وزارة التعليم العالي و البحث العلمي

BADJI MOKHTAR-ANNABA UNIVERSITY
UNIVERSITE BADJI MOKHTAR-ANNABA

جامعة باجي مختار – عنابـــة

Faculté des Sciences de l'Ingéniorat **Année 2017-2018**

Département d'Informatique

THESE

Présentée en vue de l'obtention du diplôme de
Doctorat en Sciences

INTEGRATION FLEXIBLE DES SERVICES : SPECIFICATION ET PRISE EN COMPTES DES PROFILS UTILISATEURS

Option
Sciences et Technologies de l'Information et de la Communication

Par

Mr. Abdelmalek Metrouh

Directeur de Thèse

Mr. Farid MOKHATI Professeur Université d'Oum El Bouaghi

Soutenue publiquement le 26/09/2018

DEVANT LE JURY

Président :	Mr. Djamel MESLATI	Professeur	Université Badji Mokhtar-Annaba
Examinateurs	Mlle Fadila ATIL	Professeur	Université Badji Mokhtar-Annaba
	Mr. Djamel BENMERZOUG	MCA	Université de Constantine 2
	Mr. Abdelkrim BOURAMOUL	MCA	Université de Constantine 2

Dédicace

A mon père, qu'aucune dédicace ne saurait exprimer
l'estime, le dévouement et le respect
que j'ai toujours eu pour lui.

A ma mère, symbole de la bonté par excellence,
source de tendresse et l'exemple du dévouement
qui n'a pas cessé de m'encourager
et de prier pour moi.

A ma femme pour ses sacrifices, son soutien moral,
sa gentillesse sans égal, son profond attachement
m'ont permis de réussir mes études
de post-graduation.

A mes chers enfants et
à toute ma famille.



Remerciements

Quelques mots en préambule de cette thèse, qui met un point d'orgue à de longues années riches et intenses.

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

La première personne que je tiens à remercier est mon directeur de thèse, Mr. Farid MOKHATI, Professeur à l'Université d'Oum El Bouaghi, pour ses conseils qu'il m'a prodigué, la patience, la confiance qu'il m'a témoigné, ont été déterminants dans la réalisation de ce travail de recherche. Qu'il trouve dans ce travail un hommage vivant à sa haute personnalité.

Mes remerciements s'étendent également aux président et membres de jury. A Mr. Djamel Meslati, Professeur à l'Université Badji Mokhtar – Annaba, qui m'a fait l'honneur de présider le jury de ma thèse ainsi que Mlle Fadila ATIL, Professeur à l'Université Badji Mokhtar – Annaba, Dr. Djamel Benmerzoug et Dr. Abdelkrim Bouramoul, Maîtres de conférence à l'Université Constantine 2, pour l'honneur qu'ils m'ont fait pour leur participation à mon jury de thèse en qualité d'examinateurs, pour le temps consacré à la lecture de cette thèse, et pour les suggestions et les remarques judicieuses qu'ils m'ont procurées.

À mes parents pour leur soutien, leurs prières, leurs encouragements et leurs rêves, pour avoir rester avec moi dans toutes les difficultés et me montrer la bougie de gentillesse pour chaque abstraction difficile de la vie.

À ma femme qui mérite le plus grand merci pour ses encouragements sans fin, patience constante et prières.

À mes enfants Koussai, Mazen, Abderraouf et en particulier ma petite fleur Ayat Errahman pour avoir passer leurs meilleurs moments de vie en dehors de moi tout en me permettant de me concentrer sur cette thèse.

Tébessa, le 10 Avril 2018

Abdelmalek Metrouh

ملخص:

يبقى التركيب الديناميكي أو التكامل أحد الأهداف الرئيسية لتكنولوجيا خدمات الويب. تهدف هذه الرسالة إلى اقتراح نهج مبتكر لتركيب خدمات الويب الديناميكية، يعتمد فيها على السمات الوظيفية وغير الوظيفية لخدمات الويب و كذلك التفضيلات الفردية للمستعملين. في هذا النهج ، ترتبط تركيبة خدمات الويب بالشبكات الاجتماعية وتأخذ في الاعتبار سجل استدعاء خدمات الويب الخاص بكل مستخدم. الشبكات الاجتماعية لخدمات الويب هي هياكل ديناميكية تستخدم للحفاظ على التفاعلات بين هذه الأخيرة من أجل تحديد وإنشاء خدمات الويب التي ترتبط ارتباطًا وثيقًا بتفضيلات المستخدم. نستخدم مفهوم مجتمع خدمات الويب في شبكة اجتماعية لخدمات الويب للحد بشكل كبير من مساحة البحث لهذه الأخيرة. يتم إنشاء هذه المجتمعات عن طريق المشاركة المباشرة لموفري خدمات الويب.

كلمات مفتاحية: خدمة ويب؛ تمثيل المعرفة؛ تركيب؛ اختيار؛ ملف تعريفي للمستخدم؛ مجموعة خدمات الويب؛ شبكة اجتماعية.

Abstract

Dynamic composition or integration remains one of the key objectives of Web services technology. This thesis aims to propose an innovative approach of dynamic Web service composition based on functional and non-functional attributes and individual preferences. In this approach, Web services composition is linked to social networks and takes into account the users' service invocation history. The social networks of Web services are dynamic structures used to maintain interactions between Web services in order to select and compose Web services that are more tightly related to user's preferences. We use the concept of Web services community in a social network of Web services to reduce considerably the search space of Web services. These communities are created by the direct involvement of Web services providers.

Keywords: Web service; knowledge representation; composition; selection; user profile; community; social network.

Résumé

La composition dynamique ou l'intégration reste l'un des objectifs clés de la technologie des services Web. Cette thèse vise à proposer une approche innovante de la composition dynamique des services Web basée sur les attributs fonctionnels et non fonctionnels et les préférences individuelles. Dans cette approche, la composition des services Web est liée aux réseaux sociaux et prend en compte l'historique d'invocation des services par les utilisateurs. Les réseaux sociaux de services Web sont des structures dynamiques utilisées pour maintenir les interactions entre les services Web afin de sélectionner et de composer des services Web plus étroitement liés aux préférences de l'utilisateur. Nous utilisons le concept de communauté de services Web dans un réseau social de services Web pour réduire considérablement l'espace de recherche de ces derniers. Ces communautés sont créées par l'implication directe des fournisseurs de services Web.

Mots clés: service Web; représentation de la connaissance; composition; sélection ; profil utilisateur; communauté; réseau social.

Table des matières

Introduction générale 1
1. Contexte 1
2. Problématique et objectifs 3
3. Motivations 7
3.1 Scénario 1 7
3.2 Scénario 2 8
4. Contributions 8
5. Plan de la thèse 9

Chapitre 1 : Vue d'ensemble des technologies clés 11
1. Introduction 12
2. Service Oriented Computing (SOC) 13
3. Service Web 14
3.1 Définition des services Web 14
3.2 Architecture des services Web (SOA) 15
3.3 Propriétés d'un services Web 18
3.2.1 Propriétés fonctionnelles 18
3.2.2 Propriétés comportementales 18
3.2.3 Propriétés non fonctionnelles 19
3.4 Technologies des services Web 19
3.4.1 Langage de description des services Web (WSDL) 21
3.4.2 Protocole d'accés aux objets simples (SOAP) 23
3.4.2.1 Structure d'un message SOAP 24
3.4.2.2 Mode de traitement 25
3.4.2.3 Protocole de liaison 25
3.4.2.4 Avantages et incovénients de l'utilisation de SOAP 26
3.4.3 Description universelle, protocole de découverte et d'integrétation (UDDI) 27
3.5 Caractéristiques des services Web 29
3.6 Types de services Web 30
3.7Avantages de l'utilisation des services Web 32
4. Paramètres de qualité de services (QoS) 33
5. Personnalisation des services Web 35

Table des matières

6. Réseaux sociaux sur le Web 37
7. Systèmes de recommandation et filtrage collaboratif 39
8. Conclusion 40

Chapitre 2 : Sélection des services Web 43
1. Introduction 44
2. Défintion 46
3. Sélection des services Web sociaux 47
   3.1 Approches basées sur la réputation 48
   3.2 Approches basées sur la recommandation 51
   3.3 Approches basées sur le renvoi 54
4. Sélection des services Web et défis des approches basées sur la recommandadtion 55
   4.1 Problème de démarrage à froid 55
   4.2 Problème de confidentialité 56
   4.3 Evolutivité 57
   4.4 Capture des préférences et des évaluations des utilisateurs 57
5. Etude comparative des différentes approches de sélection des services Web sociaux 59
6. Conlusion 61

Chapitre 3 : Composition des services Web 63
1. Introduction 64
2. Modèles de composition des services Web 65
   2.1 Orchestration 66
   2.2 Chorégraphie 67
   2.3 Coordination 68
   2.4 Modèle de composant 69
3. Exigences de la composition des services Web 69
   3.1 Automatisation 70
   3.2 Dynamicité 70
   3.3 Qualité de service (QoS) 71
   3.4 Evolutivité 71
   3.5 Préférences des utilisateurs 72

3.6 Indépendance au domaine 72
4. Différentes approches de composition automatique des services Web 72
4.1 Approches orientées Workflow 73
4.2 Approches basées sur les modèles 74
4.3 Approches mathématiques 78
4.4 Approches orientées intelligence artificielle 81
4.5 Approches basées sur la recommandation 84
5. Discussion 88
6. Conclusion 90

Chapitre 4 : Intégration flexible des services Web: une nouvelle approche sociale personnalisée 93
1. Introduction 94
2. Architecture proposée pour la composition des services Web sociaux personnalisés 95
3. Communautés de services Web 97
4. Construction du réseau social de services Web 100
4.1 Paramètres qualité de services (QoS) considérés 100
4.2 Fonction objective 101
4.3 Associations de collaboration pour la composition 103
4.4 Associations de collaboration pour la substitution 104
4.5 Associations de recommandation pour la compostion 104
5. Sélection des services Web atomiques pour la composition 105
6. Composition des services Web sociaux personnalisés 108
6.1 Construction des préférences utilisateur 108
6.1.1 Attributs statiques 108
6.1.2 Attributs acquis 108
6.2 Algorithme pour la composition des services Web sociaux personnnalisés 109
7. Tolérances aux pannes dans le cadre de l'approche proposée 109
6.1 Associations de recommandation pour la substitution 110
6.2 Algorithme de sélection des services Web pour la substitution 111
8. Implémentation et évaluation 113

Table des matières

8.1 Approche de composition des services Web personnalisés 113
8.2 Algorithme de sélection des services Web pour la substitution 118
9. Conclusion 119

Conclusion générale 122
1. Bilan 122
2. Perspectives 126

Publications découlant de cette thèse 127

Bibliographie 128

Liste des figures

1.1 Modèle de base de l'architecture SOA 17
1.2 Architecture SOA (Papazogloo, 2008) 20
1.2 Les éléments de WSDL (Bajaj et al., 2006) 23
2.1 Découverte, sélection et composition des services Web 44
3.1 Modèles de composition des services Web 66
4.1 Architecture pour la composition des services Web sociaux personnalisés 96
4.2 Exemple de définition de communautés de services Web dans un réseau social de services Web 99
4.3 Exemple de graphe de substitution d'un service Web 104
4.4 Associations de recommandation et de collaboration pour la composition 105
4.5 Exemple de réseaux sociaux de services Web 106
4.6 Associations de recommandation et de collaboration pour la substitution 111
4.7 Temps vs Nombre de sommets dans l'arbre associé à la communauté extraite *cT* 115
4.8 Temps Vs Nombre de sommets dans le graphe de collaboration *CG* 116
4.9 Temps Vs Nombre d'arcs dans le graphe de collaboration *CG* 116
4.10 Temps Vs Nombre d'arcs dans le graphe *RCG* 116
4.11 Temps Vs Nombre de sommets dans le graphe *G* associé au réseau social de services Web 117
4.12 Temps Vs Nombre d'arêtes dans le graphe *G* associé au réseau social de services Web 117
4.13 Temps Vs Nombre de sommets dans le graphe de substitution 119

Liste des tables

2.1 Défis abordés dans les approches basées sur les réseaux sociaux 61
3.1 Comparaison des différents travaux issus de différentes approches de composition des services Web 88
4.1 Temps nécessaire pour renvoyer l'ensemble des services Web sélectionnés pour la composition 115
4.2 Temps nécessaire pour renvoyer les services Web sélectionnés (substituts) 119

Introduction générale

Sommaire

1. Contexte 1
2. Problématique et objectifs 3
3. Motivations 7
   3.1. Scénario 1 7
   3.2. Scénario 2 8
4. Contributions 8
5. Plan de la thèse 9

1. Contexte

Le monde de l'Internet tel que nous le connaissons aujourd'hui a subi des changements profonds. Il est devenu une infrastructure de communication essentielle pour nos performances économiques et notre bien-être social. Avec un nombre d'internautes, fin 2016, d'environ 3,9 milliards (environ 53 % de la population mondiale) selon l'Union

Internationale des Télécommunications[1], l'Internet de demain est sur le point de devenir une infrastructure entièrement omniprésente offrant une connectivité à large bande n'importe où et n'importe quand.

L'interopérabilité a toujours été un sujet important en informatique. Partant des ingrédients obscurs d'un réseau simple pour échanger le code Morse et des cartes perforées permettant un métier à tisser semi-automatique, la recette a évolué en une infrastructure immensément puissante qui servirait de base à la communication et à la collaboration modernes. Il y a plusieurs décennies, les technologies fondamentales qui engendreraient Internet, tel que nous le connaissons, étaient encore à leurs balbutiements. Les trente dernières années ont amené ces technologies à un niveau mature; la communication, la collaboration et l'interopérabilité des systèmes dépendent désormais d'applications "puissantes" basées sur la technologie des services Web.

L'industrie des Technologies de l'Information et de la Communication (TIC) connaît un changement spectaculaire, similaire à l'évolution de l'industrie au siècle dernier. Le développement des solutions TIC passe maintenant d'une discipline axée principalement sur la productivité, à une autre approche visant à fournir des services personnalisés de bout en bout capables de répondre aux exigences des clients et apporter un caractère distinctif aux solutions fournies. Cette évolution est, dans une large mesure, due à la maturité que l'industrie des TIC a atteint.

Dans un environnement professionnel, c'est le service qui est important pour un client donné, pas les composants matériels ou logiciels spécifiques qui sont utilisés pour fournir ce service. Actuellement, il y a un changement de paradigme en informatique. Il y a quelques années, l'informatique a appris à faire abstraction du matériel au logiciel; elle est maintenant dans une phase d'abstraction du logiciel au "*serviceware*" en termes de l'informatique orienté service. Initialement, ce changement s'est préoccupé par la complexité de gérer des services dans des applications à grande échelle. Cependant, cela implique en fait qu'une étape supplémentaire d'abstraction dans la conception et le

[1] L'Union Internationale des Télécommunications est l'agence des Nations Unies pour le développement spécialisé dans les technologies de l'information et de la communication, basée à Genève (Suisse) http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2016.pdf, page 2.

développement des systèmes TIC est encore nécessaire pour gérer correctement la complexité de la gestion du très grand nombre de services disponibles.

Les services Web existent et fonctionnent dans un environnement en constante expansion; le nombre de services Web disponibles ne cesse de croître. Au cours des dernières années, des sociétés du monde entier ont commencé à proposer des services Web permettant l'utilisation d'un sous-ensemble de leurs fonctionnalités, telles que *Twitter*[2], *LinkedIn*[3] ou *OPAP*[4]. De plus, les sites Web de publication de services Web ont connu une augmentation significative du nombre de services Web enregistrés et utilisés; *ProgrammableWeb*[5] affirme que les Interfaces de Programmation d'Application (API) des services Web enregistrées dans celui-ci sont passées de quelques centaines en 2005 à plus de 10 000 en 2013 (*ProgrammableWeb*, 2014) et actuellement à près de 20 000. Enfin, l'utilisation des services Web ne facilite pas seulement le développement d'applications à faible coût et rapide, mais constitue également une source importante de revenus pour les entreprises; par exemple, *Amazon Web Services* a généré à lui seul des recettes de 1,82 milliard de dollars au deuxième trimestre 2015, soit une augmentation de plus de 80% par rapport à 2014 (Novet, 2015). Ainsi, l'adoption généralisée et l'utilisation de technologies telles que les réseaux sociaux, les systèmes de recommandation sont devenues un facteur positif dans la croissance continue des normes et de la recherche sur les services Web.

Pour que les technologies axées sur le service évoluent, elles devront offrir un degré d'automatisation de la découverte, sélection, composition ainsi que l'invocation du service. L'intégration manuelle n'est pas réalisable lorsque l'on vise des moyens sophistiqués de gestion des systèmes qui sont dynamiquement composés de nombreux services.

2. Problématique et objectifs

Le domaine de la technologie de l'information a considérablement évolué au cours des dernières années. La taille et la complexité des logiciels ne cessent de croître. Cela est

[2] https://developer.twitter.com/en/docs
[3] https://developer.linkedin.com/docs/rest-api
[4] https://www.opap.gr/en/web-services
[5] http://www.programmableweb.com/

dû à de nouvelles exigences de plus en plus complexes. Par conséquent, plusieurs approches ont été développées pour améliorer la productivité et l'efficacité du logiciel. Parmi ces approches, nous pouvons citer l'approche des services, qui a une grande popularité. Les services Web sont l'une des technologies les plus connues et les plus répandues qui implémentent l'approche des services. Initialement proposés par IBM et Microsoft, les services Web sont partiellement standardisés sous l'égide du W3C. Un service Web est une entité logicielle sans état, modulaire, autonome et disponible, sous un format standard, sur Internet ou dans un intranet qui s'exécute à distance sur le site du fournisseur (Huhns et Singh, 2005). Cette notion de service Web désigne essentiellement une application mise à disposition sur Internet par un fournisseur de services et accessible par les clients via des protocoles Internet standards. L'utilisation exclusive de langages et de protocoles basés sur XML, ainsi que les standards actuels d'Internet comme HTTP, favorisent l'interopérabilité entre les systèmes d'information distribués.

Il n'est pas toujours possible de répondre à une requête d'un client par un seul service Web. Cela nécessite souvent la composition de plusieurs services. La création de nouvelles fonctionnalités en combinant les fonctionnalités fournies par les services Web existants ou la composition de services Web est l'un des nombreux défis à relever dans le domaine des services Web. Les services Web restent indépendants des systèmes d'exploitation et des plates-formes de développement, ce qui facilite les interactions avec les applications clientes et entre différents services Web impliqués dans une composition. Martin et al. (Martin et al., 2005) définissent la tâche de composition des services Web comme le processus de sélection, combinaison et exécution des services Web pour atteindre un objectif donné. L'exécution dynamique d'un service Web composite inclut nécessairement l'étape de sélection dynamique des services Web (composants). Il convient de noter que la demande en dynamisme et de flexibilité est devenue très forte avec l'avènement des services Web, qui apparaissent et disparaissent à la volée. La sélection des meilleurs services représente un véritable défi. La sélection d'un service Web consiste à sélectionner le service le plus approprié en fonction d'une exigence donnée. De nombreuses approches ont été proposées dans la littérature [(Wang et al., 2006), (Jaeger & Muhl, 2007), (Wang & Vassileva, 2007)].

La création de communautés de services Web peut être utile pour réduire l'espace de recherche des services Web [(Metrouh et al., 2012), (Metrouh & Valérie, 2013), (Metrouh & Mokhati, 2013), (Metrouh & Mokhati, 2016), (Metrouh & Mokhati, 2018)]. Les communautés de services Web offriront la possibilité d'établir un cadre pour la composition dynamique de ces derniers et ainsi pallier les insuffisances des annuaires UDDI (Fauvet et al., 2005). Plusieurs définitions ont été proposées dans la littérature. Maamar et al. (Maamar et al., 2007) ont considéré une communauté de services Web comme un moyen de fournir une description commune d'une fonctionnalité souhaitée sans se référer explicitement à un service Web spécifique. Benatallah et al. (Benatallah et al., 2003) ont défini une communauté comme une collection de services Web avec des caractéristiques communes, bien que ces services Web puissent avoir des propriétés non fonctionnelles distinctes. Dans notre cas, nous décrivons la notion de communauté de services Web comme relation entre services Web (construction de maisons, réservation d'hôtels par exemple) caractérisée par des poids. Plus de détails sont donnés au chapitre 4, section 3. Il serait également intéressant de définir ces communautés de services Web au sein de réseaux sociaux de services Web.

Les réseaux sociaux de services Web ou les services Web sociaux diffèrent des réseaux sociaux conventionnels. Les réseaux sociaux sont basés sur la coopération absolue et l'assistance mutuelle entre leurs membres (c'est-à-dire sans concurrence). Par contre, les services Web dans les réseaux sociaux sont particulièrement compétitifs (Maamar et al., 2009). Les services Web sont les seuls constituants du réseau social. La plupart des approches proposées dans la littérature pour la composition des services Web se concentrent sur la description des services Web eux-mêmes et négligent leurs interactions. Notre approche tire parti de la puissance des réseaux sociaux et des principes des systèmes de recommandation pour définir les interactions entre les services Web. La construction de tels réseaux nécessite une représentation des différentes interactions dans lesquelles les services Web sont intervenus. Le processus de composition des services Web proposé est basé sur ces interactions et sur la définition d'un modèle de qualité de service (QoS).

Malik et al. (Malik et al., 2009) définissent la qualité de service comme étant un ensemble de caractéristiques quantitatives et qualitatives d'un système, nécessaire pour

obtenir la fonctionnalité requise d'une application. La QoS est un concept large qui couvre de multiples propriétés ou dimensions non fonctionnelles, dont certaines peuvent desservir des aspects spécifiques et d'autres plus générales. Des exemples d'attributs QoS incluent le temps de réponse, la disponibilité, la confidentialité, la sécurité et la fiabilité. Le modèle de QoS proposé, dans le cadre de cette thèse, n'est pas limité à un ensemble particulier de paramètres QoS. Tout modèle de QoS pourrait être considéré. La seule contrainte, est que les paramètres choisis soient collectés sur un système automatisé sans aucune intervention de l'utilisateur, afin de préserver l'aspect automatique de la sélection des services Web.

Un autre défi dans les services Web est la personnalisation. La plupart des services Web sont fournis à différents consommateurs exactement de la même manière. Ils ne sont pas conçus et mis en œuvre pour une consommation de service personnalisée. Dans de nombreuses situations, la sélection de service nécessite de prendre en considération les besoins et les préférences des consommateurs individuels. Par exemple, dans le secteur immobilier, on peut utiliser un service de référencement en ligne pour rechercher un peintre, un électricien, un constructeur, un menuisier et un plombier. Les services Web *WS-Peintre*, *WS-Maçon*, *WS-Electricien*, *WS-Plombier* et *WS-Menuisier* ont des relations fonctionnelles parce qu'ils réalisent ensemble une tâche complexe "Construction de Maison". En fait, une relation fonctionnelle est un lien entre des services Web qui peuvent coopérer pour accomplir la même tâche. Ces relations permettent de différencier les services Web qui remplissent ensemble la même tâche complexe des services Web qui remplissent la même tâche d'une manière différente en offrant différentes alternatives pour l'utilisateur. Les services Web personnalisés désignent les services qui peuvent être adaptés aux besoins et aux préférences des consommateurs de services individuels.

