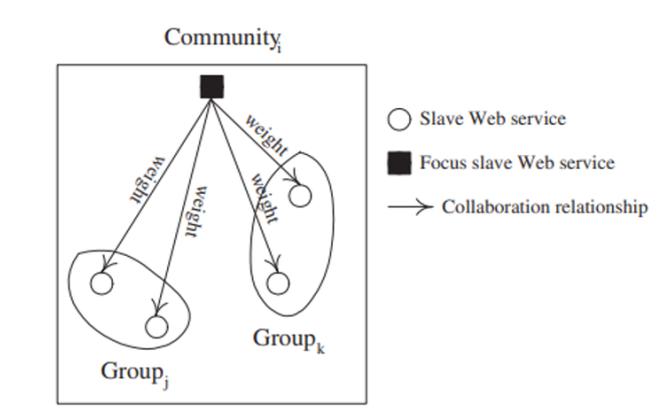


FIGURE 2.3: Réseau social de collaboration [37].

- (b) **Réseau social de compétition.** Dans ce réseau les services web ont des caractéristiques similaires donc ils sont tous en concurrence (compétition). Construire un réseau social de compétition nécessite un type de Nœud, uniquement pour représenter les services Web et des arêtes bidirectionnelles pour représenter la relation de concurrence. Dans un réseau social de compétition, il n'y a pas de nœud spécifique qui joue le rôle de point d'entrée dans ce réseau. Les services Web se font concurrence pour fournir des fonctionnalités similaires. Leur nature non fonctionnelle les distingue lorsqu'il s'agit de satisfaire les besoins non fonctionnels des utilisateurs. En conséquence, le

service Web apprend à connaître son propre réseau de concurrents, de sorte qu'il peut Tente d'améliorer ses propriétés non fonctionnelles par rapport à d'autres pairs[37].

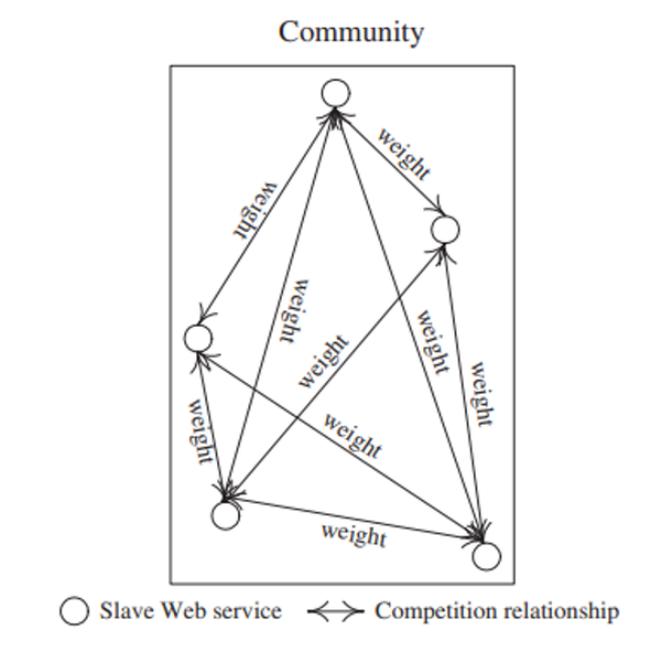


FIGURE 2.4: Réseau social de compétition [37]

- (c) **Réseau social de substitution** Le réseau social de substitution est identique à un réseau social de compétition, car les services web offrent tous les mêmes fonctionnalités à la seule différence dans la construction du réseau les arêtes ne sont pas tous bidirectionnels. Et ce réseau a une haute disponibilité des services web.

2.7 Les qualités sociales de services Web

Coopération : La qualité de coopération d'un service web fait référence au nombre de fois où ce dernier est invité par son binôme à participer à la composition. Plus le service web est sollicité plus la qualité de coopération est meilleure. Pour évaluer la qualité de coopération nommée $Coop(i, j)$ d'un service web i vers j (2.1), nous considérons le nombre de fois où le service j a appelé le service i dans une composition supplémentaire [37].

$$Coop(i, j) = \frac{\sum JS(i, j)}{\sum TP(i, j)} \quad (2.1)$$

$JS(i, j)$ est le nombre total de participation des deux service web i et j dans une composition supplémentaire. $TP(i, j)$ est le nombre total de participation du service web i dans des composition.

Égoïsme : Un service Web se comporte de manière égoïste s'il ne montre pas une attitude positive envers d'abord, son service Web maître direct et ensuite, d'autres pairs situés dans sa communauté ou d'autres communautés. Par exemple, un service Web égoïste reçoit des réponses positives constamment des autres paires lorsqu'il les sollicite pour une substitution, cependant en retour ce dernier refuse constamment de leur venir en aide en déclinant leurs requêtes [37].

Pour évaluer la qualité d'égoïsme nommée $Ego(i, j)$ d'un service web i vers j (2.2), nous considérons le nombre de fois où le service web i a refusé la requête du service web j dans un réseau social de collaboration.

$$EGO(i, j) = \frac{\sum DR(i, j)}{\sum JS(i, j)} \quad (2.2)$$

$DR(i, j)$ est le nombre de requête du service Web i que le service Web j a refusé.

Honnêteté : L'honnêteté d'un service Web est la différence entre les valeurs des paramètres publiés ou convenues de la QoS et les valeurs réelles observées de ces paramètres Qos , l'équation (2.3) évalue la qualité d'honnêteté nommée $H(i, j)$ [1].

$$H(i, j) = 1 - \frac{\sum |AQoS_j^i - OQoS_j^i|}{k} \quad (2.3)$$

$AQoS_j^i$ est la valeur de QoS convenue entre le service Web i et j .

$OQoS_j^i$, est la valeur de QoS observée du service Web i par le service Web j .

k est le nombre de fois que le service Web j a invoqué le service Web i .

2.8 Conclusion

À mesure que les technologies de social computing deviennent omniprésentes, de nombreux défis sont apparus. Les défis incluent la surcharge d'informations,

la prise de décision, la confidentialité, la sécurité, la confiance et le risque. Pour cela notre travail va nous permettre de bien choisir un web service qui est confiant et réponds à nos besoin. Nous présenterons dans le prochain chapitre un état de l'art des articles les plus récents qui ont traité le problème de la composition des services Web et l'extension du fichier WSDL.

Chapitre 3

Etat de l'art : Composition des services web et extension wsdl

3.1 Introduction

Dans ce chapitre, nous présentons les différents travaux qui ont traité le problème de la composition des services Web et l'extension du fichier WSDL.