L'objectif principal de cette recherche est donc d'explorer la possibilité d'une composition automatique de services Web dynamiques et de fournir des solutions évolutives, optimales et robustes en fonction des besoins des utilisateurs et de leurs préférences.

3. Motivations

Cette recherche est motivée par les besoins de tous les jours des personnes à prendre des décisions sur les choix disponibles à partir de services Web. La prise de décision dans des situations complexes et dynamiques a toujours été une tâche difficile. Les modèles de décision traditionnels pour la sélection et composition de services Web basés uniquement sur l'utilité ne sont plus adéquats ; la composition de services Web est de plus en plus compliquée par rapport aux approches traditionnelles car les consommateurs ne savent même pas avec qui, ils interagissent. Pour illustrer les défis liés à la composition et à la recommandation de services Web, les exemples suivants illustrent les principales difficultés, et en même temps, la motivation de l'approche proposée.

3.1. Scénario 1

Pour expliquer le processus de composition de services Web, considérons un exemple simple de planification de vacances. La planification des vacances est un exemple simple, facile à évaluer et traditionnel, discuté par différents groupes de recherche dans le domaine de la composition de services. Chaque année, des millions d'internautes du monde entier naviguent sur différents sites afin de planifier leurs vacances. Les clients ont la possibilité (via des formulaires en ligne) de fournir leurs exigences de base, tels que la durée du séjour, les préférences alimentaires, le budget et les dispositions de voyage.

Une fois que les valeurs d'entrée ont été introduites (insérées) par le client, le système tentera de les faire correspondre aux résultats basés sur les préférences du client, ainsi que l'historique de l'utilisation du système par ce dernier. Les résultats sont ensuite présentés au client. Les résultats seront analysés par le client, et il pourra rectifier le tir et redéfinir ses préférences ou les ajuster, tout en espérant des résultats meilleurs. Certaines agences de voyage offriront des forfaits sur mesure, où les clients auront certaines de leurs préférences, mais devront également accepter de payer pour des mesures supplémentaires non souhaitées. Certains sites Web fournissent des offres en quelques secondes, une fois que le client a fourni sa destination et les préférences souhaités. Ses vacances sont planifiées, non seulement par une agence de voyage, mais

par différents intervenants tels que les hôtels, les agences de location de voiture, les compagnies d'assurance voyage et les banques.

3.2. Scénario 2

Un autre scénario, dans le domaine de l'immobilier où un client est amené à effectuer les étapes nécessaires pour construire une maison. Le client a généralement une connaissance partielle de la procédure d'acquisition ou de construction de maisons. Cette connaissance n'est pas exhaustive et peut être améliorée grâce à l'utilisation du système. Par exemple, le client tente de résoudre la tâche "Construction de maison". Le client a probablement besoin de résoudre une autre tâche "achat de terrain", il aura peut être besoin de tâches complémentaires : rechercher un peintre, un électricien, un menuisier ou un plombier etc. Les services Web *WS-Peintre*, *WS-Maçon*, *WS-Electricien*, *WS-Plombier* et *WS-Menuisier etc.*, associés à chaque tâche complémentaire ont des relations fonctionnelles parce qu'ils réalisent ensemble une tâche complexe "Construction de maison". Ces tâches pourraient être proposées aux clients, sans qu'elles ne soient explicitement demandées. En fait, une relation fonctionnelle est un lien entre des services Web qui peuvent coopérer pour accomplir une même tâche complexe.

4. Contributions

Pour résoudre les problèmes mentionnés ci-dessus, cette thèse propose une approche innovante pour la composition des services Web basée sur des attributs fonctionnels et non fonctionnels et les préférences individuelles des utilisateurs. La composition des services Web est liée aux réseaux sociaux et prend en compte l'historique d'invocation des services par les utilisateurs. Plus précisément, les contributions sont les suivantes.

(1) Nous extrayons les intérêts fonctionnels et les préférences de QoS d'un utilisateur en explorant son historique d'utilisation des services Web. Les intérêts des utilisateurs et leurs préférences QoS sont utilisés pour mesurer les paramètres de qualité de service pour les services Web candidats. Un défi important vient de l'hypothèse d'un service Web non fiable et du traitement des ensembles dynamiques des utilisateurs en ligne.

(2) Un modèle de QoS est défini. Les paramètres du modèle de QoS proposé sont faciles à mesurer et peuvent être collectés sur un système automatisé sans intervention de l'utilisateur. Un score est calculé pour chaque service Web candidat en utilisant les paramètres QoS.

(3) La création du réseau social des services Web proposé est détaillée ainsi que les différents graphes associés à ce réseau: Graphe de collaboration qui définit les différentes communautés de services Web dans le réseau social de ces derniers et le graphe de recommandation des services Web pour la composition.

(4) L'espace de recherche des services Web est considérablement réduit en définissant des communautés dans le réseau social de services Web. Nous décrivons la notion de communauté de services Web comme étant une relation entre des services Web caractérisés par des poids.

(5) Nous proposons une approche innovante qui combine les réseaux sociaux de services Web avec les principes des systèmes de recommandation pour définir les interactions entre les services Web. Ces interactions sont exploitées pour développer un algorithme de sélection de services Web dans le but de leur composition. Un deuxième algorithme a été proposé pour la composition personnalisée de ces derniers. Les résultats expérimentaux montrent que l'approche proposée offre des performances intéressantes.

(6) Nous avons également proposé un algorithme de sélection de services Web dans le cas de substitution, si le service Web s'avère en panne ou indisponible. Le processus de sélection du service Web proposé prend en considération les préférences du client et le substitut proposé a des performances similaires ou les meilleures performances existantes.

5. Plan de la thèse

La présente thèse est organisée en cinq (04) chapitres. Les trois premiers chapitres représentent l'état de l'art des domaines attachés à notre problématique. Ensuite, nous présentons nos contributions dans le chapitre suivant. Ainsi, ce manuscrit est composé des chapitres suivants :

Chapitre 1 présente la terminologie et les définitions liées aux services Web utilisés tout au long de cette thèse. Un aperçu des technologies clés: les services Web, les réseaux sociaux basés sur le Web, les systèmes de recommandation et la personnalisation des services Web sont ensuite présentés.

Chapitre 2 étudie le problème de la sélection et de la recommandation des services Web sociaux. Un aperçu sur la sélection des services sociaux et le problème de la recommandation, mettant en évidence les défis actuels est présenté. Une revue de la littérature relative à l'approche proposée pour la sélection des services Web est présentée. Plus précisément trois approches de sélection et de recommandation de services sociaux: réputation, recommandation et renvoi.

Chapitre 3 décrit en détail le problème de la composition de services, ainsi que les différents types d'approches de composition qui existent dans la littérature. Une étude comparative des différentes approches est présentée. Les avantages et les inconvénients de chaque approche sont discutés.

Chapitre 4 vise à proposer une approche innovante de la composition dynamiques des services Web basée sur les attributs fonctionnels et non fonctionnels et les préférences individuelles des utilisateurs. Dans cette approche, la composition des services Web est liée aux réseaux sociaux et prend en compte l'historique d'invocation de ces derniers par les utilisateurs. Ce chapitre présente également les expérimentations et l'évaluation de l'approche proposée. Une solution innovante pour la sélection des services Web dans le contexte de la substitution de ces derniers, en cas d'échec ou de leur indisponibilité est également présentée. Ce chapitre présente les expérimentations et l'évaluation de cette approche.

Enfin, une conclusion générale fournie quelques remarques, ainsi qu'un aperçu des principales contributions de cette thèse, et se termine par présenter les grandes orientations des futures recherches.

Chapitre 1

Vue d'ensemble des technologies clés

Sommaire

1. Introduction 12
2. Service Oriented Computing (SOC) 13
3. Services Web 14
   - 3.1 Définition des services Web 14
   - 3.2 Architecture des services Web (SOA) 15
   - 3.3 Propriétés d'un service Web 18
     - 3.3.1 Propriétés fonctionnelles 18
     - 3.3.2 Propriétés comportementales 18
     - 3.3.3 Propriétés non fonctionnelles 19
   - 3.4 Technologies des services Web 19
     - 3.4.1 Langage de description des services Web (WSDL) 21
     - 3.4.2 Protocole d'accès aux objets simples (SOAP) 23
       - 3.4.2.1 Structure d'un message SOAP 24
       - 3.4.2.2 Modèle de traitement 25
       - 3.4.2.3 Protocole de liaison 25
       - 3.4.2.4 Avantages et inconvénients de l'utilisation de SOAP 26
     - 3.4.3 Description universelle, protocole de découverte et d'intégration (UDDI) 27
   - 3.5 Caractéristiques des services Web 29
   - 3.6 Types de services Web 30

3.7 Avantages de l'utilisation des services Web 32
4. Paramètres de qualité de services (QoS) 33
5. Personnalisation des services Web 35
6. Réseaux sociaux sur le Web 37
7. Systèmes de recommandation 39
8. Conclusion 40

1. Introduction

Le Web tel que nous le connaissons est un énorme succès. Actuellement, il y a des milliards de pages disponibles sur le Web et ce nombre ne cesse d'augmenter rapidement. Le succès du Web est dû, principalement, à son ouverture sous-jacente et aux approches et technologies évolutives. Un regard plus approfondi sur le Web actuel montre qu'il est plutôt statique qu'un système dynamique. Il est statique dans le sens où les applications disponibles sur le Web ne peuvent pas être facilement réutilisées dans différents contextes et scénarios. Le Web est conçu actuellement pour les interactions entre les applications et les humains, permettant le partage d'informations. Il offre un support de base pour le commerce électronique entre entreprises et clients (B2C), et très limité pour la prise en charge de l'intégration interentreprises (B2B) et de l'intégration d'applications d'entreprises (EAI). L'interopérabilité est un défi majeur que le Web et les technologies Web ne supportent pas implicitement. Dans ce contexte, l'idée des services Web a été introduite en tant que technologie permettant l'interopérabilité et l'intégration des fonctionnalités métiers.

La vision ultime est qu'un programme chargé de fournir un résultat peut utiliser des services Web comme support pour son calcul ou traitement. Le programme peut découvrir des services Web et les invoquer de manière entièrement automatisée. Par conséquent, il devient un demandeur de service. Si les services Web ont un coût, le programme doit avoir la possibilité de rechercher les services les moins chers ainsi que toutes les méthodes de paiement possibles.

Les services Web connectent les ordinateurs et les appareils les uns avec les autres en utilisant Internet pour échanger et combiner des données. Ils peuvent être définis

comme des composants logiciels qui peuvent être assemblés sur Internet en utilisant des protocoles standards pour exécuter des fonctions ou des processus métier. La clé des services Web est la création de logiciels à la volée en utilisant des composants logiciels faiblement couplés et réutilisables. Ceci a des implications fondamentales sur le plan technique et commercial. Les logiciels peuvent être livrés et payés en tant que flux de services fluides, par opposition aux produits emballés. Les services fournis peuvent être complètement décentralisés et distribués sur Internet et accessible par une grande variété de dispositifs de communication. Les entreprises peuvent ainsi être libérées du fardeau d'une intégration logicielle complexe, lente et coûteuse et se concentrer plutôt sur la valeur de leurs offres et de leurs tâches critiques. L'Internet deviendra alors une plate-forme commune et mondiale où les organisations et les individus communiquent entre eux pour mener diverses activités commerciales et fournir des services à valeur ajoutée. Les barrières pour fournir de nouvelles offres et pénétrer de nouveaux marchés seront réduites aux petites et moyennes entreprises.

La conception des systèmes d'information d'entreprises a connu plusieurs changements, ces dernières années. Afin de répondre aux exigences des entreprises pour plus de flexibilité et de dynamisme, les applications monolithiques traditionnelles se sont substituées par des unités de fonctionnalité plus petites, englobantes, appelées services. Les systèmes d'information doivent être adaptés à ce paradigme, avec de nouvelles applications développées en tant que services, et les systèmes existants doivent être mis à jour afin d'exposer les interfaces des services. Les tendances actuelles est vers une conception de systèmes d'information qui adoptent le paradigme du *Service Oriented Computing* (SOC) avec la mise en œuvre de l'architecture orientée services (SOA) et les technologies pertinentes des services Web.

2. Service Oriented Computing (SOC)

Le *Service Oriented Computing* (SOC) ou l'informatique orientée services est un nouveau paradigme informatique qui utilise les services comme éléments fondamentaux pour le développement d'applications d'entreprises rapides, économiques et faciles à intégrer (Preist, 2004). L'un des principaux objectifs de SOC est de permettre le développement des réseaux d'applications intégrées et collaboratives, indépendamment des plateformes et des langages de programmation utilisés pour les développer. Dans ce

paradigme, les services sont autonomes, auto-descriptifs et indépendants des plateformes, fournissant un accès uniforme et omniprésent à l'information pour une large gamme de dispositifs informatiques, tels que les ordinateurs de bureau, les assistants personnels numériques (PDA) et les téléphones cellulaires. Tout morceau de code et tout composant d'application déployé sur un système peut être réutilisé et transformé en un service disponible sur le réseau. Ces services peuvent alors être facilement décrits, publiés, découverts et assemblés dynamiquement pour développer des systèmes distribués et interopérables.

L'objectif principal de SOC est de faciliter l'intégration des anciennes applications et celles nouvellement construites. SOC doit surmonter et résoudre des conflits hétérogènes dus à différentes plateformes, langages de programmation, pare-feux de sécurité, etc. L'idée de base derrière cette orientation est de permettre aux applications, développées en utilisant différents languages, technologies ou plateformes d'être présentées en tant que services puis de les interconnecter en exploitant l'infrastructure Web, avec des standards tels que HTTP, XML, SOAP, WSDL détaillés plus loin dans ce chapitre, paragraphe 3.4.

3. Services Web

La notion de service est sémantiquement surchargée. Différentes communautés ont une compréhension différente de ce qu'est un service (Preist, 2004). Par exemple, dans le milieu des affaires, un service est considéré comme une activité commerciale qui entraîne souvent des résultats ou des avantages intangibles (Baida et al., 2004), tandis qu'en informatique, les termes service et service Web sont souvent considérés comme interchangeables pour nommer une entité logiciel accessible sur Internet. Dans le cadre de cette thèse nous ne faisons aucune distinction entre les deux concepts.

3.1. Définition des services Web

Définition 3.1.1 :

Un service Web est une entité logicielle sans état, modulaire, autonome et disponible, sous un format standard, sur Internet ou dans un intranet qui s'exécute à distance sur le site du fournisseur (Huhns & Singh, 2005).

Définition 3.1.2 :

Les services Web comme toutes autres technologies de middleware[1]; vise à fournir des mécanismes pour relier des plateformes hétérogènes, permettant aux données de circuler à travers différents programmes (Staab et al., 2003).

Définition 3.1.3 :

Selon le *World Wide Web Consortium* (*W3C*), les services Web sont identifiés par un *URI* (*Uniform Resource Identifier*), leurs interfaces sont définies dans *WSDL* (*Web Service Description Language*), publié dans le registre *UDDI* (*Universal Description, Discovery and Integration*), les services peuvent être découverts et invoqués par d'autres systèmes logiciels. Ces systèmes interagissent avec les services Web en utilisant des messages *XML* véhiculés par le protocole *SOAP* (*Simple Object Access Protocol*) (Booth et al., 2004).

La technologie des services Web vise donc à standardiser la présentation des services offerts par une entreprise et à rendre leur accès transparent à tout type de plateformes, à travers un certain nombre de normes d'interopérabilité.

3.2. Architecture orientée services (SOA)

Le paradigme de l'informatique orientée service (SOC) peut être mis en œuvre abstraitement par l'architecture du système appelée architecture orientée service (SOA) [(Bajaj et al., 2006), (Bellwood, 2002)]. Le but de cette architecture est de répondre aux exigences de couplage faible, standards, et le calcul distribué indépendant des protocoles, en mappant les systèmes d'information des entreprises d'une manière isomorphique au flux global des processus métiers (Stollberg et al., 2004). Cette tentative est considérée comme le dernier développement d'une longue série d'avancées en génie logiciel traitant de la réutilisation de composants logiciels.

Historiquement, la première étape majeure de cette évolution a été le développement du concept "fonction". En utilisant des fonctions, un programme est décomposé en sous-programmes plus petits et le code d'écriture est axé sur l'idée de l'interface de programmation d'application (API). Une API, pratiquement, représente le

[1] Middleware ou intergiciel.

contrat auquel un composant logiciel doit s'engager. La deuxième étape majeure a été le développement du concept "objet". Un objet est un bloc de construction de base qui contient à la fois des données et des fonctions dans une seule unité encapsulée. Avec le paradigme orienté objet, les notions de classes, d'héritage et de polymorphisme sont introduites. De cette façon, les classes peuvent être considérées comme un treillis. Le concept "service" devient la prochaine étape d'évolution introduite avec l'avènement du SOC et de sa mise en œuvre SOA.

Selon le groupe de travail du World Wide Web Consortium (W3C), la SOA est définie comme suit :

Définition 3.2.1 :

L'architecture orientée services est un paradigme pour l'organisation et l'utilisation de capacités distribuées qui peuvent être sous le contrôle de différents domaines de propriétés. Il fournit un moyen uniforme d'offrir, de découvrir, d'interagir et d'utiliser des capacités pour produire les effets souhaités compatibles avec des conditions préalables et des attentes mesurables (Booth et al., 2004).

Une définition plus récente de SOA donnée par Erl est la suivante:

Définition 3.2.2 :

L'architecture orientée services est une forme d'architecture technologique qui adhère aux principes de l'orientation service. Une fois réalisée à travers la plate-forme technologique de services Web, SOA établit le potentiel de soutenir et de promouvoir ces derniers dans les processus métiers et les domaines d'automatisation d'une entreprise (Erl, 2016).

La figure 1.1 suivante illustre l'architecture SOA, qui comprend trois composants: les consommateurs de services, les fournisseurs de services et les annuaires de services. En ignorant les techniques détaillées pour la connexion des trois composants, ce modèle représente également le modèle de base SOA. Un annuaire de service agit comme intermédiaire entre le fournisseur et le consommateur de service. Un fournisseur de services publie simplement le service. Un consommateur tente de trouver des services

en utilisant les annuaires de services ; s'il trouve le service désiré, il peut établir un contrat avec le fournisseur afin de consommer ce service.

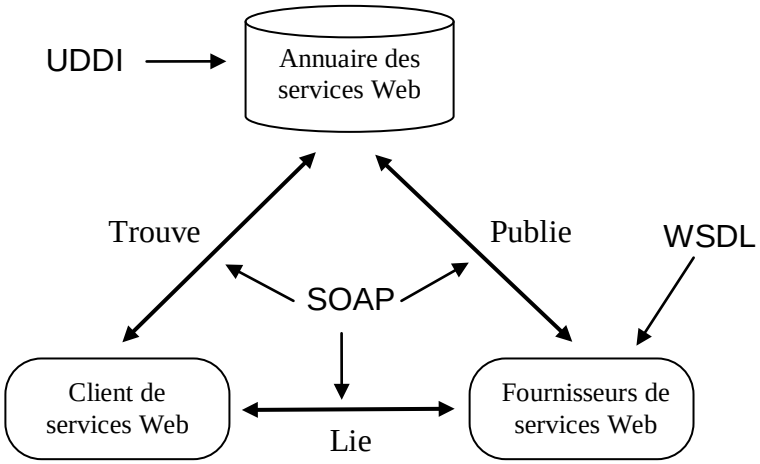


Figure 1.1 : Modèle de base de l'architecture SOA.

La vue logique fondamentale des services dans SOA est basée sur la séparation de la description du service (fréquemment appelée interface) et son implémentation (Stollberg et al., 2004). Une interface de service définit l'identité de celui-ci et sa logistique d'invocation. Une implémentation de service, implémente les opérations internes que le service est sensé réaliser. Basé sur cette séparation, les fournisseurs de services et les consommateurs de services sont faiblement couplés. De plus, les services peuvent être significativement réutilisés et adaptés à certaines exigences. Etant donné que les interfaces de service sont indépendantes de la plate-forme et que l'implémentation est transparente pour les consommateurs de services, un client depuis n'importe quel périphérique de communication utilisant une plate-forme de calcul, un système d'exploitation et un langage de programmation devrait être capable d'utiliser le service. Les deux facettes du service sont distinctes; elles sont conçues et maintenues comme des éléments distincts, bien que leur existence soit étroitement liée.

Basé sur l'autonomie de service et la séparation nette des interfaces de service de l'implémentation interne, SOA fournit une architecture plus flexible qui unifie les processus métier en modulant de grandes applications en services. En outre, des applications à l'échelle de l'entreprise ou même interentreprises peuvent être réalisées au moyen du développement, l'intégration et l'adaptation des services.

Le reste de ce chapitre se concentre sur les solutions techniques disponibles pour mettre en œuvre la vision de l'architecture orientée services et des services Web.

3.3. Propriétés d'un service Web

Selon Dumas et Fauvet[2] trois types de propriétés peuvent être identifiés pour les services Web ; Ils peuvent être brièvement définis comme suit:

3.3.1. Propriétés fonctionnelles

La description fonctionnelle contient la spécification formelle de ce qu'un service peut faire. WSDL (*Web Services Description Language*), détaillé plus loin (paragraphe 3.4.1), est un langage de la famille XML permettant de décrire les types de données supportés et les fonctions offertes par un service Web. L'objectif est de fournir la description, en XML, des services indépendamment de la plate-forme et du langage utilisés et sous une forme que des personnes ou des programmes peuvent interpréter.

3.3.2. Propriétés comportementales

La portée de WSDL est limitée à la description des types de données incluses dans les messages qu'un service est capable de recevoir ou d'émettre. Dans de nombreuses applications, ces descriptions uniquement structurelles n'offrent pas assez d'information sur les contraintes régissant les interactions dans lesquelles un service peut ou est censé s'engager. Dans certains cas, ces contraintes sont assez simples, comme par exemple : "le fournisseur n'envoie le bordereau de livraison qu'après avoir reçu la confirmation du paiement". Parfois, ces contraintes sont relativement complexes. Un fournisseur reçoit un "Bon de Commande" de la part d'un client, il doit répondre avec "un Récépissé de Bon de Commande". Par la suite, le fournisseur enverra zéro, une ou plusieurs "Réponses" au client jusqu'à avoir donné une réponse (une acceptation ou un rejet) pour chacune des lignes de Commande. Au cours de ce processus, le fournisseur peut recevoir une "Annulation" de la part du client, dans ce dernier cas, le fournisseur répond

[2] Intergiciel et Construction d'Applications Réparties © 2006-2008 M. Dumas, M.-C. Fauvet (version du 12 juin 2008 - 17:18). Licence Creative Commons (http://creativecommons.org/licenses/by-nc-nd/2.0/fr/deed.fr) [En ligne] Disponible à: http://deptinfo.unice.fr/~baude/WS/CoursICAR-WS.pdf [Dernière visite: 22/04/2018]

avec un "Récépissé d'Annulation de Bon de commande". Dès lors, le fournisseur ne doit plus envoyer de "Réponses" au client.