3.2 La composition des services web et extension WSDL

Carlisle et Sharon, présentent un framework pour implémenter la sécurité pour les services Web en utilisant l'extension UDDI et WSDL. Leurs travaux comprennent la sécurité de l'UDDI lui-même, la sécurité des transactions de services Web et les liens avec l'infrastructure existante en dehors de l'UDDI. Ils ont identifiés les extensions du schéma pour UDDI et WSDL, ainsi que les extensions de la sécurité du mécanisme de publication et de découverte lui-même[2].

UDDI, fournit de nombreux outils de sécurité nécessaires pour sécuriser le registre et ses données lorsqu'ils sont utilisés dans des domaines fermés tels qu'une entreprise. Cependant, certaines extensions sont nécessaires pour prendre en charge le déploiement à grande échelle de plusieurs registres dans différents domaines. Ces extensions sont nécessaires pour faciliter les schémas d'authentification et les politiques d'autorisation standardisés, pour garantir que les utilisateurs ont l'authenticité des données par rapport au fournisseur de registre, et pour garantir que l'UDDI peut faire partie intégrante du cadre de sécurité global spécifié pour les services Web en général[2].

EL Hog Ben Djemaa et Amous, ont défini les services web comme étant un composant logiciel utilisé pour exposer via Internet. Il assure l'interaction entre des applications et des systèmes hétérogènes. L'explosion du nombre d'internautes a conduit à une grande variété de profils. Toutefois, les services Web existants offrent le même résultat quel que soit le profil utilisateur[18]. C'est pourquoi ils souhaitent développer des solutions adaptatives qui répondent aux différentes étapes du cycle de vie des services. Dans un travail précédent, ils se sont intéressés à l'étape de la conception et ils ont proposé et détaillé une extension UML (Adaptive Web Service Unified Modeling Language) appelée AWS-UML (Adaptive Web Service Unified Modeling Language) qui décrit les différents profils d'utilisateurs autorisés au niveau du métamodèle selon les principes et les normes MDA (Model Driven Architecture).leur dernier travail réalise une extension de la norme WSDL (Web Service Description Language) utilisée pour décrire les fonctionnalités du service proposé. Leur proposition s'appelle Adaptive Web Service Web Service Description Language (AWS-WSDL). L'extension ajoutée propose d'apporter des

réponses pour atteindre les insuffisances du langage UML et du standard WSDL pour fournir aux consommateurs des services Web adaptatifs[18].

Dai et Wang, ont monté l'importance des services web dans le domaine de l'internet des objets (IOT).le web service (WS) peut jouer un rôle de représentation de chose réelle dans le système d'information. Pour décrire un objet complet, son service doit contenir des informations fonctionnelles et non fonctionnelles, car ce sont des attributs importants et les clés pour distinguer les choses les unes des autres. En tant que norme de WS, le langage WSDL (Web Service Description Language) ne peut pas décrire un objet avec toutes ses informations, car WSDL ne fournit aucune méthode pour décrire l'aspect non fonctionnel des services alors qu'il se concentre uniquement sur les parties fonctionnelles[14]. Dai et Wang ont proposé une extension souple de WSDL pour décrire les attributs non fonctionnels. Basé sur les principes et les normes de l'architecture MDA (Model Driven Architecture), le processus consiste en une extension et une transformation de méta modèle WSDL entre les modèles UML et la syntaxe XML.ils se sont basé sur une extension dont le mappage d'un objet physique à un service Web sera plus précis et plus complet. De cette façon, WSDL peut décrire complètement un objet physique sous la forme d'un service[14].

Le langage WSDL (Web Services Description Language) est une spécification XML destinée à décrire les fonctionnalités fournies par les services Web et largement adoptée dans le secteur informatique. Toutefois, les types de message définis dans WSDL pour l'interopérabilité sont insuffisants pour décrire des données complexes (telles que des données géographiques) en utilisant des types de données classiques dans de nombreux scénarios d'application. Par exemple, les données géographiques (telles qu'une image numérique capturée par des technologies de télédétection) sont généralement décrites et situées simplement sous la forme d'une chaîne URL. Étant donné que des métadonnées importantes pour la description et la vérification sont manquantes, ni les fournisseurs ni les utilisateurs n'ont de bonnes expériences avec les services SIG actuels décrits dans WSDL. Gao, You, Gui, et Wu présentent une extension WSDL spécifique pour prendre en charge la description de géodonnées complexes. Cette extension WSDL est plus pragmatique et efficace que le WSDL standard dans les scénarios spécifiques à GIService [30]. Une méthode d'extension WSDL a été présentée pour permettre la description de géodonnées complexes afin de mieux soutenir les services SIG. Le procédé proposé fournit un modèle uniforme de type de données géographiques pour décrire des données échangées entre un service SIG et ses clients lors de l'appel d'un service SIG. Le modèle de définition du type de géodonnées spécifie les données à traiter par GIServices. En étendant le WSDL

standard, les données géographiques impliquées dans un service SIG peuvent être décrites avec précision. De plus, à partir du WSDL étendu, les utilisateurs peuvent être informés des fonctionnalités et des géodonnées d'entrée/sortie de GIServices [30].

Les services Web peuvent être les plus populaires et les plus puissantes technologies de développement logiciel dans le monde actuel des logiciels. Pourtant, il apporte aux développeurs de logiciels et testeur beaucoup de défis aussi. Ceci est principalement dû au manque d'informations fournies par le fichier WSDL d'un service Web. À partir du fichier WSDL, nous ne pouvons pas obtenir les informations utiles pour les tests, telles que les informations de dépendance. Pour résoudre ce problème, W. T. Tsai, Ray Paul, Yamin Wang, Chun Fan, et Dong Wang ont proposé et pratiqué une extension WSDL pour prendre en charge ce type de description d'informations. Leur travail comprend quatre types d'extension : dépendance d'entrée-sortie, séquence d'appel, description fonctionnelle hiérarchique et spécifications de séquence simultanée [47]. Ces extensions ont été utilisées pour tester des applications utilisant des services Web sur les plates-formes d'internet. En utilisant ces extensions, nous pouvons facilement récupérer les informations utiles nécessaires pour les tests de services Web. Cela peut réduire considérablement l'effort et le coût pour effectuer ces tâches et rendre possible l'automatisation de ces tâches [47].