3.3.3. Propriétés non fonctionnelles

Les descriptions non fonctionnelles capturent les contraintes sur les deux types précédents de propriétés (Chung, 1991). Les services Web étant généralement développés par des équipes indépendantes, ont besoin d'être décrits de façon précise selon des conventions standards, et de telle manière que leurs descriptions puissent être utilisées au maximum pour automatiser le processus de développement et de déploiement des futurs services devant interagir avec un service existant. WSDL permet de décrire les opérations fournies par un service Web, les types de données des messages devant être échangés pour invoquer ces opérations, et les dépendances comportementales entre ces opérations. Cependant, ils ne permettent pas de décrire des aspects non-fonctionnels des services tels que leur capacité à garantir une certaine qualité de service par rapport à des préoccupations telles que la sécurité, la fiabilité, la journalisation des accès ou la gestion de transactions.

Considérant l'exemple du bon de commande présenté ci-dessus, en invoquant sa fonctionnalité (payement) peut être contraint en utilisant une connexion sécurisée (sécurité comme propriété non fonctionnelle) ou en effectuant réellement l'invocation des services à un certain moment (disponibilité temporelle comme propriété non fonctionnelle).

3.4. Technologies des services Web

L'objectif de la technologie des services Web est de permettre aux applications de fonctionner ensemble sur des protocoles Internet standards, sans intervention humaine directe. L'infrastructure minimale requise par le paradigme des services Web est délibérément faible pour garantir que les services Web peuvent être mis en œuvre et accessibles à partir de n'importe quelle plate-forme en utilisant n'importe quel langage de programmation et technologie.

Par intention, les services Web ne sont pas implémentés de manière monolithique, mais représentent plutôt une collection de plusieurs technologies connexes. La

définition plus généralement acceptée des services Web s'appuie sur un ensemble de normes complémentaires spécifiques, illustrées par la figure 1.2. Le développement de normes ouvertes et acceptées est une force clé des coalitions qui ont développé l'infrastructure des services Web. En même temps, comme on peut le voir à la figure 1.2, ces efforts ont entraîné la prolifération d'un nombre vertigineux de normes et d'acronymes émergents. Pour simplifier les choses, nous allons présenter les normes les plus importantes de la pile de technologie des services Web, utilisés dans le cadre de cette thèse.

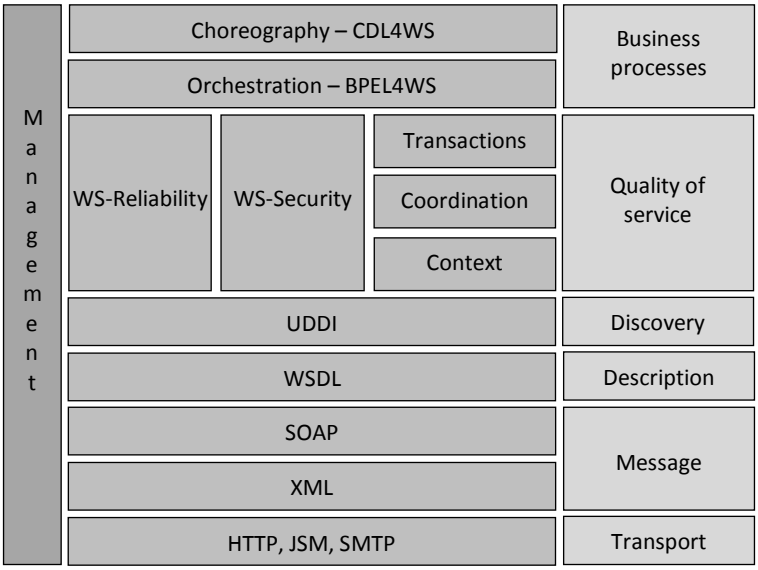


Figure1.2 : Architecture SOA (Papazoglou, 2008).

Le processus global d'utilisation des services Web est illustré à la figure 1.1. Les fournisseurs de services enregistrent et publient leurs services au niveau d'un registre UDDI en fournissant des détails tels que les coordonnées du fournisseur de services, les pointeurs vers les services et leurs descriptions WSDL, etc. Une fois le registre renseigné sur les services, les consommateurs de services, qui dans la plupart des cas sont des développeurs d'applications orientées services, pourraient accéder au registre. Les requêtes UDDI sont émises par les consommateurs de services, et potentiellement le registre UDDI répond positivement à ces requêtes. Si un service pertinent est trouvé, le registre fournit au consommateur de service les informations nécessaires pour accéder à ce dernier. Cependant, le consommateur de service doit toujours trouver comment accéder au service et l'invoquer, informations généralement disponibles dans la

description WSDL du service. Cela introduit une grosse surcharge et génère des problèmes lorsque des applications distribuées sont développées. Maintenir l'intervention humaine (c'est-à-dire l'intervention du développeur de logiciel) au minimum et automatiser les tâches liées au service pourrait surmonter le problème mentionné précédemment.

3.4.1. Langage de description des services Web (WSDL)

WSDL (*Web Service Description Language*) (Bajaj et al., 2006) est un langage de description basé sur XML pour les services Web. Un service est défini comme une collection de points de terminaison ou de ports réseau. Deux niveaux de description de service sont identifiés dans WSDL. Tout d'abord, un niveau d'interface abstrait fournit des détails sur les types, les opérations et les interfaces du service. Deuxièmement, les protocoles réseau concrets et les points de terminaison spécifient comment et où le service peut être accédé. La suite de cette section décrit en détail chacun des éléments WSDL, en utilisant la version WSDL 2.0. L'exemple 1.1 montre la structure d'un fichier WSDL, en utilisant la version WSDL 2.0.

Au niveau de l'interface abstraite, les types sont définis en termes de schémas XML. Les types sont utilisés pour définir des messages qui représentent des données échangées entre les fournisseurs et les clients de services. Les messages sont regroupés en opérations, chaque opération consistant en un ou plusieurs messages. Les opérations précisent aussi l'ordre dans lequel les messages sont envoyés ou reçus, selon un modèle de message prédéfini (c'est-à-dire "*In-Out*", "*Out-Only*", "*Robust-In-Only*", "*In-Optional-Out*"). Finalement, l'interface regroupe plus d'opérations. La représentation graphique de la structure de WSDL est présentée dans la figure 1.3.

Pour communiquer avec un service Web décrit par une interface abstraite, un client doit savoir comment les messages XML sont sérialisés sur le réseau et où exactement ils devraient être envoyés. Dans WSDL, la sérialisation des messages sur le réseau est décrite dans une liaison, puis une construction de service énumère un certain nombre de d'adresses concrètes de points de terminaison.

Exemple 1.1 : Structure d'un fichier WSDL.

```
<?xml version="1.0" encoding="UTF-8"?>
<description xmlns="http://www.w3.org/ns/wsdl"
        xmlns:tns="http://www.example.com/wsdl20sample"
        xmlns:whttp="http://www.w3.org/ns/wsdl/http"
        xmlns:wsoap="http://www.w3.org/ns/wsdl/soap"
        targetNamespace="http://www.example.com/wsdl20sample">

<!-- Abstract types -->
  <types>
    <xs:schema xmlns="http://www.example.com/wsdl20sample"
      xmlns:xs="http://www.w3.org/2001/XMLSchema"
      targetNamespace="http://www.example.com/wsdl20sample">
      <xs:element name="request"> ... </xs:element>
      <xs:element name="response"> ... </xs:element>
    </xs:schema>
  </types>

<!-- Abstract interfaces -->
  <interface name="Interface1">
    <fault name="Error1" element="tns:response"/>
    <operation name="Opp1" pattern="http://www.w3.org/ns/wsdl/in□out">
      <input messageLabel="Msg1" element="tns:request"/>
      <output messageLabel="Msg2" element="tns:response"/>
    </operation>
  </interface>

<!-- Concrete Binding Over HTTP -->
  <binding name="HttpBinding" interface="tns:Interface1"
    type="http://www.w3.org/ns/wsdl/http">
    <operation ref="tns:Opp1" whttp:method="GET"/>
  </binding>

<!-- Concrete Binding with SOAP-->
  <binding name="SoapBinding" interface="tns:Interface1"
      type="http://www.w3.org/ns/wsdl/soap"
      wsoap:protocol="http://www.w3.org/2003/05/soap/bindings/HTTP/"
      wsoap:mepDefault="http://www.w3.org/2003/05/soap/mep/request-response">
    <operation ref="tns:Opp1" />
  </binding>

<!-- Web service offering endpoints for both the bindings-->
  <service name="Service1" interface="tns:Interface1">
    <endpoint name="HttpEndpoint" binding="tns:HttpBinding"
    address="http://www.example.com/rest
    /"/>
    <endpoint name="SoapEndpoint" binding="tns:SoapBinding"
    address="http://www.example.com/
    soap/"/>
  </service>
</description>
```

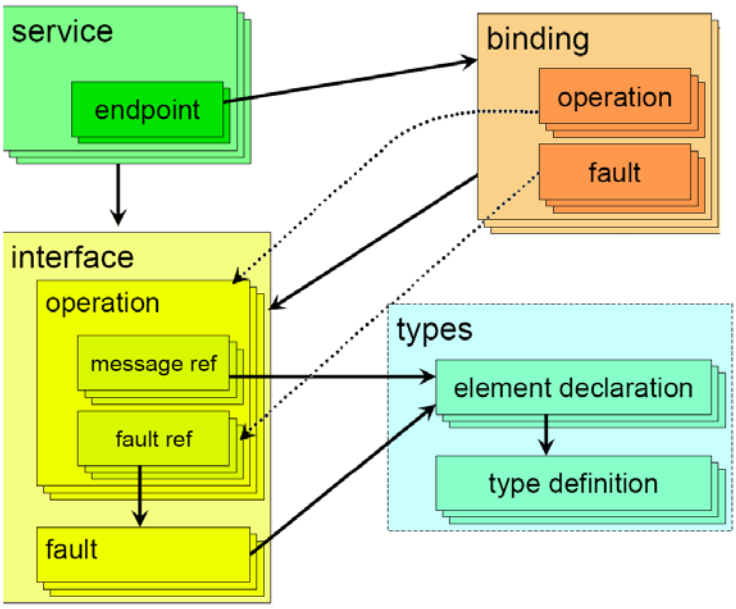


Figure 1.3 : Les éléments de WSDL (Bajaj et al., 2006).

Une liaison suit généralement la structure d'une interface et spécifie les détails de sérialisation. La spécification WSDL contient deux spécifications de liaisons prédéfinies, une pour SOAP (sur HTTP) et une pour HTTP simple. Ces liaisons spécifient comment un message XML abstrait est incorporé dans une enveloppe de message SOAP ou dans un message HTTP, et comment les modèles d'échange de messages sont réalisés dans SOAP ou HTTP. Dans SOAP chaque défaillance doit avoir un code d'erreur, *Expéditeur* ou *Récepteur*, indiquant qui a causé le problème.

Un fichier WSDL représente un seul service Web physique qui implémente une seule interface. Le service Web peut être accessible à plusieurs points de terminaison, chacun potentiellement avec une liaison différente, par exemple un point de terminaison utilisant un protocole de messagerie optimisé sans cryptage de données pour l'environnement sécurisé d'un intranet, et un second point de terminaison en utilisant SOAP sur HTTP pour l'accès à partir du L'Internet.

3.4.2. Protocole d'accès aux objets simples (SOAP)

Les protocoles conventionnels de communication d'objets distribués, tels que CORBA, DCOM, Java / RMI, et d'autres protocoles de communication d'application à application pour les communications de serveur à serveur, présentent de graves faiblesses pour les

communications client à serveur. Ces faiblesses sont particulièrement remarquables lorsque les machines clientes sont dispersées sur Internet. Les protocoles de communication distribués conventionnels ont une exigence symétrique: les deux extrémités du lien de communication devraient être implémentées sous le même modèle d'objet distribué et nécessiteraient le déploiement de bibliothèques développées en commun. Pour répondre à de telles limitations, le protocole SOAP (Simple Object Access Protocol) a été développé. SOAP facilite l'interopérabilité entre un large éventail de programmes et de plateformes, rendant les applications existantes accessibles à un plus large éventail d'utilisateurs.

SOAP (*Simple Object Access Protocol*) (Curbera et al., 2001) est un format de message basé sur XML pour échanger des données XML arbitraires. Il utilise principalement HTTP/HTTPS comme protocole de communication sous-jacent. SOAP a été conçu pour traiter le transport et la messagerie pour les grands environnements distribués. Il définit comment organiser les informations à l'aide de XML afin qu'elles puissent être échangées entre les machines. Toutefois, SOAP ne prend pas en charge l'aspect sémantique des messages transmis. La communication entre deux parties doit être codée en SOAP et tout schéma de communication complexe entre les parties impliquées doit être mis en œuvre par le système sous-jacent. Il y a trois aspects importants qui définissent ensemble le protocole SOAP: (1) la structure du message, (2) le modèle de traitement et (3) le protocole de liaison.

3.4.2.1. Structure d'un message SOAP

Un message SOAP consiste en un élément d'enveloppe dans lequel l'application code toute information devant être envoyée. L'élément d'enveloppe définit le début et la fin du message. Un élément d'enveloppe contient un corps obligatoire et un élément d'en-tête facultatif. L'élément d'en-tête contient tous les attributs facultatifs du message, soit à un point intermédiaire, soit à l'extrémité finale. L'élément *body* contient des informations d'appel et de réponse, c'est-à-dire les données XML comprenant le message envoyé ou reçu. De plus, un message SOAP généré du côté service peut contenir un élément *fault* en cas d'échec. L'exemple 1.2 contient une structure d'un message SOAP montrant la relation entre les éléments décrits ci-dessus.

Exemple 1.2 : Structure d'un message SOAP.

```
<?xml version=1.0?> <soap:Envelope
  xmlns:soap=http://www.w3c.org/2001/12/soap-envelope
  soap:encodingStyle=http://www.w3c.org/2001/12/soap-encoding>
  <soap:Header>
  ...
  </soap:Header>
  <soap:Body>
    ...
    <soap:Fault>
    ...
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

3.4.2.2. Modèle de traitement

Le modèle de traitement SOAP définit les règles de traitement d'un message SOAP. Plus précisément, un modèle de traitement distribué est proposé, dans lequel le message SOAP provient d'un expéditeur SOAP initial et est envoyé à un récepteur SOAP final avec ou sans intermédiaires SOAP. Les informations d'en-tête d'un message SOAP jouent un rôle important dans le modèle de traitement SOAP global, définissant quelle partie du message est autorisée à être traitée et par qui. Les parties impliquées dans le modèle de traitement sont appelées nœuds SOAP. SOAP définit trois rôles pour les nœuds SOAP, à savoir: *aucun*, *suivant* et *récepteur ultime*. Dans un message SOAP, tout bloc d'en-tête affecté à un rôle *aucun* ne doit pas être traité par les nœuds SOAP. Un bloc d'en-tête affecté à un rôle *suivant* doit être traité par chaque nœud SOAP le long du chemin, à l'exclusion du nœud initial. Un bloc d'en-tête affecté à un rôle *récepteur ultime* doit être traité uniquement par le destinataire du message.

3.4.2.3. Protocole de liaison

Pour spécifier comment le message SOAP va être transporté sur le réseau, la partie de liaison de protocole SOAP est utilisée. SOAP permet l'échange de messages SOAP en utilisant une variété de protocoles sous-jacents. Un protocole commun utilisé pour la liaison SOAP est HTTP, mais d'autres protocoles peuvent également être utilisés, par ex. SMTP. Trois aspects de transport sont décrits dans le cadre d'une liaison SOAP, à

savoir: (1) adressage, qui spécifie à quoi ressemble une adresse de point de terminaison, (2) sérialisation, qui spécifie comment mettre un message XML dans les bits et octets sur le réseau et (3) connexion, qui spécifient comment envoyer les bits à un point de terminaison.

3.4.2.4. Avantages et inconvénients de l'utilisation de SOAP

Comme présenté dans (Papazoglou, 2008), l'utilisation de SOAP montre plusieurs avantages:

- **Simplicité:** SOAP est simple car il utilise du XML bien structuré et facile à analyser.
- **Portabilité:** SOAP est indépendant des plateformes et donc portable.
- **Pare-feu:** SOAP est capable de passer les firewalls qui bloquent totalement les autres protocoles. Ceci est possible grâce à l'utilisation du protocole HTTP.
- **Utilisation de la norme ouverte:** SOAP est basé sur le standard XML ouvert pour formater les données. En conséquence, SOAP devient facilement extensible et bien supporté.
- **Interopérabilité:** SOAP s'appuie sur des technologies ouvertes plutôt que sur des technologies spécifiques à des fournisseurs et permet ainsi une interopérabilité répartie et des applications à couplage faible.
- **Résilience aux changements:** Il est peu probable que les futures modifications de l'infrastructure SOAP aient un impact sur les applications utilisant la méthode, tant qu'aucune modification de sérialisation significative n'est apportée à la spécification SOAP.

Il y a cependant plusieurs aspects de SOAP qui peuvent être considérés comme désavantageux. Ceux-ci incluent les points suivants:

- SOAP est sans état, ce qui implique que l'application demandeuse doit se réintroduire dans d'autres applications si des connexions supplémentaires sont nécessaires comme si elle était connectée pour la première fois.
- SOAP sérialise par valeur et ne se sérialise pas par référence.

- SOAP était au début principalement basé sur HTTP. Cela a imposé une architecture de requête/réponse qui ne convenait pas à toutes les situations. Comme HTTP est relativement lent, les performances de SOAP ont été affectées.

3.4.3. Description universelle, protocole de découverte et d'intégration (UDDI)

La description du service Web concerne la spécification de toutes les informations nécessaires pour accéder et utiliser un service. La description doit être riche et contenir à la fois des aspects fonctionnels et non fonctionnels du service, mais peut également contenir des informations sur les processus internes du service, selon que le fournisseur de services ou le propriétaire décide ou non d'exposer ces informations. Les descriptions de services doivent également être rédigées dans un langage formel et bien défini, permettant le traitement automatisé et la vérification des documents de description produits. Ces caractéristiques sont également très utiles lors de la tentative de composition de services Web.

UDDI (*Universal Description, Discovery and Integration Protocol*) (Baida et al. 2004) fournit un mécanisme permettant aux fournisseurs de services d'enregistrer leurs services Web et d'autre part aux consommateurs de services de les trouver. UDDI définit un modèle de données standard et une API pour les registres de services Web. Le modèle de données est utilisé pour stocker dans le registre des informations sur les entreprises ou les services fournis, par ces entreprises ainsi que des informations techniques sur ces services sous forme de descriptions WSDL. Cinq types de structure de données sont définis dans le cadre du modèle de données:

- ***businessEntity*** décrit une entreprise ou une entité prestataire de services en termes de nom, de description et de type d'activité exercée.
- ***businessService*** contient des informations sur le service fourni par *businessEntity*, y compris le nom, la description et les informations sur la façon d'accéder au service dans le cadre de l'élément *bindingTemplate*.
- ***bindingTemplate*** fournit des informations détaillées sur un service incluant l'adresse du point d'entrée pour accéder au service.

- ***tModel*** est la partie du modèle de données utilisée de manière intensive pour la recherche. Il collecte les informations identifiant de manière unique la spécification du service.

Un registre UDDI peut être consulté en utilisant l'API de registre UDDI. Cette API fournit des moyens de programmation pour publier des services, accéder et interroger le registre. Les méthodes de l'API sont regroupées en six ensembles :

- L'**API d'interrogation UDDI** contient des méthodes pour interroger le registre (par exemple *find_service*, *find_business*, *find_binding*, etc.) et des méthodes pour extraire des informations détaillées sur des entités spécifiques (par exemple *get_serviceDetail*, *get_bindingDetail*, etc.).
- L'**API éditeurs UDDI** contient des méthodes pour ajouter, modifier et supprimer des entités dans le registre. Des méthodes telles que *save_business*, *save_service*, *save_tModel*, *etc*. sont utilisées pour ajouter une entité à un registre UDDI, tandis que des méthodes comme *delete_business*, *delete_service*, *delete_business*, *etc*. sont utilisées pour supprimer des entités du registre.
- L'**API de sécurité UDDI** contient des méthodes pour récupérer et rejeter les jetons d'authentification utilisés pour communiquer avec le registre (*get_authToken*, *discard_authToken*, etc.).
- L'**API garde et transfert de propriétés** contient des méthodes permettant le transfert de propriétés et de structures d'un éditeur à un autre. Méthodes telles que *get_transferToken* et *transfer_custody* font partie de cette API.
- L'**API d'abonnement UDDI** contient des méthodes qui permettent la surveillance des modifications dans un registre. Les méthodes suivantes font partie de l'API d'abonnement: *get_subscriptionResult*, *get_subscriptions*, *save_subscriptions*, etc.
- L'**API de réplication UDDI** contient des méthodes qui permettent la réplication des informations entre les registres, afin que les différents registres puissent être synchronisés.

WSDL, même dans sa deuxième version, reste uniquement un langage pour la description syntaxique des services Web. Cependant, cela pose de sérieuses limitations aux applications d'architecture orientée services. Pour illustrer ce fait, examinons le scénario suivant. Un composant d'application ayant besoin d'un service spécifique décrit explicitement ce besoin d'une manière formelle, concise et compréhensible par la machine et charge un service d'annuaire de répondre à ce besoin. Le service d'annuaire tente ensuite de faire correspondre la description du service nécessaire avec les descriptions des services dont il a connaissance. Une liste de correspondances est compilée et un service candidat est sélectionné et est automatiquement lié au composant d'application demandeur. Ce scénario n'est pas réalisable car les descriptions de service fournies par WSDL sont inadéquates et incapables de traiter de tels cas, en raison de l'absence de support de composition.

3.5. Caractéristiques des services Web

Papazoglou (Papazoglou, 2008) définit les caractéristiques suivantes des services Web:

- **Basé sur XML** :
  Les services Web s'appuient sur XML pour la représentation et le transport des données. L'utilisation de XML évite tout lien réseau, système d'exploitation ou plate-forme.
- **Couplage faible :**
  Il n'y a pas de lien direct entre un service Web et son utilisateur. L'altération de l'interface du service Web ne détériore pas la capacité de l'utilisateur à interagir avec ce dernier. Alors que dans un système étroitement couplé, la logique client-serveur est étroitement liée l'un à l'autre, la mise en œuvre d'une architecture faiblement couplée facilite la gestion du système logiciel et facilite l'intégration de différents systèmes.
- **Capacité d'être synchrone ou asynchrone**
  Les liens entre le client et l'exécution du service sont appelés synchronicité. Dans les invocations synchrones, le client envoie sa requête puis attend la réponse sans pouvoir exécuter d'autres opérations pendant cette période. En revanche, les invocations asynchrones permettent aux clients

d'invoquer un service puis exécutez immédiatement d'autres opérations sans attendre le résultat (Modèle "*fire* and *forget*"). La capacité asynchrone est un facteur crucial pour rendre possible des systèmes faiblement couplés.

- **Prend en charge les appels de procédure distante (RPC) :**
  Les services Web permettent aux clients d'appeler des méthodes et des opérations sur des objets distants en utilisant un protocole basé sur XML (SOAP). Les paramètres d'entrée/sortie qu'un service Web doit prendre en charge sont rendus disponibles via des procédures à distance. Un service Web prend en charge/implémente RPC soit en fournissant ses propres services, équivalents à ceux d'un composant traditionnel, soit en traduisant les invocations entrantes en une invocation d'un composant EJB ou .NET.

- **Supporte l'échange de documents :**
  Un aspect central de XML est qu'il est capable de représenter des données, simples et même des documents complexes d'une manière générique. Cet échange transparent de documents supporté par les services Web contribue à faciliter l'intégration métier.

3.6. Types de services Web

Les services Web peuvent être classés selon leur type d'invocation, leur description ou encore leur granularité.

1. **Selon le type d'invocation et le style de communication :** On distingue trois grands courants :

   (a) Services Web basés sur SOAP : SOAP est un protocole basé sur l'échange de messages XML pour les services Web. C'est le type de services le plus souvent utilisé dans l'industrie des sociétés commerciales du fait qu'elles aient construit de nombreux outils pour appuyer leur développement.

   (b) Services Web de type RPC :

   RPC trouve ses racines dans le mouvement des objets distribués. Il y a plusieurs décennies qu'il a commencé à remplacer la messagerie d'objet en mémoire avec la messagerie d'objet inter-réseau dans les applications orientées objet. RPC considère le serveur comme un ensemble de procédures

et le client appelle ces procédures pour exécuter une tâche. Ce genre d'implémentation de service, même si elle est assez commune, n'est pas très recommandé car l'invocation des services est souvent liée au langage d'implémentation. De ce fait, Le code du client est étroitement couplé avec la mise en œuvre du serveur.

(c) Services Web basés sur REST :

Proposée par Roy Fielding (Fielding, 2000) comme un modèle pour le Web, REST est un nouveau mode de pensée pour l'abstraction de service. Cela permet vraiment de comprendre l'aspect original de HTTP et d'utiliser pleinement les fonctionnalités Web actuelles. Les services Web basés sur REST n'ont pas forcément besoin de traiter les documents XML, ils utilisent des formats plus légers comme JavaScript Object Notation (JSON[3]). REST préconise que les services Web fonctionnent dans un style natif du Web. Il rend les informations disponibles sous les mêmes règles qui régissent les sites Web bien conçus. Ainsi, les services Web RESTful sont "dans" le Web au lieu de simplement "sur" le Web. Comparée à l'architecture des services Web de type RPC, l'architecture des services Web RESTful est meilleure en termes d'évolutivité, de couplage et de performances. Avec plus d'outils de développement soutenant REST, il sera de plus en plus adopté et deviendra progressivement la technologie dominante des services Web.

2. Selon leur description :

(a) Services Web Syntaxiques : certains travaux qualifient aussi ces services par le terme "classiques". Ces derniers ne reposent que sur le profil fonctionnel sans intégration d'aucune autre description. La description syntaxique du profil fonctionnel contient les informations relatives aux méthodes proposées, leurs paramètres et les protocoles à utiliser. Ces informations sont nécessaires à l'invocation du service.

[3] JavaScript Object Notation (JSON), disponible à : http://www.json.org/

(b) Services Web Sémantiques : un service Web est dit sémantique si sa description intègre, en plus de la description du profil fonctionnel, une dimension sémantique. Cette dimension englobe toute description complémentaire (du fournisseur, des objectifs du service Web, etc.) de la description fonctionnelle qui permet d'ajouter de l'information à la description classique du service Web. Cette information peut ensuite être traitée par des mécanismes (tels que l'inférence) afin d'en extraire de la connaissance. Notons que l'aspect sémantique n'est pas traité dans le cadre cette thèse.

3. **Selon leur granularité :**

(a) Services Web atomiques : Un service Web est dit atomique s'il ne fait appel à aucun autre service Web lors de son exécution. Il est dit isolé ou seul.

(b) Services Web composites : un service est dit composite ou composé s'il combine les fonctionnalités de plusieurs services Web au sein d'un même processus métier. Souvent il est difficile qu'une requête d'un client soit satisfaite par l'exécution d'un seul service Web. Plusieurs services Web sont composés pour répondre à une demande complexe qu'un seul service ne pourrait satisfaire. La composition des services Web est l'objet de cette thèse et sera examinée au chapitre 3 et au chapitre 4.

3.7. Avantages de l'utilisation des services Web

Toujours selon Papazoglou (Papazoglou, 2008), l'utilisation des services Web a des avantages intéressants tels que:

- **Réutilisation :**

  Un service Web est un composant accessible à distance via HTTP. Les services Web fournissent un moyen de rendre un code préexistant disponible sur Internet. En conséquence, les fonctionnalités du programme peuvent être invoquées par d'autres applications.

- **Interopérabilité :**
  Les services Web permettent le partage de données et la communication entre différents applications. Par exemple, les applications .NET peuvent interagir avec les services Web Java et vice-versa. Ainsi, les applications deviennent indépendantes de la plateforme et de la technologie.

- **Protocole standardisé :**
  Les services Web utilisent le protocole standard de l'industrie pour la communication. Toutes les quatre couches (transport de service, messagerie XML, description du service et couche de découverte) utilisent un protocole bien défini dans la pile de protocoles des services Web. Cette standardisation de la pile de protocoles donne à l'entreprise de nombreux avantages comme large éventail de choix, réduction des coûts dus à la concurrence et augmentation de la qualité.

- **Découverte automatique :**
  Le mécanisme de découverte automatique des services Web permet aux entreprises de trouver facilement des fournisseurs de services et récupérer la description du service Web qui a été précédemment publiée. La première étape pour accéder à un service Web passe par le processus de découverte. Le client interroge le registre des services pour le service Web correspondant à ses besoins. La requête contient des critères de recherche tels que le type de service ou le prix préféré. Après que le processus de découverte est terminé, le client peut accéder au service Web.

4. Paramètres de qualité des services Web (QoS)

Une exigence importante pour une application SOA est de fonctionner de manière fiable et de fournir un service cohérent à différents niveaux. Cela nécessite non seulement de se concentrer sur les propriétés fonctionnelles des services, mais également sur la description de l'environnement hébergeant le service Web, c'est-à-dire décrivant les capacités non fonctionnelles des services. Chaque environnement d'hébergement de service peut offrir différents choix de QoS en fonction des exigences techniques concernant les niveaux de disponibilité, de performance et d'évolutivité, de sécurité et de confidentialité, etc., qui doivent tous être décrits. Il est donc évident que la qualité de

service offerte par un service Web devient la priorité absolue des fournisseurs de services et de leurs clients.

La qualité de service fait référence à la capacité du service Web à répondre aux invocations attendues et à les exécuter au niveau correspondant aux attentes mutuelles de son fournisseur et de ses clients. Plusieurs facteurs de qualité qui reflètent les attentes des clients, tels que la disponibilité constante du service, la connectivité et une grande réactivité, deviennent essentiels pour maintenir une entreprise compétitive et viable, car ils peuvent avoir un impact important sur la fourniture de services. La qualité de service devient ainsi un critère important qui détermine la facilité d'utilisation et l'utilité du service, qui influencent tous deux la popularité d'un service Web particulier, et un point important de différenciation entre les fournisseurs de services Web.

La fourniture de QoS sur Internet est un défi important et crucial en raison de sa nature dynamique et imprévisible. Traditionnellement, la qualité de service est mesurée par le degré dans lequel les applications, les systèmes, les réseaux et tous les autres éléments de l'infrastructure informatique prennent en charge la disponibilité des services à un niveau de performance requis dans toutes les conditions d'accès et de charge. Alors que les métriques de qualité de service traditionnelles s'appliquent, les caractéristiques des environnements de services Web offrent à la fois une plus grande disponibilité des applications et une complexité accrue en termes d'accès et de gestion des services et imposent des exigences spécifiques aux organisations. Dans le contexte des services Web, la qualité de service peut être considérée comme une assurance sur un ensemble de caractéristiques quantitatives. Ces caractéristiques sont des conditions nécessaires pour comprendre le comportement global d'un service afin que d'autres applications et services puissent s'y lier et l'exécuter dans le cadre d'un processus métier.

Les éléments clés pour la prise en charge de la qualité de service dans un environnement de services Web, les plus couramment utilisés par la communauté de chercheurs, sont résumés ci-après et ont été inspirés par (Menascé, 2002).

- **Coût** : désigne le prix que le client doit payer pour chaque invocation du service. Ce paramètre caractérise les services Web commerciaux.

- **Latence** (connue aussi comme **le temps de réponse**) : c'est le temps nécessaire pour un service pour répondre à une requête. Cela comprend, bien évidement, le retard causé par le réseau lors de l'appel du service Web.
- **Débit** : représente le nombre de demandes que le service est en mesure de traiter dans un intervalle de temps donné. Il peut inclure une fonction décrivant comment le débit varie en fonction de l'intensité de charge ou le débit maximal.
- **Fiabilité** (connue aussi comme le taux d'exécution avec succès) : désigne la capacité du service à accomplir sa tâche correctement pendant une période de temps spécifiée. Il peut être mesuré par le temps moyen entre pannes.
- **Disponibilité** : La disponibilité est l'absence de temps d'arrêt de service. La disponibilité représente la probabilité qu'un service soit disponible. Des valeurs plus élevées signifient que le service est toujours prêt à être utilisé, tandis que des valeurs plus petites indiquent une imprévisibilité sur la disponibilité du service à un moment donné. Il peut être exprimé par le ratio : *période active* / *période totale*. La période active représente le temps où le service est opérationnel en répondant aux demandes des clients. La période totale (*période active* + *période inactive*), désigne la durée durant laquelle le client souhaite que le service soit actif.
- **Sécurité** : la sécurité d'un service Web comporte différents aspects assurant que les échanges de messages entre le client et le service soient sécurisés. Plus précisément c'est un regroupement d'un ensemble de qualités à savoir : la confidentialité, le cryptage des messages et le contrôle d'accès.
- **Réputation** : dépend principalement des expériences des utilisateurs finaux. c'est une mesure de la crédibilité du service.

5. Personnalisation des services Web

La personnalisation est définie comme toute action qui adapte les informations ou les services aux besoins d'un utilisateur particulier ou d'un groupe d'utilisateurs. Amazone.com, par exemple, affiche automatiquement l'historique d'achat et de visualisation récent d'un client lorsqu'il se connecte au site Web. Après que le client a ajouté un produit à son panier, le site Web affichera une liste d'autres produits que d'autres clients ont fréquemment achetés avec ce produit, en supposant que ces produits

peuvent également présenter un intérêt. Dans ce cas, la personnalisation est renforcée par l'effort de collaboration.

La plupart des services Web existants ne sont pas conçus et mis en œuvre pour une consommation de services personnalisée. Dans de nombreuses situations, la sélection du service nécessite de prendre les besoins et préférences des consommateurs individuels en considération. Par exemple, les gens peuvent être intéressés par différents types de nouvelles. Un service Web d'alerte de nouvelles devrait être capable d'adapter les nouvelles quotidiennes et sélectionner du contenu ; Cela peut répondre spécifiquement aux besoins individuels. Un autre exemple typique est les services de la publicité ciblée, qui envoient automatiquement des publicités à des clients potentiels, basés sur leurs profils. Apparemment, la compréhension des intérêts individuels est cruciale pour assurer que chaque annonce est envoyée aux clients qui sont susceptibles d'y avoir un intérêt.

Jusqu'à présent, la plupart des services Web sont fournis à différents consommateurs exactement de la même manière. Par exemple, dans le secteur de l'immobilier, les acheteurs de maisons utilisent un service de référencement en ligne, pour rechercher des maisons, sur le marché. En raison des différents besoins et préférences, les intérêts des acheteurs individuels varient selon l'emplacement, la fourchette des prix, la taille de la maison et d'autres facteurs connexes. Il est hautement souhaitable pour les fournisseurs de services de présenter des listes de maisons qui sont automatiquement personnalisées pour les acheteurs individuels. Les services Web personnalisés désignent les services qui peuvent être adaptés aux besoins et aux préférences des consommateurs individuels de services.

Un autre défi dans le domaine des services Web est la composition dynamique des services Web personnalisés. Le travail dans le cadre de cette thèse tente de proposer une approche innovante pour la composition dynamique des services Web en tenant compte des profils utilisateurs.

6. Réseaux sociaux sur le Web

Les réseaux sociaux ont récemment reçu une attention accrue dans le monde de la recherche; Les réseaux sociaux sont des communautés en ligne: personnes, équipes ou

tout groupe social [(Cheong & Corbitt, 2009), (Shekarpour & Katebi, 2010)] interagissant sur le Web et connectés via des relations sociales telles que le travail collaboratif, l'amitié ou l'échange de connaissances dans des contextes variés : divertissement, politique, religion, rencontres ou affaires.

Les réseaux sociaux jouent actuellement un rôle important dans la vie des centaines de millions d'utilisateurs actifs sur Internet; Au cours des dernières années, l'intérêt pour les sites Web des réseaux sociaux tels que Facebook[4], MySpace[5], Friendster[6], Twitter[7] et LinkedIn[8] ainsi que leur popularité ont considérablement augmenté [(Arroyo et al., 2010), (Gürsel & Sen, 2009)]. Les réseaux sociaux ont joué un rôle important dans la révolution sociale au Moyen-Orient, de la Tunisie à l'Égypte, du Yémen à la Libye et à la Syrie.

Dans les réseaux sociaux, les utilisateurs sont amenés à contacter d'autres utilisateurs, créer et publier leur propre contenu sous forme d'opinions, de vidéos ou de photos, partager ce contenu avec d'autres utilisateurs et évaluer ou commenter le contenu publié par d'autres utilisateurs. Par conséquent, les utilisateurs sont maintenant traités comme de simples ressources Web. Une personne peut créer plusieurs identités et peut même créer des liens entre ces identités (Mislove et al., 2007), chacune de ces identités étant considérée comme un utilisateur distinct. Les réseaux sociaux du monde réel évoluent avec le temps; les individus et leurs liens sociaux changent constamment.

L'importance des réseaux sociaux peut être soulignée de plusieurs façons (Arroyo et al., 2010); par exemple, les utilisateurs peuvent bénéficier de leurs interactions avec d'autres utilisateurs pour localiser des informations pertinentes, ou ils peuvent rechercher dans les liens existants. De nombreux utilisateurs des réseaux sociaux indiquent qu'ils ont été en mesure d'obtenir un emploi grâce à leurs interactions et connexions dans LinkedIn. En outre, les réseaux sociaux facilitent la diffusion de nouvelles informations et connaissances de manière généralisée, diffusent des innovations et diffusent des points de vue et des opinions, tels que des messages