Avec le développement rapide des applications Web Services et la popularité croissante, il existe de nombreux services Web fonctionnellement similaires disponibles, ce qui a rendu la sélection des services Web basés sur des propriétés fonctionnelles est insuffisante. QoS devient un critère important dans la sélection des meilleurs services Web. Ling Jiang, Tingna Liui, et Desheng Liui proposent une méthode de recherche des services Web fondée sur des facteurs de qualité de service objectifs et subjectifs. Les facteurs Objective-QoS ont été ajoutés en extension WSDL, qui devrait être offerts par le fournisseur de services. Les facteurs subjectifs de QoS viennent du point de vue des utilisateurs, représentent les expériences réelles des utilisateurs de services. Un système prototype a été mis au point pour mettre en œuvre les facteurs de QoS objective et de QoS subjective [27].

Leur méthode a bien était une méthode de recherche de service web en introduisant la QoS objective et la QoS subjective. Les exigences Objective-QoS sont utilisées pour répondre aux besoins de l'utilisateur du service Web en filtrant le service non qualifié, tandis que Subjective-QoS peut être utilisé comme référence pour les utilisateurs ultérieurs. Les résultats expérimentaux montrent que la méthode réalisée est non seulement efficace, mais fonctionne aussi bien pour

des scénarios complexes. La méthode de découverte de service Web prise en charge par QoS possède une grande précision de découverte ainsi qu'un temps de réponse acceptable. Pendant ce temps, le système a également quelques problèmes, comme il ne garantit pas la validité des résultats de recherche avec beaucoup de liens morts des services Web[27].

WSDL est utilisé pour décrire l'interface d'un service, au format XML. L'interface décrit les propriétés fonctionnelles ainsi que les propriétés non fonctionnelles. D'autre part on s'intéresse à la spécification de «critères» comme une propriété non fonctionnelle d'un service Web. Pour cela, Parimala N et Anu Saini, présentent une extension WSDL à X-WSDL. Afin d'ajouter des informations de critères, l'extension du schéma WSDL est réalisée en ajoutant un nouvel élément 'criteriaservice', qui est disponible dans le nouvel espace de noms. En utilisant cet élément 'criteriaservice', il est possible de spécifier les critères avec un service dans un document X-WSDL. Le document WSDL est également étendu par l'ajout de nouveaux attributs « nom de critère » et « description » à l'élément de service. En utilisant cette extension, il est possible de spécifier les critères avec le service dans le document X- WSDL. Les critères sont spécifiés par l'utilisateur lors de l'appel d'un service. En conséquence, ils fournissent un soutien pour découvrir un service plus approprié en fonction de ses besoins [43].

Dans [43], Parimala N et Anu Saini ont expliqué leur approche à l'aide de la version WSDL 1.1. Cependant, l'extension est également applicable à la version WSDL 2.0. Ces deux versions sont différentes lors de la description de l'interface. WSDL1.1 contient le <porttype> pour décrire l'interface. Ici, les éléments XML sont utilisés à la place des messages d'entrée et de sortie. L'interface utilise le nom de l'opération directement dans la description de WSDL2.0. L'élément <port> dans WSDL1.1 est remplacé par l'élément <endpoints> dans WSDL 2.0.

Les deux éléments <binding> et <service> n'ont pas changé, ils restent identiques dans WSDL1.1 et WSDL2.0. Dans ce cas, nous étendons l'élément <service>. Par conséquent, nous pouvons utiliser n'importe quelle version pour incorporer les informations de critères avec le WSDL. Leur prochaine étape est de fournir des services composables basés sur des critères. Même si WSDL1.1 est compatible avec BPEL et que WSDL2.0 est compatible avec Choreography Description Language, il nous sera possible d'explorer la composition en utilisant à la fois l'orchestration et la chorégraphie. Le système complet sera évalué à l'aide d'exemples concrets [43].

De nos jours, les entreprises reconnaissent le besoin d'être orientées vers le

client en fournissant un service supérieur pour satisfaire les besoins des clients. Mais à mesure que les clients et leurs besoins deviennent de plus en plus divers, des coûts et une complexité inutiles s'ajoutent inévitablement aux opérations. Les prestataires de services ont découvert la nouvelle frontière de la concurrence : Personnalisation collaborative. Cette approche se déroule en trois étapes : d'abord, dialoguer avec les clients individuels pour les aider à articuler leurs besoins ; deuxièmement, identifier l'offre précise qui répond à ces besoins ; et troisièmement, créer des produits personnalisés pour eux. Les services Web déployés sur le Web sont accessibles à une base d'utilisateurs plus large. Les services Web sont conçus et sous-traités pour répondre aux besoins d'un grand nombre d'utilisateurs. Souvent, plusieurs personnalisations de la fonctionnalité de base sont nécessaires pour répondre aux besoins de plusieurs ensembles d'utilisateurs. Cela force le fournisseur de services à déployer plusieurs services Web personnalisés pour chaque ensemble d'utilisateurs, ce qui entraîne une augmentation du coût de l'infrastructure et de la maintenance. Étant donné que plusieurs versions de services Web personnalisés sont déployées plusieurs fois à différentes URL, il est difficile et plus coûteux de maintenir, mettre à jour et sauvegarder ces services et leurs données [8].

Banati, Bedi, et Marwaha ont réalisé un travail qui a comme objectif de réduire les efforts et les coûts résultant de ces multiples versions des Services Web. Ils ont fait une extension WSDL et WSDL-T à WSDL-TC afin de réduire les coûts en maintenant les différentes versions personnalisées collaboratives des opérations du service Web en un seul déploiement. L'approche gère également le contrôle d'accès de ces opérations à leurs groupes respectifs. WSDL-TC étant une extension de WSDL-T est capable de gérer les versions de chaque opération personnalisée qui a résulté en raison des changements dans leurs besoins commerciaux sur une période de temps. Leur travail présente un exemple de service Web de réservation d'hôtel pour présenter l'approche WSDL-TC. WSDL-TC simplifie également la tâche des administrateurs de services Web car ils doivent gérer une seule instance au lieu de plusieurs instances d'un service Web. En outre, les services WSDL-TC déployés dans l'environnement cloud peuvent contribuer à atteindre un plus grand degré de multi-tenancy et à réduire davantage les coûts pour les producteurs de services [8].