[4] http://www.facebook.com
[5] http://www.myspace.com
[6] http://www.friendster.com
[7] http://twitter.com
[8] http://www.linkedin.com

politiques et sociaux, entre les membres et annoncent de nouveaux services et produits [(Arroyo et al., 2010), (Kleinberg, 2008)].

Les réseaux sociaux, dans une large mesure, reflète la confiance sociale entre les individus dans un groupe social. Intuitivement, face à des choix écrasants sans connaissances spécifiques du domaine, un utilisateur se référera généralement à des amis pour des opinions sur un problème spécifique (Liu & Yuan, 2010). Récemment, l'approche basée sur la confiance pour faire des recommandations a émergé. Cette approche exploite le réseau de confiance entre les utilisateurs et génère des recommandations en utilisant les évaluations des individus et leurs relations, telles que les amis directs ou indirects qui sont approuvés par l'utilisateur cherchant une recommandation (Moghaddam et al., 2009). Le réseau de confiance parmi les utilisateurs est très clairsemé (Moghaddam et al., 2009); par conséquent, les approches d'inférence de confiance sont utilisées pour calculer les valeurs de confiance parmi les utilisateurs indirectement connectés au sein du réseau de confiance.

Certains membres des réseaux sociaux ont une influence considérable sur les autres membres et sont connus comme des leaders ou des pionniers (Xu et al., 2010). Newman (Newman, 2003) a utilisé le principe de "l'attachement préférentiel" pour décrire ce phénomène; où les nœuds qui ont beaucoup de liens tendent à être acquis plus. Des recherches récentes ont montré comment l'attachement préférentiel peut conduire certains nœuds à agir comme des nœuds hautement connectés (Kleinberg, 2008).

Dans le cadre de cette thèse nous introduisons les réseaux sociaux de services Web ou les services Web sociaux qui diffèrent des réseaux sociaux conventionnels. Les réseaux sociaux sont basés sur la coopération absolue et l'assistance mutuelle entre leurs membres (c'est-à-dire sans concurrence). Par contre, les services Web dans les réseaux sociaux sont particulièrement compétitifs (Maamar et al., 2009). La construction de tels réseaux nécessite une représentation des différentes interactions dans lesquelles les services Web sont intervenus. Le processus de composition des services Web proposé est basé sur ces interactions.

7. Systèmes de recommandation

Les systèmes de recommandation ont pour but de fournir aux utilisateurs des recommandations sur des éléments pour les personnes ayant des préférences similaires ou qui semblaient avoir préféré dans le passé (Massa & Avesani, 2007). L'idée de base des systèmes de recommandation est de fournir aux utilisateurs les éléments les plus pertinents. Ils sont devenus un domaine de recherche riche depuis les années 1990, lorsque le premier système de recommandation, Tapestry, a été développé, depuis lors, beaucoup de travail a été fait dans l'industrie et le milieu universitaire pour élaborer de nouvelles approches de recommandation. Les approches de recommandation classiques sont généralement classées en trois catégories: recommandation basée sur le contenu dans laquelle l'utilisateur lui est recommandé des éléments similaires à ceux qu'il préférait auparavant, recommandation basée sur le filtrage collaboratif dans laquelle l'utilisateur lui est recommandé des articles sélectionnés par des utilisateurs partageant les mêmes préférences, et hybrides qui combinent les méthodes collaboratives et basées sur le contenu. Cette dernière catégorie permet d'éviter les carences des approches basées sur le contenu et collaboratives et intègre les avantages de ces deux méthodes.

Au cours des dernières années, les services Web et les systèmes de recommandation ont été des domaines de recherche actifs. Le mariage entre ces deux concepts a conduit à l'application de systèmes de recommandation dans le domaine des services Web. Ainsi, nous parlons maintenant des systèmes de recommandation de services Web qui sont très utiles, en particulier que les moteurs de recherche de services Web disponibles ont une performance de recommandation médiocre. En fait, ces moteurs de recherche ignorent les caractéristiques non fonctionnelles des services Web (Yao et al., 2015) et, en les utilisant, les utilisateurs doivent introduire des requêtes correctes car elles sont basées sur des mots clés. Investissant dans le domaine des services Web, les systèmes de recommandation actuels se concentrent principalement sur la découverte et la sélection de services Web. Nous assistons à un large éventail d'articles sur la recommandation de service Web principalement la découverte de services Web [(Ma et al., 2012),(Xu et al., 2007)] et la sélection [(Zibin et al., 2009), [(Zhao et al., 2010)]. Mais, il est intéressant de noter qu'il n'y a pas un nombre relativement important de recherches consacrées aux systèmes de recommandation pour

la composition des services Web. Un tel système de recommandation peut être très utile à la communauté, en particulier depuis le début du Web 2.0, les utilisateurs finaux sont de plus en plus impliqués dans le processus de composition.

8. Conclusion

Ce chapitre a abordé les bases des technologies des services Web. Il a commencé par introduire les paradigmes qui gagnent de plus en plus d'élan dans le milieu universitaire et l'industrie, à savoir l'informatique orientée services (SOC) et les architectures orientées services (SOA). Ils sont centrés autour de la notion de service et constituent une nouvelle façon de concevoir des applications distribuées, soutenant l'interopérabilité et l'intégration à grande échelle. Nous avons d'abord introduit des définitions de concepts et de notions utilisées dans le domaine des services Web. Une distinction importante que nous avons faite était entre un service, qui représente une provision de valeur dans un domaine, et un service Web, c'est-à-dire une entité de calcul accessible sur Internet.

Le paradigme de l'informatique orientée service permet au concept de logiciel en tant que service "*software-as-a-service*" de s'étendre pour inclure la fourniture de processus métiers complexes en tant que service, tout en permettant de construire des applications à la volée et de réutiliser des services partout et par n'importe qui. Les services sont des éléments de calcul ou de traitement auto-descriptifs, indépendants de la plate-forme, qui prennent en charge la composition rapide, économique et simple d'applications distribuées faiblement couplées.

L'approche des services Web repose sur un couplage faible entre un demandeur de service et un fournisseur de services. Cela signifie que le demandeur de service n'a aucune connaissance des détails techniques de la mise en œuvre du fournisseur, tels que le langage de programmation, la plate-forme de déploiement, etc. Le demandeur de service appelle généralement des opérations au moyen de messages - un message de demande et la réponse - plutôt que par l'utilisation d'API ou de formats de fichiers. Cela

est possible car les services Web distinguent la partie de l'interface de service, qui est définie de manière neutre, indépendante de la partie implémentation d'un service Web.

La clé du concept de services Web est l'architecture orientée services. La SOA est un moyen logique de concevoir un système logiciel pour fournir des services à des applications d'utilisateurs finaux ou à d'autres services distribués dans un réseau, via des interfaces publiées et détectables. La SOA de base définit une interaction entre agents logiciels en tant qu'échange de messages entre des demandeurs de services (clients) et des fournisseurs de services. Les clients sont des agents logiciels qui demandent l'exécution d'un service, tandis que les fournisseurs sont des agents logiciels qui fournissent le service. Les agents peuvent être simultanément des clients de service et des fournisseurs.

Nous avons également discuté en détail les trois technologies piliers pour les services Web: WSDL, SOAP et UDDI. Le langage utilisé pour décrire les services Web est WSDL, notamment les systèmes de type utilisés dans les définitions d'interface, les échanges de messages par un service, les opérations qu'il expose et la liaison au protocole réel à utiliser pour l'invocation. D'un autre côté, SOAP peut être vu comme le protocole de communication/messagerie pour les services Web. SOAP spécifie un moyen standard pour coder les messages transmis entre les services Web. Il peut utiliser différents protocoles de transport, tels que HTTP ou SMTP, spécifiés en tant que liaisons. Enfin, nous avons discuté de l'UDDI comme approche de découverte pour les services Web. Le registre UDDI définit la structure d'un registre de services Web et d'API pour accéder, publier, interroger et modifier les entités de registre.

Actuellement, la plupart des architectures de services Web adoptent RPC comme style architectural. Mais en raison de la complexité de RPC, il existe des goulots d'étranglement des services Web de type RPC dans les applications Web. REST peut non seulement utiliser pleinement les fonctionnalités Web, mais aussi l'avantage de la simplicité. REST devient ainsi une nouvelle alternative à RPC pour l'architecture des services Web. Les services Web basées sur SOAP est le type de services Web retenu dans le cadre de cette thèse, bien que rien n'empêche d'opter pour les services Web

RESTful, dans une extension possible de nos travaux. Un aperçu des technologies clés: la qualité de service, les réseaux sociaux basés sur le Web, la personnalisation des services Web et les systèmes de recommandation sont ensuite présentées.

Chapitre 2

Sélection des services Web

Sommaire

1. Introduction 44
2. Définition 46
3. Sélection des services Web sociaux 47
   - 3.1. Approches basées sur la réputation 48
   - 3.2. Approches basées dur la recommandadtion 51
   - 3.3. Approches basées sur le renvoi 54
4. Sélection des services Web et défis des approches basées sur les réseaux sociaux 55
   - 4.1. Problème de démarrage à froid 55
   - 4.2. Problème de confidentialité 56
   - 4.3. Evolutivité 57
   - 4.3. Capture des préférences et des évaluations des utilisateurs 57
5. Etude comprative des différentes approches de sélection des services Web sociaux 59
6. Conclusion 61

1. Introduction

Les composants de l'architecture de référence des services Web SOA sont WSDL (langage de définition de services Web), SOAP (protocole d'accès simple aux objets) et UDDI (découverte et intégration de descriptions universelles), utilisés pour décrire des services, transférer des messages et référencer des services, respectivement. Le mécanisme des services Web est séparé en découverte, sélection et composition. La découverte de services Web permet aux fournisseurs de publier des descriptions de services et des informations de profils concernant les entreprises, les services et d'autres détails connexes dans les registres UDDI. Cependant, il existe des cas dans lesquels nous devons utiliser des propriétés non fonctionnelles et sélectionner le service le plus approprié afin de répondre aux besoins des utilisateurs. La sélection est utilisée pour atteindre ce but. Enfin, un service Web composite pourra être crée à partir d'un ensemble de services Web candidats pour répondre aux exigences des utilisateurs, dans les délais requis par ces derniers (Figure 2.1).

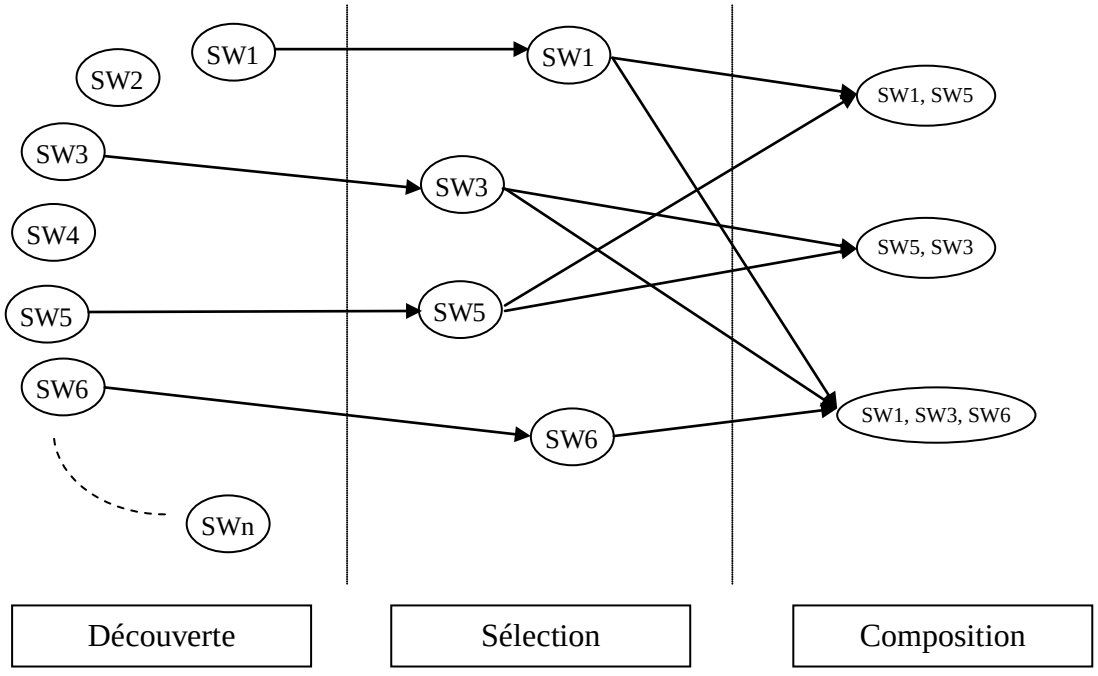


Figure 2.1 : Découverte, sélection et composition des services

La sélection de services Web est un processus complexe, dans lequel le "meilleur" service correspondant aux préférences de l'utilisateur est sélectionné parmi un ensemble de services candidats généralement fournis par un processus de découverte de services basé sur les besoins de l'utilisateur (Vitvar et al., 2007). Les utilisateurs expriment généralement leurs critères de sélection en fonction de propriétés non-fonctionnels dites qualité de services (QoS).

Un grand nombre de services Web fonctionnellement équivalents (ou similaires) ont été construits et déployé, les clients sont confrontés à une tâche difficile en choisissant le "meilleur" service pour construire leur service composite qui répond à leurs exigences personnalisées de QoS. Ainsi, l'efficacité associée à la sélection d'un service Web avec une garantie QoS est devenu de plus en plus un problème critique dans le processus de sélection du service Web (Wang et al., 2010).

Ce chapitre est consacré à présenter les différentes approches de sélection des services Web dans le contexte des réseaux sociaux. L'approche de composition des services Web personnalisés présentée au chapitre suivant (Chapitre 4), qui constitue le cœur de la contribution dans le cadre de cette thèse repose sur la sélection des services Web dans le contexte des réseaux sociaux. Les services Web et les réseaux sociaux sont deux notions émergentes dans lesquelles les applications peuvent rechercher et composer des services en fonction des besoins des utilisateurs de manière transparente et automatique (Malik et al., 2009). On s'attend à ce qu'un large éventail de fonctionnalités semblables soient offertes par un grand nombre de services Web. On s'attend également à ce que des services Web externalisent certaines de leurs fonctionnalités à d'autres services Web (Malik et al., 2009). Dans de telles situations, certains services peuvent être nouveaux sur le marché, et d'autres peuvent se comporter de manière malveillante pour être sélectionnés. Ainsi, pour garantir une sélection et une récupération de qualité des services Web appropriés, il est essentiel de prévoir des mécanismes garantissant la fiabilité du service sélectionné.

La sélection et la recommandation de services Web deviennent un enjeu crucial pour répondre au besoin de fournir aux utilisateurs un service crédible qui répond le plus probablement à leurs besoins [(Metrouh & Valérie, 2013), (Metrouh & Mokhati,

2013), (Metrouh & Mokhati, 2016), (Metrouh & Mokhati, 2018)]. De nombreux défis de recherche se profilent dans la quête d'une sélection et d'une recommandation automatiques de services basées sur les réseaux sociaux. Des méthodes innovantes pour automatiser la sélection des services Web et la recommandation basée sur le comportement des utilisateurs dans les réseaux sociaux sont donc nécessaires.

2. Définition

La sélection d'un service Web est un processus complexe dans lequel le service qui satisfait le mieux les préférences de l'utilisateur est sélectionné parmi un ensemble de services candidats, généralement renvoyés à partir d'un processus de découverte de services basé sur les exigences spécifiques de l'utilisateur (Vitvar et al., 2007).

La sélection de services Web est une tâche compliquée, en particulier si elle prend en considération différentes propriétés non fonctionnelles. La sélection de service implique :

1. la spécification des exigences fonctionnelles et non fonctionnelles de l'utilisateur,
2. la correspondance des offres de service par rapport aux besoins des utilisateurs,
3. la sélection des meilleurs résultats d'agrégation et d'évaluation (Vitvar et al., 2007).

Le coût d'un service est un facteur clé qui influence la sélection d'un service par un consommateur de service (Wang & Vasilva, 2007). Un service, de haute qualité, coûte généralement plus cher qu'un service de qualité inférieure avec les mêmes fonctionnalités. Ainsi, un processus de sélection de services doit équilibrer le coût du service et la qualité de ce dernier.

Selon les critères de sélection, un nombre varié de propriétés non fonctionnelles ou d'attributs QoS peuvent être utilisés en tant que préférences de l'utilisateur. De tels attributs QoS peuvent appliquer des restrictions sur le comportement et la fonctionnalité des descriptions de services (Vitvar et al., 2007), c'est-à-dire que les préférences QoS peuvent contrôler l'utilisation du service. Alors que la concurrence entre les fournisseurs

de services augmente de façon spectaculaire, les consommateurs font face à un grand nombre de services disponibles; ceci a conduit à un problème appelé problème de surcharge d'information (Kavitha & Venkatesh, 2010).

Un nombre considérable de travaux ont été consacrés à l'étude de divers aspects de la sélection des services Web sociaux (sélection des services Web basée sur les réseaux sociaux), ainsi que de la recommandation de ces derniers. Ces efforts varient dans leurs dimensions (Billionniere et al., 2009); sélection de service avec ou sans QoS, sélection de service basée sur la réputation (Papazoglou et al.,2008), ou sélection de service basée sur la confiance.

3. Sélection des services Web sociaux

La sélection des services sociaux est la recherche des services Web souhaités en utilisant les informations relatives à d'autres utilisateurs en fonction de leurs expériences (Billionniere et al., 2009). Sélection de services sociaux, parfois appelée navigation sociale, se caractérise par sa personnalisation, ou comment les gens agissent dans un espace, ainsi que le dynamisme. Dans (Adomavicius & Tuzhilin, 2005a), La personnalisation est définie comme la capacité de fournir du contenu et des services adaptés aux individus en fonction de leurs préférences et de leur comportement, ainsi que l'utilisation de la technologie et des informations sur le client pour adapter les interactions du commerce électronique entre une entreprise et chaque client.

Maximilien et Singh (Maximilien & Singh, 2005) indiquent que les services Web représentent des relations reflétées dans leurs interactions avec d'autres participants tels que le consommateur de services et le fournisseur de services. Ils considèrent les services Web comme des agents permettant aux utilisateurs d'augmenter les styles d'interaction des services Web entre les agents consommateur de services et les agents fournisseur de services. Les agents peuvent assurer la médiation, la négociation, la communication et la collaboration entre les parties concernées, c'est-à-dire les services Web et les consommateurs, en utilisant des interactions de type confiance et référence (Maximilien & Singh, 2005). De plus, Maamar et al. (Maamar et al. 2011) identifient

les interactions de collaboration, de substitution et de concurrence entre les services Web impliqués.

Selon Singh et Huhns (Singh & Huhns, 2005), la sélection des services sociaux repose sur trois approches principales:

1. L'approche fondée sur la réputation regroupe les évaluations des consommateurs de services auprès d'une source centrale pour faire des recommandations aux utilisateurs en termes de qualité.
2. L'approche de recommandation est utilisée pour faire une prédiction des intérêts d'un utilisateur, éventuellement basée sur des utilisateurs ayant des caractéristiques ou des comportements similaires.
3. L'approche basée sur le renvoi utilise la confiance parmi les utilisateurs pour obtenir les évaluations des utilisateurs sur les services.

Chaque approche peut utiliser ou non la qualité de service. De plus, chaque approche a des points forts et des points faibles pour certaines situations et certains contextes (ceux-ci seront abordés dans la section suivante); par conséquent, lorsque l'utilisateur est incapable de prendre une décision concernant la sélection d'un service Web parmi un grand nombre de services, il s'appuie sur une approche de sélection de service basée sur les réseaux sociaux. La section suivante passe en revue les trois approches de sélection des services sociaux.

3.1. Approches basées sur la réputation

La réputation représente une perception collective sur un service Web par des utilisateurs dans un réseau social. La réputation d'un service Web est une évaluation collective des utilisateurs qui ont déjà utilisé le service, ou interagi avec (Wang et al., 2014a). La réputation reflète ce qu'on croit généralement à propos d'un caractère ou d'un comportement d'une entité (Jøsang et al., 2007). Un système de réputation collecte, distribue et regroupe les commentaires des autres membres sur le comportement de chaque membre. Les systèmes de réputation ont été utilisés pour prédire la fiabilité des fournisseurs de services (Jøsang, 2008).

Khosravifar et al. (Khosravifar et al., 2009) proposent un mécanisme traitant l'évaluation de la réputation des communautés de services Web. Le mécanisme proposé définit les métriques d'évaluation impliquées dans l'évaluation de la réputation d'une communauté, et supervise le système de journalisation afin de vérifier la validité et la solidité des feedback fournies par les utilisateurs. Limam et Boutaba (Limam & Boutaba., 2008) proposent un QoS automatisé et un framework basé sur la réputation pour la sélection des services. L'algorithme de sélection prédit l'adéquation d'un service donné aux exigences de qualité de l'utilisateur et la conformité de la qualité livrée à l'offre initiale.

Malik et al. (Malik et al., 2009) présentent une approche pour l'évaluation de la réputation dans les environnements orientés services. Ils définissent des métriques clés pour agréger les feedback des différents évaluateurs, afin d'évaluer la réputation d'un fournisseur de services. Dans les situations où les feedback des évaluateurs ne sont pas facilement disponibles, ils utilisent un modèle de Markov caché (MMC) pour prédire la réputation d'un fournisseur de services. Les modèles de Markov cachés ont fait leur preuve dans de nombreux domaines de recherche pour la modélisation des systèmes dynamiques. Ils proposent d'émuler le succès de tels systèmes pour évaluer la réputation des services afin de permettre des interactions basées sur la confiance avec et entre les services Web.

Les mécanismes de sélection de service basés sur la qualité de service jouent un rôle essentiel dans les architectures orientées services. Les approches standard dans le domaine du e-commerce sont généralement basées sur la prédiction des performances des services à partir de la qualité annoncée par les fournisseurs ainsi que le feedback des utilisateurs sur les niveaux réels de QoS qui leur sont fournis. La question clé dans ce contexte est de détecter et de traiter les fausses notes attribuées par les fournisseurs et les utilisateurs malhonnêtes. Vu et al. (Vu et al., 2005) propose une solution de sélection et de classement de services Web sémantiques basés sur la qualité de service avec l'application d'une méthode de gestion de confiance et de réputation pour résoudre ce problème. Sherchan et al. (Sherchan et al., 2006) proposent une approche basée sur la logique floue pour analyser le comportement de notation des utilisateurs afin d'en déduire la logique des évaluations dans un environnement de services Web. Cette

inférence de la logique facilite au système la validation des évaluations, détecter la tromperie et la collusion, identifier les préférences des utilisateurs et fournir des recommandations à ces derniers.

Dans les systèmes de réputation centralisée, un centre de réputation recueille des informations sur les participants au marché ou la communauté en utilisant les commentaires d'autres consommateurs. Le centre de réputation fournit également un mécanisme intégré de retour d'informations par l'intermédiaire d'une évaluation du produit ou du service [(Dieberger et al., 2000), (Lin et al., 2005)]. Le centre de réputation calcule pour chaque membre un score de réputation et met tous les scores à la disposition du public (Pujol el al., 2002). Les membres peuvent ensuite utiliser les scores des autres pour prendre des décisions; ainsi, les entités qui attirent le plus de membres réputés sont susceptibles de produire des résultats favorables (Jøsang, 2008).

Wang et Vassileva (Wang & Vassileva, 2007) identifient les problèmes suivants dans les systèmes basés sur la réputation:

1. Les utilisateurs sont généralement réticents à donner des notes négatives ; ils ont peur de la vengeance.
2. Un utilisateur peut facilement changer d'identité, surtout s'il a une mauvaise réputation.
3. La réputation d'un utilisateur est représentée par une seule valeur numérique, ce qui rend difficile la visualisation des différents aspects de la réputation tels que la sécurité ou la confidentialité.
4. Le système calcule la réputation en traitant toutes les notes de manière égale sans tenir compte de la crédibilité des évaluateurs.

Xiong et Liu (Xiong & Liu, 2004) mettent en évidence les problèmes suivants dans la plupart des systèmes basés sur la réputation:

1. ne peuvent pas faire la différence entre une feedback honnête et malhonnête,
2. ne supportent pas le contexte dans l'évaluation de la fiabilité,
3. souffrent d'un feedback insuffisant ; il n'y a pas d'incitation pour les pairs à fournir des commentaires,
4. sont incapables de faire face à la personnalité dynamique des pairs; certains pairs agissent malicieusement après avoir bâti une bonne réputation.

Peut-être en raison de la simplicité, la majorité des frameworks proposés dans la littérature [(Chen et al., 2003), (Day & Deters, 2004), (Kalepu et al., 2004)] impliquaient un ensemble fixe d'attributs QoS à mesurer. La justification de la sélection d'un ensemble fixe n'a jamais été abordée de façon adéquate, si ce n'est en indiquant que les études précédentes les avaient souvent sélectionnées, bien que les attributs choisis semblent raisonnables pour les services. Les attributs de service communs sélectionnés sont les suivants: réputation, disponibilité, temps d'exécution, coût et bande passante. Il serait possible, mais pas parfait, d'ajouter un autre ensemble d'attributs généraux à cette liste, tels que les évaluations diverses que les utilisateurs pourraient utiliser pour leurs cotes propres à l'entreprise (Billionniere et al., 2009).

Étant donné que les techniques de recommandation sont personnalisées par définition, et en raison de l'approche distribuée des techniques basées sur le renvoi, les techniques de réputation pourraient également bénéficier du stockage de l'historique des évaluations spécifiques à l'utilisateur (Kalepu et al., 2004). Généralement, les classements moyens pour chaque attribut QoS sont stockés. Kalepu et al. (Kalepu et al., 2004) suggèrent de manière convaincante que chaque classement moyen des utilisateurs pour chaque service devrait également être stocké et pris en compte avec les données de la communauté pour éviter le désaccord avec la majorité.

3.2. Approches basées sur la Recommandation