En conclusion, en utilisant WSDL-TC, il est possible de personnaliser n'importe quelle version valide de l'artefact, disponible à un moment donné pour n'importe quel client. Cela permet de créer une fonctionnalité personnalisée dans un service pour chaque entité (ensemble d'utilisateurs). L'un des principaux avantages de cette approche est l'avantage opérationnel. Comme tout le code de l'application se trouve au même endroit, il est beaucoup plus facile et moins

cher de maintenir, de mettre à jour et de sauvegarder le service et ses données. En outre, dans le cas de WSDL, si un correctif doit être appliqué pour toute correction de bogue ou gestion de la sécurité, il doit être appliqué à tous les déploiements, ce qui rend le processus long et des efforts supplémentaires sont nécessaires. Ici, si le patch fait parti des opérations qui ne sont pas modifiées ou personnalisées, il doit être appliqué à un emplacement unique, quel que soit le nombre d'entités qui utilisent le service. Un autre avantage réside dans les ressources système globalement inférieures requises pour exécuter le déploiement unique par rapport aux déploiements multiples. Il n'est pas nécessaire d'avoir une version dédiée des ressources et de l'infrastructure de service pour chaque entité (ensemble d'utilisateurs). Il s'agit d'une nette amélioration de l'utilisation des ressources, d'un gain de temps et d'un coût global réduit [8].

3.3 Tableau comparatif

Le tableau suivant compare les articles qui traitent les différents problèmes liés à la sélection et la composition des services Web ainsi que l'extension de la description WSDL du service web selon plusieurs critères d'évaluation : Extension, Implémentation, QoS.

Articles	Extension	Implémentation	QoS	Confiance
[2]	UDDI/WSDL	+	-	-
[18]	WSDL	-	-	-
[14]	WSDL	-	-	-
[30]	WSDL	-	-	-
[47]	WSDL	+	-	-
[27]	WSDL	+	+	-
[43]	WSDL	+	-	-
[8]	WSDL	+	-	-

TABLE 3.1: Tableau comparatif.

3.4 Synthèse

Ce chapitre présente une synthèse des recherches effectives dans l'état de l'art. Cette synthèse permet de mettre en évidence les éléments que nous devons apporter pour faciliter la tâche de conception.

Dans un premier temps, les études réalisées dans ces travaux ,portant sur les travaux concernant la sécurité [2], solutions adaptatives qui répondent aux différentes étapes du cycle de vie des services [18] , une extension de WSDL pour décrire les attributs non fonctionnels [14] , la description de géodonnées complexes afin de mieux soutenir les services SIG [30], manque d'informations fournies par le fichier WSDL d'un service Web [47], la sélection des services Web [27], l'ajout des informations de critères [43], réduire les efforts et les coûts résultant de ces multiples versions des Services Web [8]. Ces travaux nous permet de dresser un tableau comparatif selon des critère déjà mentionnés (extensions, implementations et les qualités de service). Finalement, nous avons remarqué que le critère qualité de service n'a pas été fréquemment utilisé, ainsi que la confiance n'a jamais été utilisée auparavant. Par conséquent, la présente enquête est la première à s'en prévaloir.

3.5 Conclusion

Dans ce chapitre nous avons présenté un état de l'art sur le problème de la composition des services Web et l'extension du fichier WSDL, dans lequel nous avons mentionné les différents travaux les plus récents qui traitent ce problème.

Nous avons pu ensuite construire un tableau récapitulatif pour résumer tous les articles cités dans ce chapitre. Ce tableau nous a permis de nous positionner par rapport aux travaux déjà réalisés.

Le prochain chapitre sera consacré à l'étude et le développement de notre application à savoir les outils utilisés, les bases de données, l'environnement de développement. . .

Chapitre 4

Conception et réalisation

4.1 Introduction

Dans ce dernier chapitre nous allons décrire les principaux axes de la réalisation de notre application qui consiste à proposer une méthode pour la sélection de service web à base de confiance.

Pour arriver à notre objectif, nous avons proposé une méthode basée sur les qualités sociales, on va d'abord calculer les valeurs des trois qualités mentionner (égoïsme, coopération, honnêteté) ensuite on détermine la valeur de la confiance « Trust » selon le besoin de l'utilisateur. Nous présentons ensuite la base de test utilisé pour évaluer l'environnement, après nous montrons les différentes expérimentations menées, et nous discutons les résultats obtenus.

4.2 Environnement de développement

Notre système est développé sous l'environnement suivant :

- Système d'exploitation : windows 10
- CPU : intel® core TM i5-4200U @ 1.60Hz 2.30 GHz
- Mémoire : 4Go

4.3 Langage de programmation

Le langage de programmation utilisé pour le développement de notre système est le PYTHON, Python est un langage de programmation inventé par Guido van Rossum. Sa première version est sortie en 1991.

Python est un langage de programmation open source, multiplateforme et orienté objet. Grâce à des bibliothèques spécialisées, Python peut être utilisé dans de nombreuses situations, telles que le développement de logiciels, l'analyse de données ou la gestion d'infrastructures. Par conséquent, il ne se concentre pas uniquement sur la programmation Web comme le langage HTML.

En tant que langage de programmation interprété, Python permet au code de s'exécuter sur n'importe quel ordinateur. Disponible pour les programmeurs débutants et experts, et il rend la création de programmes rapide et facile.

4.4 Implémentation

On a utilisé Python version 3.10.4 sous l'environnement Visual Studio Code. En premier lieux on a installé Conda qui est un gestionnaire de packages et un

système de gestion d'environnement open source, multiplateforme et indépendant du langage. En effet, la gestion de paquets est d'une grande importance pour le langage Python. On a utilisé l'extension Jupyter et Jupyter Keymap sur vs code qui émulent l'environnement Jupyter Notebook sur le navigateur. Jupyter (anciennement IPython Notebook) est un projet opensource qui vous permet de combiner facilement du texte Markdown et du code source Python exécutable sur un canevas appelé un notebook. Visual Studio Code prend en charge l'utilisation native de blocsnotes Jupyter et les fichiers de code Python.

4.4.1 Conception

Pour montrer le déroulement de notre algorithme qui calcule la valeur de confiance « Trust », on a utilisé l'organigramme aussi appelé organigramme de programmation qui est la représentation visuelle d'un algorithme. Il montre les enchaînements de décisions et d'opérations à faire pour un algorithme donné.