Les systèmes de recommandation sont la technique couramment utilisée pour la sélection des produits. Il existe diverses catégories de systèmes de recommandation, des systèmes de recommandation basés sur le contenu, collaboratifs et hybrides, dans lesquels chaque système de recommandation a ses propres points forts et ses propres points faibles (Adomavicius & Tuzhilin, 2005b). Il existe de nombreux sites Web de commerce électronique qui utilisent des systèmes de recommandation pour fournir une recommandation à l'utilisateur. Amazon, eBay ont tous des services de recommandation pour alléger le fardeau de la sélection des produits.

La plupart des approches de recommandation de services Web utilisent des évaluations de services Web fondées sur des "opinions subjectives" des consommateurs

de services [(Kerrigan, 2006), (Sherchan, 2006)]. Manikrao et Prabhakar (Manikrao & Prabhakar, 2005) proposent un Framework pour la sélection de services Web en utilisant la correspondance sémantique des besoins de service dans le système de recommandation. Leur approche utilise les commentaires des utilisateurs et le filtrage collaboratif. Le système demande à l'utilisateur d'évaluer le service après qu'il a invoqué le service sélectionné. Cependant, les utilisateurs hésitent à fournir des évaluations explicites (Resnick, 2000).

L'évaluation déloyale est un problème typique dans la plupart des systèmes de recommandation. Néanmoins, il existe certaines approches utilisées pour éliminer les évaluations d'agents malveillants (Xu et al., 2005). L'idée est d'ignorer les notations inférieures à un certain seuil de qualité, ou reçues d'un agent dont la crédibilité est inférieure à un certain seuil de crédibilité. Sherchan et al. (Sherchan et al., 2006) déduisent les évaluations des utilisateurs en fonction de l'analyse du comportement des évaluations des utilisateurs dans un environnement de services Web.

Le filtrage collaboratif [(Herlocker, 2004), (Massa & Avesani, 2004), (Schafer et al., 2007), (Zheng et al., 2009)] est l'une des technologies majeures utilisées dans les systèmes de recommandation pour suggérer des éléments que les utilisateurs pourraient apprécier (Schafer et al., 2007). Le système de filtrage collaboratif typique est centralisé, où un nœud centralisé est chargé de collecter les évaluations auprès des utilisateurs et de les stocker dans une matrice avec une ligne pour chaque utilisateur et une colonne pour chaque élément. Selon Massa et Bhattacharjee (Massa & Bhattacharjee, 2004), l'algorithme de filtrage collaboratif standard comporte trois étapes:

1. La similarité entre un utilisateur donné et tous les autres utilisateurs est calculée, en fonction de la similitude de leurs évaluations sur les éléments qu'ils ont tous classés avant. Le coefficient de corrélation de Pearson est l'algorithme le plus populaire pour mesurer la similarité entre deux utilisateurs.

2. La note prédite d'un utilisateur sur un article qu'il n'a pas évalué est calculée en fonction des notes attribuées par les autres utilisateurs, qui ont évalué l'article pondéré par la similarité entre l'utilisateur et les autres utilisateurs.
3. Les articles ayant les évaluations prédites les plus élevées sont recommandés à l'utilisateur.

Bien que les algorithmes des systèmes de filtrage collaboratif et des systèmes de réputation personnalisés soient similaires, ils sont différents dans leur orientation (Wang & Vassileva, 2007). Les systèmes de filtrage collaboratif mettent l'accent sur la similarité des préférences des utilisateurs, tandis que les systèmes de réputation personnalisés mettent l'accent sur la confiance entre les utilisateurs. Cependant, ils sont tous deux utilisés pour mesurer la fiabilité des opinions donc l'évaluation des autres utilisateurs. Selon certaines définitions, la confiance signifie la similitude des préférences des utilisateurs [(Borzymek et al., 2009), (Golbeck, 2009), (Lee & Brusilovsky, 2009), (Berners-Lee et al., 2001), (Ziegler & Golbeck, 2007)] dans certains contextes. Si deux utilisateurs ont pratiquement les mêmes préférences, ils peuvent se faire davantage confiance. Par conséquent, un système de filtrage collaboratif peut être considéré comme une variante d'un système de réputation personnalisé.

Zheng et al. (Zheng et al., 2009) proposent une méthode de filtrage collaboratif hybride pour la recommandation de services Web, dans laquelle les expériences passées des utilisateurs pour les informations de qualité de service des services Web sont collectées pour prédire la qualité de service des services candidats. En outre, Chiu et al. (Chiu et al., 2009) proposent un modèle de recommandation quantitatif utilisant l'utilité, la réputation, la fiabilité, l'attitude au risque et la persuasion.

Un système de recommandation basé sur le filtrage collaboratif peut faire des recommandations de bonne qualité lorsque le système dispose de suffisamment de données requises, mais il est vulnérable au manque de données, ce qui pose deux problèmes, les problèmes de démarrage à froid et de manque de données discutés au début de la section 4 du présent chapitre.

La recherche liée à la recommandation de services Web apparaît dans des efforts tels que [(Birukou et al., 2007), (Blake & Nowlan, 2007), (Kim & Lee, 2006), (Li et al., 2010), (Li & Kao, 2009), (Manikrao & Prabhakar, 2005), (Zheng et al. , 2009)]; certains de ces efforts traitent les attributs QoS en tant que préférences de l'utilisateur.

Chung et Rao (Chung et Rao, 2012) présentent une approche de recommandation pour les services de divertissement Internet pour les produits tels que les films et les jeux, en utilisant un modèle de préférence générale pour les consommateurs. Ils calculent l'utilité du produit à partir de deux composants (observés et non observés) en fonction de la disponibilité des attributs QoS; le composant non observé est évalué en utilisant des résidus d'experts virtuels. Leurs résultats montrent des améliorations dans le taux de succès de la prédiction par rapport aux algorithmes collaboratifs.

3.3. Approches basées sur le renvoi

Les approches basées sur le renvoi sont utilisées pour évaluer les services et les renvois afin d'aider les utilisateurs à trouver des experts (Billionniere et al., 2009); des agents qui fournissent des informations sur la crédibilité d'autres agents leur sont référés en tant que témoins, et l'information fournie est appelée confiance de référence. Sabater et Sierra (Sabater & Sierra, 2002) ont proposé comment combiner l'information sur la réputation de différents agents dans des contextes variés; cette approche peut identifier des agents fiables dans chaque contexte. La confiance de référence est présentée dans les "réseaux ouverts" par Beth et al. (Beth et al., 1994).

Dans la réalité, si un consommateur doit éviter une transaction risquée, il peut consulter un ami de confiance, ce qu'il pense de cette transaction. Si cet ami est incapable de fournir une opinion sur cette transaction, il peut demander à un ami de ses amis, et ainsi de suite jusqu'à ce qu'il trouve quelqu'un (i.e. Un recommandateur) qui peut donner une opinion sur cette transaction.

Un système basé sur le renvoi remplit les fonctions suivantes (Billionniere et al., 2009):

– Évaluer les services et les renvois pour aider les utilisateurs à trouver des experts.
– Maintenir des répertoires sur d'autres agents.

– Intégrer l'interaction entre agents (Dieberger et al., 2000).

4. Sélection des services Web et défis des approches basées sur les réseaux sociaux

Bien que l'avenir de la sélection de services Web semble très prometteur, le processus de recommandation comporte encore des problèmes complexes. Certains de ces problèmes ont trait à la connaissance et au comportement des utilisateurs, tandis que d'autres se rapportent aux approches recommandées pour trouver une personne digne de confiance qui prodiguerait des conseils à l'utilisateur cible. Dans cette section, un aperçu sur ces défis est détaillé.

4.1. Problème de démarrage à froid

Dans le filtrage collaboratif, les nouveaux éléments et les nouveaux utilisateurs du système de recommandation ne disposent pas de suffisamment de données de classement et ne peuvent donc pas être utilisés dans la recommandation. Les utilisateurs qui démarrent à froid représentent l'un des défis les plus importants dans les systèmes de recommandation. Un utilisateur débutant à froid est décrit comme un nouvel utilisateur qui s'est joint au système de recommandation basé sur le filtrage collaboratif et qui n'a fourni aucune évaluation, ou qui a fourni un nombre limité d'évaluations. Dans de tels cas, le système de recommandation ne peut pas faire de recommandations réalisables et utiles aux utilisateurs qui démarrent à froid, et ne peut fournir aucune recommandation si l'utilisateur cible n'a pas exprimé d'avis. Pour les mêmes raisons, les approches basées sur la similarité sont incapables de trouver des utilisateurs similaires. Cependant, les approches de recommandations basées sur les réseaux sociaux peuvent générer des recommandations à condition que le nouvel utilisateur soit relié à une grande partie du réseau social.

Bien que certaines solutions aient été rapportées dans la littérature pour le problème de démarrage à froid, la plupart de ces efforts supposent que l'utilisateur actif fournit au moins une évaluation ou que l'utilisateur actif fournisse des informations de profil complètes comme, l’âge, le sexe et la fonction pour saisir les préférences de l'utilisateur

et en déduire leurs similitudes. Cependant, que se passe-t-il si l'utilisateur actif ne fournit pas d'avis ou d'informations personnelles pour capturer ses préférences?

Kavitha Devi et Venkatesh (Kavitha Devi & Venkatesh, 2010) ont abordé le problème du démarrage à froid dans leur modèle en considérant l'utilisateur actif comme un "utilisateur novice", et ils ont fourni à l'utilisateur un ensemble d'éléments avec des évaluations élevées comme recommandation par défaut. Néanmoins, le problème du démarrage à froid reflète plus de la moitié des utilisateurs de la population qui ont besoin d'une solution réalisable.

4.2. Problème de confidentialité

La protection de la vie privée a suscité beaucoup d'attention de la part de l'industrie et du milieu académique au cours des dernières années. Les fournisseurs de services Web offrent des services qui exigent de plus en plus la révélation des informations personnelles des utilisateurs, ce qui pose un problème croissant de confidentialité à leurs utilisateurs. Les fournisseurs de services Web utilisent les informations personnelles de leurs clients pour développer et générer des publicités personnalisées. Certains fournisseurs peuvent également libérer ou vendre les informations de leurs clients à des tiers à des fins diverses. De plus, Resnick et Varian (Resnick & Varian, 1997) indiquent que "les gens peuvent ne pas vouloir que leurs habitudes ou leurs points de vue soient largement connus".

Dans les réseaux sociaux sur le Web, profils d'utilisateurs, préférences, notations, préférences et l'historique des achats sont utilisés comme des atouts précieux pour faire des recommandations sociales, cependant, amélioré les recommandations sociales ont un coût; ils peuvent potentiellement conduire à des violations de la vie privée par révéler les informations sensibles des utilisateurs. Une façon de gérer la confidentialité des utilisateurs sur les sites de réseaux sociaux est pour l'utilisateur de décider spécifiquement qui peut voir leurs informations, déterminer si elle est visible à quelques amis, collègues, membres de la famille ou tout le monde sur le Web en fonction des différentes options d'identité de l'utilisateur.

4.3. Evolutivité

Le problème d'évolutivité est considéré comme un problème de performance. Rana et Stout (Rana & Stout, 2000) définissent l'évolutivité comme: "la capacité d'une solution à un problème de fonctionner lorsque la taille du problème augmente".

Selon Sarwar et al. (Selon Sarwar et al., 2001), les calculs dans les algorithmes de similarité augmentent considérablement avec l'accroissement du nombre d'items et le nombre d'utilisateurs en même temps. Comme il existe des millions d'éléments et d'utilisateurs dans un système de recommandation, ces algorithmes souffrent de graves problèmes d'évolutivité.

En résumé, le calcul de similarité pour une matrice de N utilisateurs et M éléments est un O (N^2M) problème. Étant donné que de nouveaux utilisateurs et / ou de nouveaux éléments entrent fréquemment dans le système, il est obligatoire de mettre régulièrement à jour la matrice de similarité. Par conséquent, les approches fondées sur le filtrage collaboratif souffrent généralement de restrictions d'extensibilité. Bien que divers mécanismes, tels que le clustering des kmeans, aient tenté de résoudre le problème de l'évolutivité, l'extensibilité reste un problème de recherche ouvert dans la plupart des systèmes basés sur les systèmes de filtrage collaboratifs.

4.4. Capture des préférences et des évaluations des utilisateurs

Les propriétés non fonctionnelles du service Web ou la QoS telles que la disponibilité, le temps de réponse, la sécurité, la confidentialité et la fiabilité sont difficiles à déterminer et à contrôler pour l'utilisateur. Les utilisateurs hésitent généralement à décrire leurs préférences complètes et à fournir des évaluations pour le système, et sont encore moins disposés à attribuer des poids relatifs correspondants à leurs préférences, surtout lorsque les conséquences de leurs entrées sont inconnues ou indéterminées. De plus, selon Ziegler (Ziegler, 2007), certains achats sont des cadeaux et ne reflètent donc pas les intérêts de l'utilisateur actif, et les utilisateurs peuvent même ne pas être conscients de leurs préférences implicites.

Les travaux existants ont étudié les préférences explicites d'un utilisateur tels qu'ils sont spécifiées dans son profil, ou leurs préférences implicites indiquées par leurs interactions antérieures avec différents types d'informations sur le Web. Récemment, l'augmentation des réseaux sociaux en ligne a suscité l'intérêt de tirer parti des réseaux sociaux pour inférer les préférences des utilisateurs, sur la base de l'existence de l'influence sociale et de la corrélation entre voisins dans les réseaux sociaux. Pour les applications qui peuvent observer directement les comportements des utilisateurs (c'est-à-dire, les journaux des moteurs de recherche), inférer les préférences des amis dans les réseaux sociaux fournit une amélioration supplémentaire utile. Pour beaucoup d'autres applications, cependant, il est difficile d'observer un comportement suffisant d'un grand nombre d'utilisateurs; dans de telles situations, inférer les préférences de leurs amis peut être la seule solution viable. Par exemple, pour un nouvel utilisateur dans une application sociale, l'application peut seulement avoir des informations sur des amis qui l'utilisent déjà. Pour motiver le nouvel utilisateur à participer activement, l'application peut vouloir fournir des recommandations personnalisées de contenu pertinent; par conséquent, l'application doit déduire des intérêts d'amis, ce qui est problématique si l'utilisateur n'a pas d'amis.

D'un autre côté, les commentaires des utilisateurs peuvent être explicites ou implicites selon le mécanisme utilisé pour capturer la satisfaction des utilisateurs. Bien que les approches explicites captent plus précisément la satisfaction des utilisateurs, ces approches sont coûteuses et souvent les utilisateurs ne coopèrent pas. Resnick et al. (Resnick et al. , 2000) indiquent que «les gens ne se soucient pas du tout de donner leur avis» et qu'il est «difficile d'obtenir des commentaires négatifs» et qu'il est difficile d'obtenir des rapports honnêtes. Par conséquent, les utilisateurs ont tendance à éviter d'indiquer explicitement leurs préférences et à quitter le système ou à compter sur le «free-riding». Cependant, les évaluations explicites sont difficiles à réaliser car elles imposent des frais supplémentaires à l'utilisateur. L'approche de la réputation ne nécessite que des évaluations d'utilisateurs, mais dans certains cas, il peut être difficile de les obtenir. La recherche sur eBay révèle que les notations ne sont soumises que dans 50% des transactions. Cependant, l'exemple d'eBay traite de l'interaction humaine. Avec

des évaluations automatisées susceptibles d'être utilisées avec les services Web, des problèmes d'élicitation peuvent toujours exister en raison de problèmes de confidentialité ou d'utilisateurs désintéressés.

Selon Klein et al. (Klein et al. , 2009), les acheteurs d'eBay commentent les vendeurs 52,1% du temps. Les notes laissées par les acheteurs sont positives dans 98,1% des cas. À partir de ce ratio, nous pouvons conclure que 47,9% des acheteurs ont peur d'être vengés ou ne se donnent pas la peine de donner une note au vendeur. Dellarocas et Wood (Dellarocas & Wood, 2008) ont étudié les conséquences des commentaires manquants non aléatoires sur les scores de réputation; ils ont constaté que les consommateurs insatisfaits sont réticents à fournir des commentaires. Le score de retour est généralement l'indicateur le plus important de la réputation du service Web. Comme un nombre significatif de membres ne participent pas au processus de notation, l'absence de notation peut avoir un impact majeur sur la qualité du système de réputation.

Un client débutant à la recherche d'un service Web, recherche souvent l'aide de ses amis, collègues, experts et partenaires commerciaux ayant une expérience pertinente. La capture et la spécification des préférences de l'utilisateur sont parmi les problèmes les plus complexes du processus de sélection de service. Certaines approches font des sélections basées sur l'ontologie des propriétés non-fonctionnelles ou synthétisent la sélection en utilisant des requêtes précédentes d'une situation similaire. Une alternative proposée par Singh et Huhns (Singh & Huhns, 2005) consiste à obtenir implicitement les évaluations des utilisateurs en déduisant les évaluations des utilisateurs tirées de leurs actions sur le site Web. Des actions telles que la navigation, les visites de retour et l'historique des achats peuvent être utilisées pour déduire les évaluations implicites des utilisateurs. Néanmoins, l'incomplétude des données reste un défi majeur.

5. Etude comparative des différentes approches de sélection des services Web sociaux

Dans cette section, nous proposons une étude comparative des différentes approches de sélection des services et la façon dont chacun des défis (démarrage à froid, parcimonie,

confidentialité, évolutivité et saisie des préférences des utilisateurs) a une incidence sur chacune des approches de sélection (voir table 2.1).

	Réputation	**Recommandation**	**Renvoi**
Portée	Regroupe les évaluations des utilisateurs des services à une source centrale pour faire des recommandations aux utilisateurs en termes de qualité.	Faire une prédiction des intérêts d'un utilisateur basée sur des utilisateurs ayant des caractéristiques ou des comportements similaires.	Utilise la confiance parmi les utilisateurs pour obtenir les évaluations des utilisateurs sur les services.
Opération	– Centralisée	– Centralisée	– Décentralisée
Références	(Khosravifar et al., 2009) (Limam & Boutaba,2008) (Malik et al., 2009)	(Birukou et al, 2007) (Blake et al., 2007) (Chung & Rao, 2012) (Li et al., 2010) (Li & Kao, 2009) (Zheng et al., 2009)	(Billionniere et al., 2009) (Sabater & Sierra, 2002)
Avantages	– Conception simple généralement.	– Trouver des services facilement en raison des suggestions automatisées. – Dans certains cas, peut éviter l'élicitation due au suivi du comportement de l'utilisateur.	– Le système se restructure dynamiquement en fonction de la confiance de ses paires. – Stocke moins de données.
Inconvénients	– Plus vulnérable aux aux représailles. – Difficulté à obtenir des évaluations (notes). – Difficile pour le service de rang inférieur de se promouvoir.	– Nécessite de nombreuses évaluations pour faire une recommandation. – Point de défaillance unique.	– Le système distribué peut être plus lent qu'autonome. – Plus complexe en raison de la nature distribuée.
Démarrage à froid	– Non considéré pour les utilisateurs.	– manque de données implique ne pas trouver de similitudes avec le démarrage à froid. – certains sont traités comme des utilisateurs naïfs.	– Considéré, l'utilisateur doit connaître au moins un ami.

Confidentialité	– Difficile de voir différents aspects de la réputation tels que la sécurité ou la confidentialité.	– Violations de la vie privée en révélant les informations des utilisateurs.	– Non considéré en raison de la nature distribuée.
Evolutivité	– Évolutivité basé sur le nombre de services / produits.	– Puisque le calcul de similarité augmente, l'évolutivité est un défi.	– Le système distribué peut être plus lent qu'autonome.
Préférences et évaluations des utilisateurs	– Implicite ou explicite – Difficulté à obtenir des notes. – La réputation est représentée par une seule valeur numérique – crédibilité des évaluateurs non prise en compte.	– Implicite ou explicite – Difficulté à obtenir des évaluations (notes). – La cote déloyale est un problème typique. – Problème de crédibilité des évaluateurs dans la plupart des approches.	– Plus de transactions et d'utilisateurs nécessaires pour les évaluations. – Problème d'évaluation (note) déloyale. – Problème de crédibilité des évaluateurs.

Table 2.1 : Défis abordés dans les approches basées sur les réseaux sociaux.

La recommandation de services Web diffère de la recommandation des produits. Les services Web ont des attributs QoS qui doivent être pris en compte en plus des attributs fonctionnels lors de la sélection et de la recommandation de services. Bien que l'ajout de QoS améliore la qualité du service recommandé et augmente la satisfaction des utilisateurs, car il fournit plus de précision dans les évaluations; Les systèmes QoS doivent faire face à des problèmes supplémentaires: complexité accrue, stockage de données supplémentaire, surcharges de calcul, augmentation du trafic réseau et interface impliquée pour exprimer les attributs QoS pendant la sélection du service. Ainsi, la prise en compte des attributs QoS est utile pour gérer les problèmes de préférences des utilisateurs lors du processus de sélection.

Il convient également de mentionner que presque tous les travaux entrepris sur les systèmes recommandation ont des approches variées qui traitent la sélection et de la recommandation des services Web, notamment toutes les solutions proposées: démarrage à froid, confidentialité, évolutivité et saisie des préférences des utilisateurs.

6. Conclusion

La sélection des services sociaux est une problématique difficile en théorie et en pratique. Ce domaine de recherche est incroyablement immature; cet argument est

soutenu par le fait qu'il n'existe pas de systèmes automatisés de sélection de service public déployé dans le monde commercial à l'heure actuelle. Bien que certaines approches utilisées pour gérer les services en tant que produits; les services ont des propriétés non fonctionnelles ou des attributs QoS qui les distinguent des produits.

Ce chapitre a brièvement présenté le problème de la sélection et de la recommandation des services sociaux. Trois technologies puissantes peuvent être intégrées pour y remédier ont été identifiées et décrites: les services Web, les réseaux sociaux et les systèmes de recommandation. Ces technologies possèdent les synergies catalytiques nécessaires pour résoudre le problème complexe de la sélection et de la recommandation des services. Les défis et les lacunes de la recherche associés au problème de la sélection des services sociaux et des recommandations dans un environnement dynamique ont été identifiés et discutés.

Dans le cadre de cette thèse, nous nous intéressons en particulier aux réseaux sociaux de services Web qui diffèrent des réseaux sociaux conventionnels. Les réseaux sociaux sont basés sur la coopération absolue et l'assistance mutuelle entre leurs membres (les utilisateurs eux-mêmes). Par contre, les services Web dans les réseaux sociaux sont particulièrement compétitifs. Les services Web sont les seuls constituants du réseau social. Les utilisateurs interagissent avec ces derniers sans qu'ils ne soient membre du réseau social des services Web, en acceptant ou en rejetant un service Web. Il n'y a aucune possibilité de diffuser ce rejet ou même l'acceptation aux autres utilisateurs. Il n'existe pas un moyen de le faire et ainsi préserver la confidentialité de la réputation du service Web, qui encouragera les fournisseurs à enregistrer leurs services Web dans les réseaux sociaux de ces derniers et éviter les opinions subjectives. La recommandation se fait exclusivement entre services Web, dans une même communauté (objet du chapitre 4).

Chapitre 3

Composition des services Web

Sommaire

1. Introduction 64
2. Modèles de composition des services Web 65
   - 2.1 Orchestration 66
   - 2.2 Chorégraphie 67
   - 2.3 Coordination 68
   - 2.4 Modèle de composant 69
3. Exigences de la composition des services Web 69
   - 3.1 Automatisation 70
   - 3.2 Dynamicité 70
   - 3.3 Qualité de service (QoS) 71
   - 3.4 Evolutivité 71
   - 3.5 Préférence des utilisateurs 72
   - 3.6 Indépendance au domaine 72
4. Différentes approches de composition automatique des services Web 72
   - 4.1 Approches orientées Workflow 73
   - 4.2 Approches basées sur les modèles 74
   - 4.3 Approches mathématiques 78

4.4 Approches orientés intelligence artificielle 81
4.5 Approches basées sur la recommandation 84
5. Discussion 88
6. Conclusion 90

1. Introduction

La composition des services Web est à l'origine d'un nombre considérable de travaux, tant dans la recherche académique que dans l'industrie. Malgré tous les efforts, la composition des services Web demeure un problème très complexe. Cette complexité provient du fait que les solutions proposées doivent prendre en considération le nombre croissant de services déployés sur le Web, leur mise à jour continue et leur hétérogénéité (Rao & Su, 2004).

Le plein potentiel des services Web ne peut être atteint que si nous envisageons une collaboration entre une communauté de nombreux fournisseurs de services et les consommateurs de services qui interagissent pour atteindre certains objectifs commerciaux. Ainsi, les services sont combinés en une application métier (composition de services) qui spécifie une logique métier complexe dans laquelle l'exécution de plusieurs services est coordonnée. La composition des services est une force motrice clé de la SOA qui cherche à fournir les moyens de créer rapidement de nouveaux services et applications (services composites) en "composant" les services existants (services atomiques).