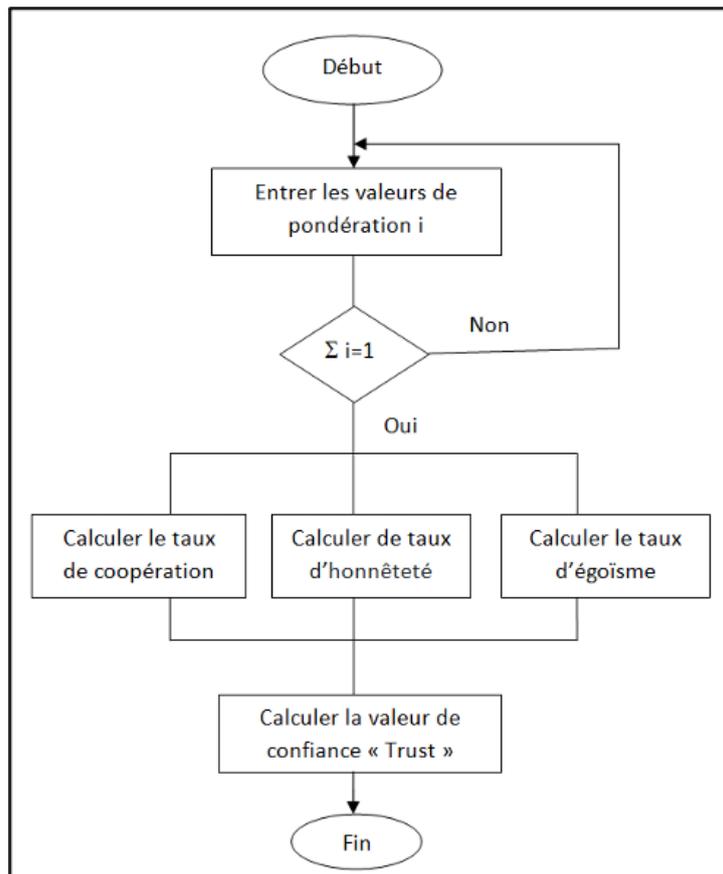


FIGURE 4.1: Organigramme.

4.4.2 Les fonctions

- (a) Configuration :
- `drop_database(database_name)` : permet la suppression d'une base de données.
 - `create_database(database_name)` : la création d'une base de données.
 - `connect_to_database(database_name)` : se connecter a une base de données.
 - `create_table(table_name, fields, database_name)` : creation d'une table.
- (b) L'importation des données du fichier xml vers la base de données :
- `import_data(file_name)` : permet d'importer les données d'un fichier XML vers une base de données.
- (c) Extraction de données :
- `query_database(query, one = false)` : pour récupérer les données de la base.
 - `get_all_services()` : pour lister les id des services.

```
[1, 2, 5, 8, 3, 11, 12, 14, 13, 4, 15, 16, 6, 7, 10, 9, 23, 81, 32, 33]
```

FIGURE 4.2: La liste des services.

- (d) Egoïsme :
- `calculate_total_number_of_participation()` : calcule le nombre totale de participation des services .
 - `calculate_total_rejections()` : calcule le nombre de fois où le service a était rejeté.
 - `calculate_egoism()` : calcule le taux d'égoïsme pour chaque service, en applicant l'équation (2.2) .

```
print(calculate_egoism())
[0.125, 0.14285714285714285, 0.3333333333333333, 0.3333333333333333,
0.6, 0.0, 0.0, 0.0, 0.0, 0.2857142857142857, 0.5, 0.0,
0.3333333333333333, 0.3333333333333333, 0.16666666666666666, 0.0,
0.0, 1.0, 0.5, 1.0]
```

FIGURE 4.3: Résultat de la fonction `calculate_egoism()`.

(e) Honnêteté :

- *calculate_abs_diff_aqos_oqos(requester)* : calcule la valeur absolue de la différence entre les deux qualités sociales mentionner.
- *calculate_number_of_invoqued_times()* : calcule le nombre de fois où le service a été invoqué.
- *calculate_honnetete()* : calcule le taux d'honnêteté pour chaque service, en appliquant l'équation (2.3).

```

print(calculate_honnetete())
[0.36999999999999999, 0.5028571428571428, 0.9963333333333333,
0.9998333333333334, 1.0, -1, -1, -1, -1, 0.855, 0.9700000000000001,
-1, 0.999, 0.965, 1.0, 1.0, -1, 1.0, 1.0, 1.0]
    
```

FIGURE 4.4: Résultat de la fonction *calculate_honnetete()*.

(f) Coopération :

- *calculate_total_acceptance()* : calcule le nombre de fois où la requête du service a été accepté.
- *calculate_cooperation()* : calcule le taux de coopération pour chaque service, en appliquant l'équation (2.1).

```

print(calculate_cooperation())
[2.6666666666666665, 1.5555555555555556, 6.0, 1.8, 2.5, 2.0, 2.0,
1.0, 2.0, 1.4, -1, -1, 3.0, 3.0, 1.2, 1.0, -1, -1, 2.0, -1]
    
```

FIGURE 4.5: résultat de la fonction *calculate_cooperation()*.

(g) Trust :

- *calculate_trust(coop_weight, egoism_weight, honnetete_weight)* : calcule le taux de confiance pour chaque service en entrant des paramètres de pondérations pour chaque qualité.

Pour les pondérations (0.5, 0.2, 0.3), on obtient les résultats suivant :

```

[1.4693333333333332, 0.9572063492063492, 3.3655666666666667, 1.2666166666666667, 1.67, 1.0, 1.0, 0.5, 1.0,
1.013642857142857, 0.391, 0.0, 1.8663666666666667, 1.8561666666666665, 0.9333333333333332, 0.8, 0.0, 0.5,
1.4000000000000001, 0.5]
    
```

FIGURE 4.6: Taux de confiance pour chaque service.

Pour cette fonction l'utilisateur doit entrer trois valeurs de pondération selon le besoin. pour le taux de coopération, pour le taux d'égoïsme, μ pour le taux d'honnêteté. Avec une condition $\alpha + \beta + \mu = 1$.

Sa formule est comme suit :

$$Tr = \alpha * Coop(i, j) + \beta * Ego(i, j) + \mu * H(i, j)$$

- (h) A la fin de notre programme on a mis une fonction qui permet l'ajout automatique des données (taux d'égoïsme, d'honnêteté, de coopération, et la valeur de confiance) de chaque service en extension au document descriptif WSDL.

```

1  {?xml version="1.0" encoding="utf-8"}
2  <definitions name="InteropLab" xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:w="http://schemas.xmlsoap.org/wsdl/"
3  <message name="ServiceP01a1182959_Request"
4  <part name="P19a8620411" type="xsd:string"/>
5  </message>
6  <message name="ServiceP01a1182959_Response"
7  <part name="P82a3034588" type="xsd:string"/>
8  </message>
9  <portType name="ServiceP01a1182959Port">
10 <operation name="ServiceP01a1182959">
11 <input message="tns:ServiceP01a1182959_Request"/>
12 <output message="tns:ServiceP01a1182959_Response"/>
13 </operation>
14 </portType>
15
16 <trust>
17 <value>0.4036666666666666</value>
18 </trust>
19 </definitions>

```

FIGURE 4.7: Description WSDL du service « S32 ».

4.4.3 Collection de données

Pour évaluer les performances de notre application, nous avons entrepris une série d'expérience sur une base de données proposée par WS-DREAM [49] et que nous avons reconstruit et adapté selon nos données d'application car on avait besoin que de deux qualités de services. Notre base nommée *swt_db* contient une table *transactions* :

- *id* : identifiant de chaque service.
- *requester* : contient l'identifiant de service demandeur.
- *requested* : contient l'identifiant de service demandé.
- *result* : prend les deux valeurs, 0 si la requête est rejeté sinon 1.
- *QoS* et *qos* : deux qualités de services.

id	requester	requested	▲ 1	result	QoSO	qos	competency
19	12	1	0	0.5	0.5	1	
5	12	1	1	0.124	0.124	1	
1	1	2	1	0.2	0.2	1	
23	1	2	0	0.8	0.7	1	
24	1	2	0	0.46	0.4	0	
22	1	2	0	0.5	0.5	0	
4	11	2	0	0.345	0.35	0	
25	1	2	1	0.9	0.1	1	
10	11	2	1	0.56	0.1	0	
12	16	3	0	0.63	0.1	0	
9	2	3	0	0.9	0.1	1	
8	15	3	0	0.67	0.7	0	
6	14	3	1	0.234	0.3	0	
3	2	3	1	0.7	0.7	1	
28	6	4	0	0.109	0.11	0	
16	13	4	0	0.145	0.145	0	
26	2	4	1	0.9	0.1	1	
7	13	4	1	0.56	0.1	1	
14	6	4	1	0.209	0.209	0	
11	2	4	1	0.87	0.87	1	
27	4	5	1	0.98	0.11	1	
30	2	5	0	0.34	0.36	0	
15	7	5	0	0.23	0.3	1	
31	2	6	0	0.34	0.12	0	
32	2	7	0	0.12	0.3	0	

TABLE 4.1: Table des transactions.

le tableau suivant fourni quelques exemples de transaction :

	TA	$Coop(i, j)$	Requested(K)	$ aqos - oqos $	$H(i, j)$	TP	TR	$Ego(i, j)$
1	3	2.66	2	1.26	0.37	8	1	0.125
2	18	1.55	7	3.48	0.5	28	4	0.14
3	2	2.5	5	0	1	5	3	0.6
4	5	1.4	6	0.87	0.855	7	2	0.28
5	1	6	3	0.011	0.967	6	2	0.33
6	1	3	1	0.001	0.999	3	1	0.33
7	1	3	2	0.07	0.965	3	1	0.33
8	5	1.8	6	0.001	0.9998	9	3	0.33
9	4	1	4	0	1	4	0	0
10	5	1.2	6	0	1	6	1	0.16
11	1	2	0	0.465		2	0	0
12	1	2	0	0		2	0	0
13	1	2	0	0.46		2	0	0
14	1	1	0	0.066		1	0	0
15	0		1	0.03	0.97	2	1	0.5
16	0		0	0.53		1	0	0
23	0		0	0.01		1	0	0
32	1	3	2	0	1	2	1	0.5
33	0		1	0	1	1	1	1
81	0		1	0	1	1	1	1

TABLE 4.2: Exemples de transactions.

- TA : total acceptance.
- TP : total participation.
- TR : total rejection.

Remarques :

- D'après les tests appliqués sur notre base on remarque qu'elle n'est pas suffisante, si on prend l'exemple du service 33 qui a une valeur d'égoïsme 1 (plafond) alors qu'il a participé qu'une seule fois et sa requête était refusée contrairement aux autres services qui ont eu plusieurs participations. Le service 33 n'a pas eu la chance pour augmenter ces performances en diminuant la valeur de son égoïsme.

- Dans le cas où le dénominateur prend la valeur 0 (les cases rouges) on le rend a -1 pour ne pas avoir des formes indéfinies.
- Si on compare les valeurs qu'on a calculé nous même (Table 4.2) avec les valeurs obtenues dans l'application, celle de l'application sont plus précise que nos résultats.

4.5 Expérimentation

Le graphe suivant représente une comparaison entre les services avec leurs taux des trois qualités sociales (Table 4.2). On remarque que pour chaque service plus le taux de coopération est élevé plus la qualité d'égoïsme se diminue.

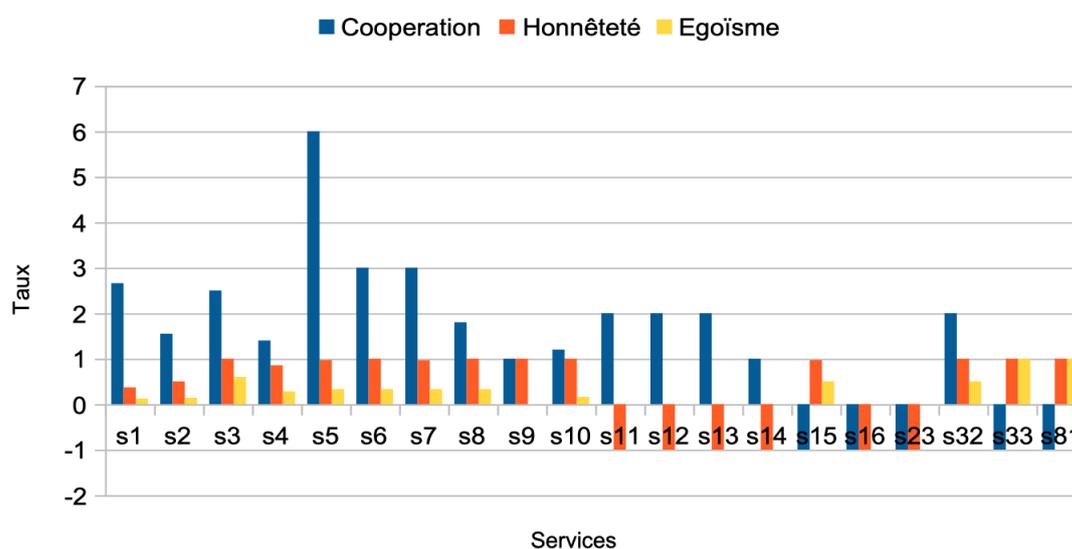


FIGURE 4.8: Comparaison des services.

Dans le graphe suivant, on remarque que la valeur de confiance s'augmente et se diminue en fonction des deux qualités : coopération et honnêteté.