Les services Web atomiques ne peuvent pas satisfaire toutes les requêtes de services d'un client. Lorsque cela se produit, il est souhaitable de chercher des possibilités de combiner les services existants pour répondre à une requête donnée. Un exemple typique d'une requête de service consiste à passer une commande à un détaillant en ligne via un site Web. Cette commande peut être remplie par un service composite qui intègre plusieurs services internes et externes tels que le service de vérification de crédit, service de vérification de l'état du stock (par exemple, si le produit commandé est en rupture de stock, un autre service de commande doit être envoyé aux fournisseurs potentiels pour approvisionnement), service d'expédition, etc.

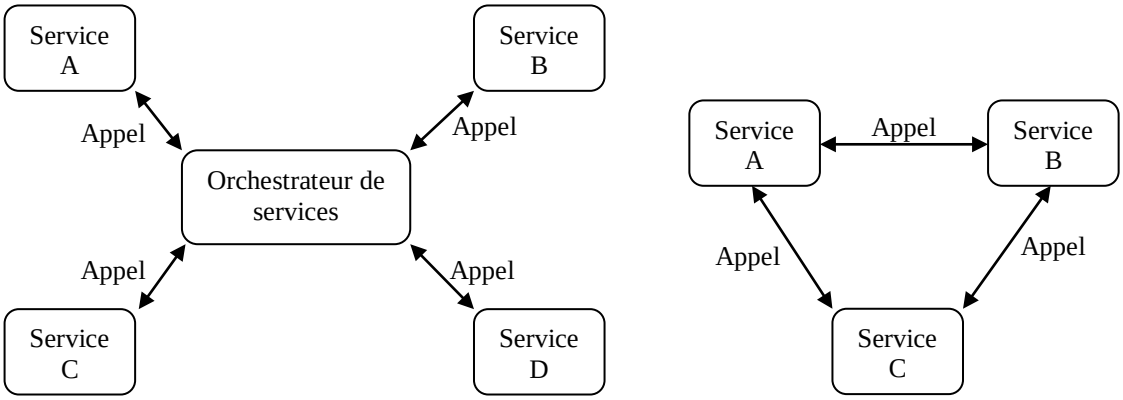
La composition des services Web est de construire des services de plus haut niveau basés sur plusieurs services existants pour répondre à des exigences commerciales plus sophistiquées. La composition de services Web existants en nouveaux services avancés et complexes favorise le développement rapide d'applications, la réutilisation des services et la collaboration interentreprises. La composition des services Web a le potentiel de réduire le temps de développement et les efforts pour de nouvelles applications. Elle fournit un mécanisme pour prendre en charge l'intégration d'applications interentreprises et intra-entreprises. Comme il peut être coûteux et fastidieux d'identifier et de composer des services manuellement, la composition automatique ou semi-automatique de services Web existants pour obtenir de nouvelles fonctionnalités est devenue le centre d'attention actuel. Du point de vue du développeur, la composition du service offre la possibilité d'interaction et de réutilisation. Du point de vue du consommateur de service, la composition automatisée offre l'accès à une grande variété de services complexes avec une intervention manuelle minimale. La capacité à effectuer des compositions des services Web automatisés ou semi-automatisés a permis de révolutionner de nombreux domaines d'application, notamment les applications de commerce électronique et l'intégration de systèmes. Les avantages des services Web et de leurs compositions ne sont pas seulement attrayants pour le paradigme SOC, mais ils sont très attrayants et prometteurs dans le domaine du *Cloud Computing* par exemple, qui est l'un des plus récents paradigmes qui a pris beaucoup d'ampleur au cours des dernières années (Buyya et al., 2010). Il n'est pas surprenant qu'un développement colossal en termes de technologies et de standards SOA vise à résoudre le problème de la composition des services, et c'est l'un des domaines les plus activement étudiés dans l'architecture SOA et le *Cloud Computing*.

2. Modèles de composition des services Web

Avant de présenter les approches les plus importantes proposées dans la littérature pour la réalisation de la composition automatisée des services Web, nous examinerons les modèles de composition les plus connus et les mieux cotés qui ont été définis. Ces modèles sont utilisés par la plupart des approches qui constituent l'état de l'art de la section suivante. Il convient de noter que ces modèles ne sont pas utilisés exclusivement: une approche peut implémenter plus d'un modèle en même temps.

2.1. Orchestration

L'orchestration (voir figure 3.1) est une description de la façon dont les services qui participent dans une composition interagissent au niveau des messages, y compris la logique métier et l'ordre dans lequel les interactions doivent être exécutées (Papazoglou et al., 2007). Ainsi, une orchestration devrait définir quel message est envoyé, quand et par quel service participant ? Une orchestration de service peut être considérée comme proactive, car elle définit les interactions avec les services qu'elle orchestre, avant toute exécution réelle. Hull et Su [(Hull & Su, 2005) font distinction entre la synthèse de composition et l'orchestration. La synthèse de composition consiste à synthétiser une spécification de la façon de coordonner les services composants pour satisfaire la requête d'un client, en générant un plan qui dicte comment obtenir un comportement souhaité en combinant les capacités de plusieurs services. D'autre part, l'orchestration consiste à exécuter le résultat de la synthèse de composition en coordonnant le contrôle et le flux de données entre les services participants, ainsi qu'en supervisant et surveillant cette exécution.



Orchestration de services Chorégraphie de services

Figure 3.1 : Modèles de composition des services Web.

Les orchestrations de services sont généralement décrites et exécutées à l'aide de langages de workflow. Le langage le plus répandu et universellement adopté pour décrire l'orchestration de services est le langage d'exécution de processus métier (WS-

BPEL)[1] . BPEL est un langage basé sur XML pour représenter la logique métier, c'est-à-dire le flux de données et le flux de contrôle des processus. Le flux de contrôle peut contenir des chemins d'exécution alternatifs, la gestion des exceptions et des erreurs, la gestion des événements et des règles et contraintes supplémentaires. En dehors de cela, les processus BPEL contiennent des participants (services Web) affectés à chaque activité contenue dans le processus. Dans son document officiel, seules les descriptions WSDL peuvent être associées à un processus BPEL, qui limite BPEL à des compositions de service statiques et syntaxiques (sans sémantique). Cependant, il y a eu des travaux de recherche qui visent à combler ces lacunes et à soutenir la dynamicité et la sémantique.

2.2. Chorégraphie

La chorégraphie (voir figure 3.1) est associée aux échanges de messages visibles à l'échelle mondiale, aux règles d'interaction et aux accords intervenant entre plusieurs processus métier. C'est leur principale caractéristique de définition, car ils diffèrent de l'orchestration qui spécifie généralement un processus métier exécuté par une seule partie. Dans les chorégraphies, les partenaires contrôlent entièrement leurs processus internes et ne les exposent pas à d'autres partenaires, à moins qu'ils ne soient essentiels à la communication. Les chorégraphies sont conceptuellement liées aux conversations, qui sont définies comme un échange de messages entre deux partenaires qui suit les règles de la chorégraphie globale.

Le principal langage de définition des chorégraphies est le Web Services Chorégraphie Description Langage (WS-CDL). WS-CDL utilise des interactions bilatérales avec un ou deux messages pour modéliser de manière incrémentielle toute interaction. Les blocs de construction de base sont des interactions de requête uniquement, de réponse uniquement et de demande-réponse. Il existe un ensemble limité de constructions de flux de contrôle (séquence, parallèle, choix et unité de travail) afin de représenter la composition des interactions.

Bien que l'orchestration et la chorégraphie puissent être utilisées séparément pour implémenter une composition de services, leurs différents points de vue peuvent être

[1] OASIS : [en ligne] http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.pdf [accédé : 22 Mai 2018]

combinés pour une représentation plus complète. L'orchestration peut être utilisée pour décrire les services participants au niveau d'abstraction inférieur et la chorégraphie peut donner une description plus détaillée de la manière dont ces orchestrations interagissent entre elles et capturent les conversations complexes entre elles afin de réaliser l'objectif fixé par le client. Bien sûr, on peut affirmer que la description de niveau supérieur peut également être réalisée en utilisant un modèle d'orchestration, mais cela impliquerait implicitement que la synthèse de composition soit impliquée pour la création de la description de processus de l'orchestration, ce qui laisse la chorégraphie comme le choix le plus approprié.

2.3. Coordination

La coordination de service implique le regroupement temporaire d'un ensemble d'instances de service suivant un protocole de coordination. Ce protocole dicte la façon dont les services participants interagissent et aussi le résultat de l'interaction, qu'elle soit réussie ou non. La caractéristique déterminante de la coordination est l'existence d'un tiers, appelé coordinateur, qui sert de lien entre les participants et qui est responsable du maintien du protocole et de la décision et de la diffusion du résultat, une fois l'activité terminée. Les services participants ne communiquent généralement pas directement entre eux dans un modèle de coordination, toutes les communications impliquent le coordinateur.

OASIS[2] a développé le WS-CAF[3] (Web Services Composite Application Framework) qui comprend, entre autres, le WS-CF (Web Services Coordination Framework). En WS-CF, une coordination est définie sur la base de trois composantes, le coordinateur, les participants et le service de coordination. Le coordinateur fournit une interface permettant aux participants de s'inscrire, de coordonner les participants selon le protocole et de diffuser les résultats du protocole. Les participants peuvent être des services Web atomiques ou des services composites. Le service de coordination est simplement la mise en œuvre du protocole de coordination.

[2] OASIS : http://www.oasisinet.com.
[3] https://www.oasis-open.org/committees/ws-caf/.

2.4. Modèle de composant

Le dernier modèle que nous allons examiner est le modèle de composant, parfois appelé câblage de services. Le câblage de services implique la liaison réelle entre les entrées et les sorties lorsque les services sont composés ensemble. Un exemple indicatif de câblage de service est SCA[4] (Service Component Architecture). Dans cette architecture, les services composants peuvent être implémentés dans n'importe quel langage de programmation disponible et encapsulé de sorte que tous les composants partagent une description similaire. Les services composants peuvent être câblés ensemble pour former ce que l'on appelle un composite. Les composites peuvent alors fonctionner eux-mêmes comme des services composants. Le modèle de composant peut être utilisé avec une orchestration avec cette dernière décrivant le contrôle et le flux de données entre les services participants et le premier implémentant les connexions d'entrée et de sortie réelles d'un service à un autre.

3. Exigences de la composition des services Web

Le processus de création d'un schéma de composition pour satisfaire une série d'objectifs fixés par un client est un problème complexe et multiforme, car il faut traiter plusieurs problèmes différents à la fois. Tout d'abord, il s'agit de rechercher dans un registre de services mondial en pleine croissance afin de trouver des services correspondants pouvant contribuer à la satisfaction complète des besoins du client. En supposant que ces services ont été trouvés, il faut les combiner avec succès, en résolvant les conflits et les incohérences entre eux, car ils seront très certainement créés par des personnes différentes utilisant des langages et des systèmes d'implémentation différents. Comme des incohérences peuvent survenir au moment de l'exécution, il peut être nécessaire de prévoir de tels événements afin de garantir que le système fonctionnera correctement. Enfin, même après avoir surmonté ces problèmes, nous devons être en mesure de nous adapter aux caractéristiques dynamiques des systèmes basés sur les services, avec des services qui fonctionnent, de nouveaux services en ligne et des services existants qui changent leurs caractéristiques.

[4] http://www.oasis-opencsa.org/sca

Si on tente de résoudre tous ces problèmes manuellement, on obtiendra certainement un schéma de composition qui ne soit pas imputable à la faute, alors que toute la procédure consommera beaucoup de temps et de ressources et ne peut pas être considérée comme évolutive. Par conséquent, il est évident qu'un certain degré d'automatisation doit être introduit dans la procédure de composition. Les approches qui impliquent l'automatisation dans la création du schéma de composition ainsi que pendant son exécution constituent une famille majeure connue sous le nom de composition automatisée de services. La composition automatisée de services a été une "solution miracle" dans la recherche sur les services Web et a attiré de nombreux chercheurs dans le monde entier. Dans le cadre de cette thèse, nous allons analyser et comparer les efforts les plus représentatifs dans le domaine de la composition de services Web automatisés et identifier les lacunes qui pourraient faire l'objet de recherches futures.

Dans cette section, nous allons présenter et analyser brièvement un ensemble d'exigences que nous avons retenu, dans le cadre de notre thèse, qui doivent être satisfaites, pour qu'un processus de composition soit considéré comme réussi.

3.1. Automatisation

Étant donné que notre étude se concentre sur la composition automatisée des services Web, la principale exigence est que la génération du schéma de composition doit être au moins partiellement (sinon entièrement) automatisée. L'objectif principal derrière la conception d'une approche pour gérer la composition de services Web est de minimiser l'intervention de l'utilisateur et d'accélérer le processus de production d'un service composite qui répond aux besoins des utilisateurs prédéfinis. L'automatisation réduit le temps passé à créer un schéma de composition par rapport au temps requis dans une approche de composition manuelle, élimine les erreurs humaines et réduit le coût global du processus. Ainsi, nous devons nous attendre à ce qu'une approche de composition de service Web réussie offre le niveau d'automatisation le plus élevé possible.

3.2. Dynamicité

Une des caractéristiques de définition d'une approche de composition de service est de savoir si elle produit un schéma de composition statique ou dynamique. Cette

distinction est étroitement liée à l'instant à laquelle le schéma de composition est créé, au moment du design ou au moment de l'exécution. Une approche de composition statique implique la sélection des services composants à utiliser pour les relier entre eux afin de former un processus de composition, et de déployer le processus composite. Ce processus ne peut être modifié d'aucune façon après le début de l'exécution, d'où le terme de composition au moment du design qui est souvent attribué aux approches de composition statique.

D'un autre côté, les approches de composition dynamique produisent un schéma de composition abstrait, qui est essentiellement un processus de composition sans les liaisons réelles de services. Le schéma de composition abstrait est concrétisé à l'exécution. La dynamicité garantit qu'un schéma de composition produit sera cohérent et exécutable longtemps après sa conception initiale. Un schéma de composition dynamique est capable de résoudre des problèmes tels que, la non disponibilité des services au moment de l'exécution, ce qui rendrait un schéma de composition statique invalide, incohérent et impossible à exécuter. Pour cette raison, les approches de composition dynamique sont préférables aux approches statiques.

3.3. Qualité de service

Une autre exigence fondamentale pour la composition automatisée des services Web est la qualité de service (QoS). Les approches qui prennent en considération la qualité de service tiennent compte non seulement des caractéristiques fonctionnelles des services, mais aussi des caractéristiques non fonctionnelles, tels que le temps de réponse, le prix, la disponibilité, etc. Il est important de prendre en compte les aspects QoS, lors de la décision d'inclure un service dans un schéma de composition, quand les exigences fonctionnelles sont satisfaites par plusieurs services. En conséquence, les services composites produits par des approches sensibles à la QoS offrent non seulement les capacités requises par l'utilisateur, mais garantissent également la meilleure qualité possible.

3.4. Evolutivité

L'apport incontestable des applications du monde réel est l'évolutivité. Le fait qu'une approche de composition fonctionne bien avec un ensemble de services n'est pas une

garantie qu'elle fonctionnera aussi bien avec un ensemble de services différent, plus vaste ou plus complexe. Les approches de composition doivent être testées par rapport à des registres de service de plus en plus importants afin d'examiner comment leurs performances sont évaluées. Une question récurrente dans ce contexte, celle de l'évolutivité par rapport à la performance. Il est important d'identifier les parties de l'approche qui peuvent présenter des limites et d'essayer de les résoudre afin d'assurer l'évolutivité maximale pour une performance donnée ou vice-versa.

3.5. Préférences des utilisateurs

La plupart des services Web existants ne sont pas conçus et mis en œuvre pour une consommation de service personnalisée. Dans de nombreuses situations, la sélection de service nécessite de prendre en considération les besoins et les préférences des consommateurs individuels. Par exemple, les gens peuvent être intéressés par différents types de nouvelles. Un service Web d'alerte de nouvelles devrait être capable d'adapter les nouvelles quotidiennes et choisir le contenu qui peut spécifiquement satisfaire les besoins individuels. Un autre exemple typique est celui des services de publicité ciblée, qui envoient automatiquement des publicités électroniques à des clients potentiels en fonction de leurs profils. Apparemment, la compréhension des intérêts individuels est cruciale pour s'assurer que chaque publicité est envoyée aux clients qui sont susceptibles de s'y intéresser.

3.6. Indépendance au domaine

Une approche de composition ne devrait pas être limitée à un domaine spécifique (à moins bien sûr, l'objectif de la recherche était exclusif à ce domaine dès le début). On devrait pouvoir appliquer la même approche à différents domaines, permettant de résoudre un large éventail de problèmes.

4. Différentes approches de composition automatique des services Web

Un service composite est toujours associé à une spécification, qui décrit entre autres la liste des services Web composants qui participent au service composite, la chronologie

d'exécution de ces services et les types de dépendances entre eux. La spécification doit également être affinée et adaptée aux caractéristiques de l'environnement dans lequel les services Web des composants doivent être exécutés, afin qu'un service composite puisse répondre aux besoins des utilisateurs, en fonction de ces caractéristiques et des préférences des utilisateurs. L'intégration des préférences des utilisateurs dans la spécification d'un service composite présente plusieurs avantages, de sorte que les services Web composants peuvent être personnalisés en fonction de ces préférences.

En raison de la multitude et de la diversité des approches de recherche existantes, il est impératif que nous effectuions un certain regroupement afin de faciliter la présentation des approches. Cinq groupes distincts d'approches seront examinés:

4.1. Approches orientés workflow

L'organisation et la gestion de workflow constituent une thématique de recherche majeure depuis trois décennies. En conséquence, il y a eu beaucoup d'efforts sur la façon de représenter une séquence d'actions. Tirant principalement parti du fait qu'un service composite est conceptuellement similaire à un workflow, il est possible d'exploiter les connaissances accumulées dans la communauté de workflow afin de faciliter la composition des services Web. Les frameworks de composition basés sur les techniques de workflow ont été l'une des premières solutions proposées pour la composition automatique des services Web. Initialement, la plupart des travaux ont porté sur des compositions statiques et manuelles. Cependant, des travaux plus récents ont tenté de réaliser une composition de service Web automatisée. En raison de la popularité de BPEL, la plupart des approches de cette catégorie utilisent BPEL d'une manière ou d'une autre.

Majithia et al. (Majithia et al., 2004) présentent un framework pour la composition automatisée de workflow en utilisant les technologies du Web sémantiques. Le framework consiste en un composeur de workflow abstrait qui accepte en entrée un objectif de haut niveau spécifié par l'utilisateur et le transforme en un workflow abstrait. Cette transformation est effectuée en interrogeant un registre de workflow existant pour vérifier si la même requête a déjà été traitée. Sinon, il fera marche arrière pour trouver un ensemble de services similaires offrant les mêmes fonctionnalités. Un composeur de

workflow concret prend le workflow abstrait généré en entrée et essaie de faire correspondre les tâches individuelles avec les instances disponibles des services réellement déployés en utilisant un algorithme de correspondance. Si le processus de correspondance réussit, l'algorithme de mise en correspondance génère un graphe exécutable qui sert d'entrée à un *Workflow Service Manager* qui combine les services disponibles en fonction des préférences spécifiées par l'utilisateur. Si le processus de correspondance échoue, le composant *Abstract WorkFlow Composer* est appelé récursivement pour trouver une combinaison de services pouvant fournir les fonctionnalités requises.

L'une des contributions les plus récentes basées sur les techniques de workflow est le travail de (Paik et al., 2014), qui présente une approche d'orchestration de workflow pour permettre une composition multiniveau imbriquée pour atteindre l'évolutivité. Le travail propose quatre étapes pour effectuer la composition automatique de services. Cette approche traite des situations où les objectifs de composition ne peuvent pas être atteints dans un processus en une seule étape. La première étape consiste à planifier un workflow de services atomiques à l'aide d'un planificateur de tâches hiérarchique. La deuxième étape fournit un mécanisme pour localiser les services à partir d'un registre de services. L'étape suivante implique le déploiement et l'exécution des services candidats les plus adaptés, sélectionnés en fonction de leurs propriétés non fonctionnelles. La phase finale implique l'exécution des services; dans le cas où il y a un échec, l'exécution s'arrête et la replanification est effectuée à nouveau. Les étapes ci-dessus sont utilisées pour traduire un service d'un workflow abstrait en un service concret. Cette approche tente de traiter des propriétés fonctionnelles et non-fonctionnelles pendant la composition de services.

4.2. Approches basées sur les modèles

Les approches dans cette catégorie exploitent les langages formels de modélisation tels que les automates à états finis (AEF), réseaux de Petri et Unied Modeling Language (UML) pour modéliser le problème de la composition des services. Ces langages formels fournissent un moyen pour spécifier à la fois la composition et ses exigences. Ils permettent également une manipulation aisée de la spécification en utilisant

différentes opérations fournies par chaque langage, respectif. Les langages formels sont très utiles pour le contrôle de cohérence.

Approches basées sur les automates à états finis: L'un des premiers travaux dans cette catégorie qui utilise les Automates à Etats Finis (AEF) est celui de [(Berardi et al., 2005a), (Berardi et al., 2005b), (Berardi et al., 2005c)], qui présente un framework formel dans lequel les arbres d'exécution sont utilisés pour décrire le comportement exporté des services (une abstraction pour ses exécutions possibles). Ces arbres d'exécution sont représentés à l'aide de machines à états finis. Dans l'approche, un service est modélisé en utilisant deux schémas, un schéma interne et externe qui sont représentés en utilisant les AEF. Le schéma externe spécifie le comportement exporté (visible de l'extérieur) des services, tandis que le schéma interne contient les informations sur les instances de service exécutant une action donnée au sein de la communauté de services. L'approche réduit le problème de la synthèse de composition dans la satisfaction d'une formule appropriée de Logique Dynamique Propositionnelle Déterministe (LDPD). C'est-à-dire que les modèles AEF des services disponibles et le service cible sont codés en LDPD et qu'un service cible existe si et seulement si l'ensemble des formules est satisfaisant et qu'un AEF est automatiquement synthétisé. Le AEF résultant est en outre traduit dans un processus BPEL et exécuté dans un moteur BPEL. Cependant, cette approche ne traite pas les échanges de données et de messages pendant le processus de composition. Une extension de ce travail utilise des systèmes de transition étiquetés pour représenter des services qui utilisent une topologie de communication prédéfinie (une structure de liaison exprimée en un ensemble de canaux) pour spécifier la manière dont les services échangent des messages (Berardi et al., 2005b). Le but principal de cette approche est de synthétiser un médiateur à partir d'un service objectif et d'un service disponible donné (les deux étant des systèmes de transition étiquetés) qui orchestrent l'exécution des services composants afin que leurs conversations soient conformes à la spécification du service objectif. Analogue à la première approche, le problème de composition est codé sous la forme d'une formule de Logique Dynamique Propositionnelle (LPD).

Pathak et al. (Pathak et al., 2008) proposent un framework pour la modélisation et l'exécution de la composition des services (MoSCoE) où les services disponibles et le

service objectif présentent ensemble un comportement d'état fini. L'approche utilise des Systèmes de Transitions Symboliques (STS) pour modéliser des services associés à des gardes sur des variables de domaine infini. Ils utilisent l'analyse de raffinement pour guider les utilisateurs à revoir leur objectif de composition en cas d'échec. Le framework comprend trois étapes: l'abstraction, la composition et le raffinement. Les services composants et le service objectif sont décrits à l'aide de machines d'état UML et sont traduits en STS. Les auteurs appliquent leurs algorithmes de composition pour synthétiser une composition si elle existe. Dans le cas contraire, l'utilisateur reformule ses besoins et essaie à nouveau. Même ainsi, cette approche ne tient pas compte de la manière dont une spécification peut être appliquée puisqu'elle n'explore pas toutes les exécutions possibles des STS.