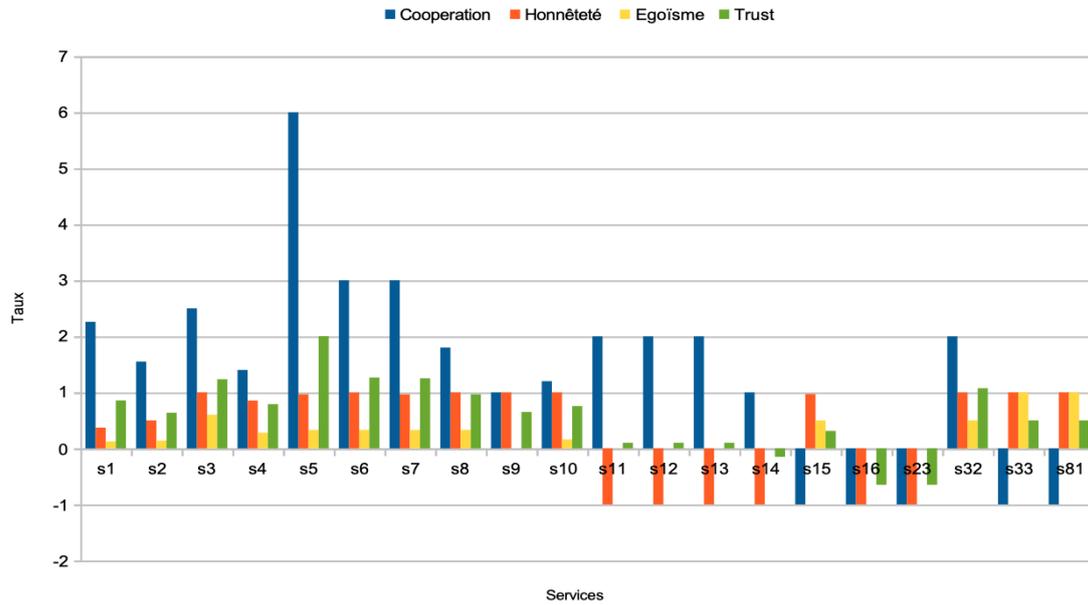


FIGURE 4.9: Taux des trois qualités avec la valeur de confiance « Trust ».

4.6 Conclusion

Dans ce chapitre nous avons détaillé les étapes de la réalisation de notre application qui avait pour but la construction d'une méthode qui facilite la sélection de service web à base de confiance, nous avons ainsi discuté l'expérimentation mise en place afin d'évaluer la fiabilité de notre application et démontrer que la méthode proposé peut contribuer a l'assistance de l'utilisateur.

Conclusion générale et perspectives

La sélection des services web est l'un des enjeux les plus importants dans l'architecture orientée service. C'est aussi l'une des étapes les plus importantes dans le processus de composition qui a été un domaine de recherche actif au cours des dernières années. Ce projet de fin d'études, nous a permis de concevoir une application qui a pour but de déterminer la valeur de la confiance "trust", cette valeur on a l'intégrer dans la description du service web WSDL pour améliorer la sélection et la composition des services web.

Pour atteindre notre objectif, nous proposons une méthode basée sur les qualités sociales, nous allons d'abord calculer la valeur des trois qualités évoquées (égoïsme, coopération, honnêteté) puis déterminer la valeur de confiance en fonction des besoins de l'utilisateur.

Quelques améliorations et extensions peuvent être envisagées pour enrichir la méthode suggéré :

- La prise en considération de la sémantique pour une meilleure sélection de services web sémantique.
- La prise en compte de la problématique de la composition des services web qui reste une tâche très complexe et présente certaines nombre de défis.
- intégrer la confiance dans des systemes collaboratifs sémantiques pour la sélection et la recommandation des services web.
- l'utilisation du web mashup qui fournit aux applications des moyens supplémentaires utiles et économiques pour consommer l'information seule à partir d'une source qui est augmentée et amenée à de nouveaux niveaux une fois combinée avec une ou plusieurs sources connectées supplémentaires. Il permet aux données provenant de plusieurs sources d'interagir entre elles.

Bibliographie

- [1] Naziha Abderrahim et al. *Contribution des réseaux sociaux dans l'ingénierie des services Web*. PhD thesis, 2015.
- [2] Carlisle Adams and Sharon Boeyen. Uddi and wsdl extensions for web service : a security framework. In *Proceedings of the 2002 ACM workshop on XML security*, pages 30–35, 2002.
- [3] Benmounah Kenza Adel Belbekri. Schéma de Fonctionnement d'une architecture SOA. https://www.researchgate.net/figure/Schema-de-Fonctionnement-dune-architecture-SOA_fig1_336210121, 2019. [Online ; accessed 23-Mars-2022].
- [4] Mohamed Ben Ahmed. Classification, recherche et composition de services web. 2009. Conference Paper Université de la Manouba.
- [5] Aniss Alkamari. Composition de services web par appariement de signatures. 2008.
- [6] João Diogo Amaral Rodrigues. *Architecture et mise en place de services web dans un environnement sécurisé garantissant la confidentialité des informations d'un centre de contrôle de qualité externe*. PhD thesis, Haute école de gestion de Genève, 2016.
- [7] Fallery B. et Rodhain F Aurélie G. L'apparition des medias sociaux dans l'e-grh : Gestion de la marque employeur et e-recrutement. 2011.
- [8] Hema Banati, Punam Bedi, and Preeti Marwaha. Wsdl-tc : Collaborative customization of web services. In *2012 12th International Conference on Intelligent Systems Design and Applications (ISDA)*, pages 692–697. IEEE, 2012.
- [9] B. S. M. Bekkouche Amina. Composition des services web sémantiques À base d'algorithmes génétiques. 2012. Master's thesis, Université Abou-bekr Belkaid Tlemcen.
- [10] Youcef Belaid. Cours :nfe107 urbanisation amp ; architecture des systèmes d'information. 2008-2009.

-
- [11] Meriem Billami and Chaib Bounoua. La confiance comme institution et mode de coordination des activités informelles-étude exploratoire auprès des entreprises privées de tlemcen. *Revue d'études sur les institutions et le développement*, 3(1) :34–61, 2016.
- [12] Sebastien Gastaud Cyrielle Lablanche, Florens Seine. Les services web. 2005. Rapport de TE/Université de Nice-Sophia Antipolis.
- [13] Lifa Siham Daha Hanane. Une approche formelle pour la composition des services web. 2015. web-Mémoire de fin d'étude (Master Académique) université Echahid Hamma - EL-OUED.
- [14] Changying Dai and Zhibin Wang. A flexible extension of wsdl to describe non-functional attributes. In *2010 2nd International Conference on E-business and Information System Security*, pages 1–4. IEEE, 2010.
- [15] Mandrioli Damien. Les web services. <http://deptinfo.unice.fr/twiki/bin/view/Minfo03/DocFinal#Fonctionnement>, 2004. [Online; accessed 23-Mars-2022].
- [16] Jean-Michel Doudoux. Les services web de type Soap . <https://www.jmdoudoux.fr/java/dej/chap-service-web.htm>, 2019. [Online; accessed 22-april-2022].
- [17] Nicola Dragoni. A survey on trust-based web service provision approaches. In *2010 Third International Conference on Dependability*, pages 83–91. IEEE, 2010.
- [18] Chiraz El Hog, Raoudha Ben Djemaa, and Ikram Amous. Aws-wsdl : a wsdl extension to support adaptive web service. In *Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services*, pages 477–480, 2011.
- [19] DEHIMI Nour El Houda. Architectures orientées services (soa). 2019.
- [20] Lécué Freddy. Composition de web services sémantiques, rapport de stage master-r en informatique, l'école nationale supérieure des mines de st-etienne. 2005.
- [21] J. A. Golbeck. Computing and applying trust in web-based social networks. 2005. PhD thesis, University of Maryland.
- [22] T. W. A. Grandison. Trust management for internet applications. 2003. Technical report.
- [23] Fei Hao, Doo-Soon Park, and Zheng Pei. When social computing meets soft computing : opportunities and insights. *Human-centric Computing and Information Sciences*, 8(1) :2, 2018.
- [24] P. Henri, F. et Plante. Réseau social et apprentissage. ted 1272 : Les tic et l'apprentissage en milieu de travail. 2019.

-
- [25] G. Huron, D. amp ; Spieth. Médias sociaux et déterminants de la confiance. pages 74–88, 2019. Les fédérations sportives françaises sur Facebook. *Management international*,23(3),.
- [26] Ionos. Services Web : services entre machines. <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/services-web/>, 2020. [Online ; accessed 27-February-2022].
- [27] Ling Jiang, Tingna Liu, and Desheng Liu. Objective and subjective qos factors supported web service search method based on extended wsdl. In *2015 23rd International Conference on Geoinformatics*, pages 1–4. IEEE, 2015.
- [28] Irwin King. Introduction to social computing. In *International Conference on Database Systems for Advanced Applications*. Springer, 2010.
- [29] Li J. King, I. and K. T Chan. A brief survey of computational approaches in social computing. 2009. In *International Joint Conference on Neural Networks*, Atlanta, Georgia, USA.
- [30] Zhenlong Li, Zhipeng Gui, Barbara Hofer, Yan Li, Simon Scheider, and Shashi Shekhar. Geospatial information processing technologies. In *Manual of Digital Earth*, page 201. Springer, Singapore, 2020.
- [31] lkbal Taleb. Qualité de service dans les services web. 2006.
- [32] C. Lopez-Velasco. Composition de web services sémantiques, rapport de stage master-r en informatique, l’école nationale supérieure des mines de st-etienne. 2008. Sélection et composition de services Web pour la génération d’applications adaptées au contexte d’utilisation. PhD thesis, université Joseph Fourier.
- [33] Céline Lopez-Velasco. *Sélection et composition de services Web pour la génération d’applications adaptées au contexte d’utilisation*. PhD thesis, Université Joseph-Fourier-Grenoble I, 2008.
- [34] Molina Christine Lyet Philippe. Nouvelle figure de l’encadrement dans l’intervention sociale. pages 3–12, 2012. *Revue Vie Sociale* n°1/2011. Cédias.
- [35] Zakaria Maamar, Hakim Hacid, and Michael N Huhns. Why web services need social networks. *IEEE Internet Computing*, 15(2) :90–94, 2011.
- [36] Zakaria Maamar, Philippe Thiran, and Jamal Bentahar. Web services communities : From intra-community cooperation to inter-community competition. In *E-business applications for product development and competitive growth : emerging technologies*, pages 333–343. IGI Global, 2011.
- [37] Zakaria Maamar, Hamdi Yahyaoui, Erbin Lim, and Philippe Thiran. Social engineering of communities of web services. In *2011 IEEE/IPSJ International Symposium on Applications and the Internet*, pages 100–109. IEEE, 2011.

-
- [38] Hacene Mechedou. Un environnement de composition de services web/memoire presente comme exigence partielle de la maitrise en informatique par hacene mechedou ;[directeur de recherche,]. 2007.
- [39] Tarek Melliti. *Interopérabilité des services Web Complexes : Application aux systèmes multi-agents*. PhD thesis, Paris 9, 2004.
- [40] Rana O. F. Matthews B. Olmedilla, D. and W Nejd. Security and trust issues in semantic grids. in dagstuhl seminar on semantic grid : The convergence of technologies,. page 52–71, 2005.
- [41] Oracle. Le fonctionnement d’un web service. Available at <https://www.oracle.com/fr/cloud/definition-web-service/>, 2020.
- [42] Mike P Papazoglou. Service-oriented computing : Concepts, characteristics and directions. In *Proceedings of the Fourth International Conference on Web Information Systems Engineering, 2003. WISE 2003.*, pages 3–12. IEEE, 2003.
- [43] N Parimala and Anu Saini. Web service with criteria : Extending wsdl. In *2011 Sixth International Conference on Digital Information Management*, pages 205–210. IEEE, 2011.
- [44] Mozghan Tavakolifard and Kevin C Almeroth. Social computing : an intersection of recommender systems, trust/reputation systems, and social networks. *IEEE Network*, 26(4) :53–58, 2012.
- [45] Op.Cit Teixeira M. page 32, 2018.
- [46] Mike Thelwall. Social network sites : Users and uses », advances in computers, vol.76,. 2009.
- [47] Wei-Tek Tsai, Ray Paul, Yamin Wang, Chun Fan, and Dong Wang. Extending wsdl to facilitate web services testing. In *7th IEEE International Symposium on High Assurance Systems Engineering, 2002. Proceedings.*, pages 171–172. IEEE, 2002.
- [48] Kerhervé B. von Bochmann G. Vogel, A. and J Gecsei. Distributed multimedia and qos : A survey. page 59, 1995.
- [49] WS-DREAM. S-DREAM : Towards Open Datasets and Source Code for Web Service Research, 2017. [Online ; accessed 20-Mai-2022].
- [50] Weishi Zhang and Xiuguo Zhang. Modeling service interactions using kahn process network. In *2006 Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT’06)*, pages 203–208. IEEE, 2006.