Khoumsi (Khoumsi, 2013) pose le problème de composition de services en tant que problème de contrôle en utilisant une simple méthode d'automates d'entrées-sorties. Le problème est de synthétiser un orchestrateur *Orch* à partir d'un ensemble donné de services Web *S1*, ..., *Sn* et un objectif souhaité *S0* tel que *Orch* coordonne les services disponibles pour atteindre *S0*. Cependant, cette approche est limitée à la description des paramètres d'entrée/sortie des services et ne tient pas compte des contraintes de comportement des services.

Approches basées sur les réseaux de Petri : Dans le contexte de l'utilisation des réseaux de Petri comme langage de modélisation pour la composition automatisée des services, Hamadi et Benatallah (Hamadi & Benatallah, 2003) proposent une algèbre de Petri pour modéliser les flux de contrôle des services Web qui capturent également la sémantique d'une agrégation de services complexes. Le Framework proposé fournit un support pour la création et la vérification d'une composition en utilisant diverses constructions de programmation structurée (par exemple, séquence). Narayanan et McIlraith (Narayanan & McIlraith, 2002) présentent une approche où la sémantique d'un sous-ensemble pertinent de OWL-S (*Web Ontology Language for Web Services*) est codée en tant que logique de premier ordre connue sous le nom de calcul de situation. En utilisant cette sémantique, les auteurs transforment les descriptions de services en un formalisme de réseau de Petri et fournissent des procédures de décision pour la simulation, la vérification et la composition des services Web. Le résultat de ce

travail a été implémenté dans un outil qui prend en entrée une description OWL-S d'un service Web et synthétise automatiquement un réseau de Petri sur lequel les auteurs effectuent diverses analyses structurelles.

L'un des travaux les plus récents utilisant le formalisme du réseau de Petri est le travail décrit par (Xia et al., 2013). Le travail propose une approche pour l'analyse de fiabilité des processus OWL-S qui utilise un réseau de Petri stochastique non-markovien comme modèle de base. OWL-S est traduit en un réseau de Petri stochastique non Markovien. Une technique de calcul de fiabilité proposée qui prend en entrée les paramètres probabilistes des invocations de service est appliquée, et un SOAP est utilisé pour calculer la fiabilité du OWL-S.

Dans une autre direction, le travail de (Cheng et al., 2014) utilise des réseaux de Petri à prédicat flou pour traiter le problème de la composition automatique des services Web sémantiques. Un ensemble de règles de clauses de Horn est utilisé pour spécifier un service Web et sa sémantique floue, y compris le service objectif. Les auteurs produisent une transformation qui mappe une tâche de composition de services dans un problème de raisonnement de clauses de Horn. Le jeu de clauses Horn est ensuite transformé en un réseau de Petri à prédicat flou et une technique connue sous le nom d'analyse T-variant (Cheng et al., 2014) est utilisée pour vérifier l'existence d'une composition qui réalise un but donné. Les auteurs proposent un algorithme qui synthétise une composition basée sur une spécification donnée ainsi qu'un modèle de réseau de Petri à prédicat flou qui montre l'ordre d'appel des services sélectionnés.

Approches basées sur UML: Il existe un nombre considérable de travaux de recherche qui utilisent UML pour modéliser le problème de la composition automatisée des services. Pour citer quelques exemples, Wang et al. (Wang et al., 2014) introduisent une approche dans laquelle les structures de branchement conditionnelles sont utilisées pour modéliser le problème de la composition de services. Cette approche prend en charge les préférences de l'utilisateur ainsi que la capacité à s'adapter aux changements dans un environnement dynamique réel. Afin de modéliser avec précision les structures de branches conditionnelles, les auteurs utilisent des diagrammes d'activités UML pour représenter les dépendances dans les services composites. Ils considèrent deux types de préférences de l'utilisateur lors de la synthèse de la composition. Le premier est que

l'utilisateur choisi une classe de services plutôt qu'une autre en fonction d'une contrainte donnée. Le deuxième type est que l'utilisateur sélectionne des services qui offrent des fonctionnalités similaires basées sur certaines affectations de priorités**.**

De même, Skogan et al. (Skogan et al., 2004) proposent également une approche pour la composition des services Web sémantiques utilisant le développement piloté par les modèles (Model-Driven Development (MDD)). Les auteurs traduisent d'abord les descriptions WSDL en modèles UML. Cette étape de traduction permet de modéliser les services existants à l'aide de plates-formes UML conçues pour la construction des compositions. Skogan et al., appliquent des techniques MDD pour générer une composition basée sur les modèles UML des services Web, qui peuvent à leur tour être traduits en spécifications BPEL exécutables. De plus, Skogan et al. (Skogan et al., 2004) présentent une implémentation open-source qui réalise leur technique. Compte tenu de certaines préférences de l'utilisateur et d'un ensemble de services (chacun spécifié par un diagramme d'activité), les auteurs fournissent un algorithme qui génère tous les services composites réalisables.

4.3. Approches mathématiques

Cette catégorie est créée pour inclure toutes les travaux ayant une caractéristique commune d'être basée sur des fondements mathématiques tels que diverses logiques, calculs ou algèbres et théorie des graphes (d'où le terme mathématique). Bien sûr, cela n'implique nullement que les approches non incluses dans cette catégorie ne peuvent pas avoir un fondement mathématique, mais elles ont finalement été regroupées dans d'autres catégories en raison de similarités plus fortes que celles d'une base mathématique commune.

Raisonnement algébrique et logique : Les langages algébriques de processus tels que le calcul des systèmes de communication (CCS) (Milner, 1982) et le pi-calcul (Milner, 1999) ont été recommandés comme l'une des principales méthodes algébriques pour modéliser la composition des services Web (Ripon et al., 2014). Il n'est pas surprenant que les frameworks du Web sémantiques tels que WSMO (Bruijn et al., 2005) et SWSF (Battle et al., 2005) aient leur origine dans l'algèbre de processus. Par exemple, Ferrara (Ferrara, 2004) recommande un framework d'algèbre de processus en

tant que spécification abstraite pour modéliser un service Web. Leur framework utilise une analyse basée sur la bisimulation pour établir si un service peut servir de substitut à un autre service dans une composition. Le framework peut être également utilisé pour vérifier la redondance d'un ensemble de services dans une communauté. De même, comme l'algèbre de processus supporte l'analyse de simulation, elle permet l'utilisation des techniques de raffinement hiérarchiques (Kurshan, 1994) qui permettent le raffinement itératif d'une description de processus abstraite en une description moins abstraite. De plus, une vérification est ensuite effectuée pour vérifier diverses propriétés temporelles. Plus spécifiquement, les auteurs utilisent une spécification de langage temporel (LOTOS) (Eijk & Diaz, 1989) comme algèbre de processus et fournissent une correspondance bidirectionnelle entre WS-BPEL et LOTOS, et des directives générales pour les traductions entre WS-BPEL et LOTOS. Cette approche permet de vérifier les propriétés temporelles en utilisant des techniques de vérification de modèles.

Salaün et al. (Salaün et al., 2004) montrent comment l'algèbre de processus peut être utilisée pour modéliser des services Web à l'étape de conception où le système à développer est représenté comme une spécification abstraite utilisant l'algèbre de processus sur laquelle la validation peut être effectuée et utilisée comme modèle . En outre, les auteurs montrent qu'en appliquant l'ingénierie inverse, une spécification d'algèbre de processus peut être dérivée des descriptions d'interface de services Web existantes. Plus en détail, les auteurs développent une technique de conversion des processus BPEL en CCS dans laquelle diverses propriétés de service Web spécifiées dans la logique temporelle peuvent être vérifiées et raisonnées. Ils soutiennent que l'algèbre de processus peut être utilisée pour aborder divers problèmes de chorégraphie.

Rao et al. (Rao et al., 2004) présentent également une approche utilisant le théorème de la Logique Linéaire (LL) (Girard, 1987) prouvant la composition automatique des services Web sémantiques. Ils affirment que la sémantique de la logique linéaire leur donne le pouvoir expressif de décrire les attributs fonctionnels et non-fonctionnels des services Web. Initialement, les descriptions de service OWL-S sont automatiquement traduites en axiomes de la logique linéaire. Le service composite est généré automatiquement à partir des services disponibles à l'aide d'un démonstrateur de théorème de la logique linéaire. Le démonstrateur de théorèmes génère toutes les

compositions possibles qui sont ensuite traduites pour traiter des modèles en utilisant un langage de processus pi-calcul. Ce modèle peut être converti en modèles de workflow BPEL en phase d'exécution. En général, l'utilisation du sous-ensemble propositionnel de la logique linéaire peut ne pas être suffisante pour représenter les exigences de la composition. Par conséquent, les auteurs proposent d'utiliser le principe de la déduction partielle (Lloyd et Shepherdson, 1991) pour permettre plus de flexibilité dans la spécification du but des utilisateurs.

Récemment, Yang et al. (Yang et al., 2014) adoptent la logique étendue Croyance-Désir-Intention (Belief-Desire-Intention (BDI)) pour traiter le problème de la composition des services Web dans le cas où l'objectif d'un utilisateur n'est pas cohérent avec l'objectif de la composition. Le modèle Croyance-Désir-Intention est utilisé pour spécifier la croyance, le désir et l'intention d'un service, qui sont mappés dans l'environnement de BDI, l'objectif du service Web (et de l'utilisateur) et des schémas de composition respectivement. Un modèle de processus est ensuite utilisé pour modéliser les résultats. Afin de permettre une évolution dynamique de leur travail, ils utilisent AgentSpeak (L) (un langage de communication) pour l'exprimer.

Les graphes : La théorie des graphes et les algorithmes utilisant des graphes comme moyen de représentation ont été largement utilisés pour modéliser le problème de la composition des services. Pour modéliser les services Web et leurs compositions, des graphes sont utilisés pour représenter et modéliser la structure, le flux de données et les dépendances, ainsi que pour spécifier les exigences de composition d'une tâche de composition donnée.

Hashemian et Mavaddat (Hashemian & Mavaddat, 2005) présente une approche pour la composition automatique de services Web en combinant des techniques basées sur des automates d'interface et la théorie des graphes. En particulier, les auteurs utilisent un modèle basé sur l'état, similaire aux diagrammes d'états finis, pour représenter le comportement des composants logiciels pour modéliser les services Web. L'automate d'interface est utilisé pour modéliser les entrées et les sorties d'un composant avec l'ordre temporel des actions qu'il effectue. Un graphe de dépendances est utilisé pour modéliser les dépendances de données entre les services composants, de sorte que les nœuds du graphe correspondent aux entrées et sorties des services Web et que les

arêtes représentent les services Web associés. L'algorithme de synthèse d'une composition est constitué de deux étapes. La première étape consiste à trouver un ensemble de services Web pouvant potentiellement participer à la composition et la seconde étape consiste à générer un graphe de dépendances basé sur une requête utilisateur donnée en utilisant l'ensemble des services Web trouvé à l'étape précédente.

Une approche connexe présentée par (Li et al., 2016) propose un algorithme à base de graphes pour modéliser le problème de la composition automatisée des services Web. L'algorithme développé par ces auteurs génère d'abord toutes les combinaisons de services possibles, puis les stocke dans un système de base de données relationnelle. Maintenant, lorsqu'un client effectue une requête, un algorithme est utilisé pour composer un ensemble de requêtes SQL et rechercher dans la base de données. L'algorithme retourne la meilleure solution possible en tenant compte des propriétés de qualité de service.

Ba (Ba, 2016) décrit une approche de composition de service automatique qui utilise une technique de réécriture de requête. L'approche utilise une spécification abstraite d'une composition et une définition d'un ensemble de services concrets. Compte tenu de ces spécifications de plus haut niveau, l'auteur de cet article utilise une version améliorée d'un algorithme de réécriture de requête connu sous le nom MiniCon qui calcule une description de ce dernier. Ensuite, un graphe bipartite est utilisé pour réduire le problème de réécriture à un problème de couverture exacte. Ensuite une composition est déduite à partir de ce graphe.

4.4. Approches orientés intelligence artificielle

La planification automatisée est une branche de l'intelligence artificielle qui se concentre sur la génération automatique de plans ou de stratégies qui réalisent un but donné dans un domaine donné. Étant donné une description des états initiaux possibles du monde et une spécification d'un ensemble d'actions possibles, le problème de planification implique l'élaboration d'un plan ou d'une stratégie pour générer une séquence d'actions atteignant l'un des objectifs souhaités.

L'un des langages de description les plus couramment utilisés pour modéliser les problèmes classiques de planification est le langage PDDL (Planning Domain

Definition Language) (Estivill-Castro & Ferrer-Mestres, 2013) qui a contribué au développement de langages de description de services Web tels que OWL-S. (Akkiraju et al., 2004) posent le problème de composition de service basé sur le workflow comme un problème de domaine de planification. Les auteurs développent un prototype de moteur de workflow qui prend en entrée un flux abstrait BPEL4WS augmenté d'annotations sémantiques dans OWL-S et effectue la découverte, la composition, la liaison et l'exécution de services Web. Ce moteur traduit d'abord les descriptions de service OWL-S en PDDL; Le résultat de cette étape est ensuite transmis dans le framework de planification de IBM Planner4J pour une analyse plus approfondie. Le module de planification exige un service concret pour chaque tâche dans le flux BPEL abstrait au moment de l'exécution. Si ce processus échoue, le module de planification utilise les services disponibles pour résoudre le problème de planification. Cependant, puisque la création du flux abstrait est manuelle, cette approche est considérée comme une approche semi-automatique.

De même, Zeng et al. (Zeng et al.,2008) formulent également le problème de composition de service sous la forme d'un problème de planification orienté vers un but qui utilise un mécanisme d'inférence de règles complet pour générer dynamiquement un schéma de composition. L'approche prend en compte trois entrées: (i) un ensemble de règles de composition de services spécifiques au domaine (règles métier) associées à des conditions préalables et des obligations; (ii) une description des objectifs de l'utilisateur; et (iii) une description des hypothèses commerciales (règles et structures organisationnelles). Ces concepts sont ensuite modélisés sur la base d'une ontologie de service cohérente que les auteurs ont conçue. Pour se faire, les auteurs proposent un mécanisme d'inférence de règles en trois phases qui génère un schéma de composition de services de façon incrémentielle. La première phase connue sous le nom de chainage arrière consiste à déterminer toutes les actions possibles (de l'objectif métier à l'état initial) qui doivent être exécutées pour satisfaire l'objectif métier. La deuxième étape consiste à utiliser une technique appelée inférence à chaînage direct qui tente de prendre le résultat de l'exécution de tâches dans la phase précédente, puis de déterminer les tâches qui doivent éventuellement être incluses. L'étape finale fournit l'inférence de flux de données.

Le travail de (Rao et al., 2006) présente un framework mixte pour la découverte et la composition de services Web sémantiques qui n'essaie pas d'automatiser toutes les décisions, mais suppose que les utilisateurs conservent un contrôle étroit sur de nombreuses décisions tout en leur déléguant des tâches fastidieuses. Ils ont utilisé un algorithme de planification appelé *GraphPlan* pour construire leur moteur de composition qui combine un raisonnement basé sur des règles sur les ontologies OWL avec Jess (un moteur de règles et un environnement de script pour les plates-formes Java) et des fonctionnalités de planification. L'utilisation principale de la planification ici est de fournir des suggestions de schémas de composition à l'utilisateur, au lieu de faire respecter les décisions qui forment le but ultime de ce travail.

De la même manière, Wu et al. (Wu et al., 2007) ont utilisé la planification basée sur les graphes pour résoudre le problème de composition de services. L'approche tient compte à la fois de l'hétérogénéité des processus et des problèmes d'hétérogénéité des données. Ils ont implémenté leur propre définition d'un service Web sémantique abstrait construit sur SAWSDL et WSDL-S. Ensuite, les auteurs ont étendu GraphPlan qui génère automatiquement le contrôle d'un processus Web. Le système génère automatiquement un processus BPEL exécutable à partir d'une spécification donnée de l'état initial, de l'état de l'objectif et d'une description de service Web sémantiquement annotée dans SAWSDL. La médiation de données est effectuée en utilisant des activités d'affectation dans BPEL ou par un médiateur de données qui peut être intégré dans un middelware. Au moment de l'exécution, le médiateur de données convertit le service disponible dans le format du message d'entrée de l'opération qui est invoquée lorsqu'elle est appelée par le processus BPEL.

Sirin et al. (Sirin et al., 2003) tentent de tirer parti de la technique de planification HTN (Hierarchical Task Network) pour la composition automatisée des services Web sémantiques. Les auteurs sont motivés d'utiliser cette technique basée sur le fait que le concept de décomposition de tâches dans la planification HTN est très similaire au concept de décomposition de processus composite dans l'ontologie de processus OWL-S. Ils construisent un système qui traduit les descriptions de services OWL-S en SHOP2 (un système de planification HTN indépendant du domaine) (Nau et al., 2003) et ensuite les auteurs fournissent une méthode pour synthétiser automatiquement un plan de

composition réalisable. Le système est également capable d'exécuter des informations fournissant des services Web pendant le processus de planification. Ils prouvent l'exactitude de leur approche en montrant la correspondance avec la sémantique du calcul situationnel de OWL-S.

Peer (Peer, 2005) montre comment le planificateur d'ordres partiels connu sous le nom de planificateur heuristique partiel polyvalent (VHPOP) peut être combiné avec un algorithme de replanification pour la composition automatique de services. L'auteur fournit une définition des services Web sémantiques qui est ensuite traduite en PDDL en tant qu'entrée pour VHPOP. La description PDDL du service Web est introduite dans VHPOP ainsi qu'un ensemble de liens entre les tâches à éviter. Un ou plusieurs plans sont générés automatiquement et peuvent être partiellement définis. L'exécution est effectuée une étape à la fois, car le ou les plans générés ne garantissent pas nécessairement une exécution correcte. Par conséquent, si un plan échoue, la replanification est effectuée et un nouveau plan est produit, compte tenu des conditions de l'échec; cependant, si l'exécution d'un plan réussit, il n'est pas nécessaire de replanifier et l'on peut passer à la tâche suivante.

Récemment, Sohrabi et McIlraith (Sohrabi & McIlraith, 2009) ont présenté une approche qui prend en charge la personnalisation et l'optimisation au cours de la construction de la composition en intégrant les préférences dans la planification HTN. Ce travail s'appuie sur la technique présentée par McIlraith et Son (McIlraith & Son, 2002) en étendant et en personnalisant Golog (Ferrein et al., 2009) pour supporter les contraintes personnalisées et non déterministes dans les exécutions séquentielles puis les auteurs ont reconçu ConGolog, l'interprète de Golog pour prendre en charge ces changements. Fait intéressant, ce développement a eu lieu parallèlement au développement de la définition d'OWL-S et a été l'un des premiers travaux à avoir utilisé les services Web sémantiques comme paramètres d’entrée des planificateurs grâce à la traduction vers PDDL.

4.5. Approches basées sur la recommandation

La sélection et la recommandation des services ont été largement étudiées pour faciliter la composition des services Web au cours des dernières années. (Shao et al., 2007) ont

proposé une approche fondée sur le filtrage collaboratif pour calculer l'extraction et la prédiction de la similarité à partir des expériences des consommateurs sur les services Web utilisés. Tang et al. (Tang et al., 2012) ont incorporé les informations de localisation des utilisateurs et des services dans la prévision de la qualité de service, et ont proposé une méthode de filtrage collaboratif sensible à l'emplacement pour la recommandation de service. Lo et al. (Lo et al., 2012) ont proposé un framework de prévision de QoS, appelé WSPred, pour fournir une prédiction de valeur de QoS personnalisée et sensible au temps pour différents utilisateurs de services.

Kang et al. (Kang et al., 2012) ont proposé une approche de recommandation de services Web basée sur l'historique d'utilisation du service qui incorpore à la fois les préférences de l'utilisateur et les paramètres QoS dans la recommandation des services Web.

Yao et al. (Yao et al., 2013) ont proposé une approche de recommandation de service hybride en combinant le filtrage collaboratif avec les fonctionnalités basées sur le contenu des services Web. Dans leur travail, les préférences des utilisateurs sont représentées par un ensemble de variables latentes, qui est développé hors ligne. Cependant, les préférences QoS des utilisateurs ne sont pas prises en compte.

Zhao et al. (Zhao et al., 2010) ont fourni une plate-forme complète et extensible pour la consommation de services et la navigation nommée HyperService. Avec HyperService, les utilisateurs peuvent facilement rechercher et naviguer parmi les services lorsqu'ils naviguent sur un site Web contenant des hyperliens. Selon les mots-clés et le contexte de navigation, un ensemble de services pertinents est recommandé. Chaque fois que les utilisateurs sélectionnent un service parmi l'ensemble suggéré, un autre ensemble de services est recommandé en fonction du service sélectionné. Ainsi, les services Web composés de manière interactive sont présentés aux utilisateurs finaux via une interface utilisateur interactive de style Web 2.0. Le moteur sémantique est le noyau de la fonction HyperService. Il fournit les fonctions de découverte automatique des relations, d'analyse du comportement de l'utilisateur (nombre d'utilisations, score d'évaluation, etc.), d'accessibilité des services par recherche par mot-clé et de recommandations contextuelles effectuées grâce aux méthodes basées sur le contenu. Les services adaptés au contexte de navigation actuel de l'utilisateur, ayant les meilleurs

facteurs d'importance globale (plus un service est lié à d'autres services, plus il est populaire) sont recommandés et présentés aux autres utilisateurs d'une manière simple et attrayante.

Picozzi et al. (Picozzi et al., 2010) ont proposé un modèle visant à aider les utilisateurs finaux à reconnaître les composants les plus appropriés. L'approche de recommandation proposée basée sur la qualité calcule la qualité des mashups pour produire une recommandation de mashup de haute qualité. Cette valeur est en fait une mesure de qualité agrégée calculée sur la base de la qualité de chaque composant du mashup. Les utilisateurs finaux qui ont déjà créé un mashup final ou intermédiaire peuvent obtenir des recommandations de mashups sur les extensions possibles d'un mashup donné. Ils peuvent étendre un mashup particulier basé sur une certaine recommandation et continuer à étendre le mashup obtenu en considérant d'autres recommandations. Les mashups recommandés sont classés sur la base d'un algorithme de recommandation basé sur la qualité.

Cappiello et al. (Cappiello et al., 2012) ont illustré l'incorporation de recommandations basées sur la qualité dans le processus de développement de mashup pour permettre aux utilisateurs finaux de compléter et / ou d'améliorer leurs mashups. Ainsi, un travail de composition assistée dans lequel la qualité et le rôle des services candidats sont le moteur des recommandations de mashup a été souligné dans ce travail. La qualité de la composition est calculée comme une agrégation pondérée de la qualité des composants individuels. Les poids reflètent les rôles, c'est-à-dire l'importance de chaque service candidat dans la composition. Une fois que l'utilisateur a sélectionné le premier composant candidat, l'algorithme de classement basé sur la qualité est exécuté. Il procède selon deux étapes: i) la catégorisation du composant à inclure dans le mashup courant en utilisant des mécanismes de filtrage collaboratif; ii) la sélection d'un composant particulier, appartenant à la catégorie définie dans i). Une autre fonctionnalité intéressante de cet algorithme consiste à recommander des compositions similaires mais de meilleure qualité lorsqu'elles sont appliquées sur des mashups finaux prêts à l'emploi.

Une nouvelle méthode de recommandation de services tenant compte de la catégorie de ces derniers est proposée dans (Xia et al. , 2015). Il s'agit en fait d'une

approche en trois étapes pour surmonter les restrictions susmentionnées et propose une recommandation de services sensible à la catégorie pour la création de mashups. En fait, après avoir reçu un texte d'exigence d'un utilisateur, le moteur de recommandation sensible à la catégorie l'analyse pour déduire les catégories de services qui vont être impliquées dans la tâche de composition du mashup. Ensuite, le moteur recherche les services candidats appartenant aux catégories déduites et les classe dans ces catégories. Enfin, le moteur de recommandation renvoie "la listes de classement des services candidats par catégorie". L'utilisateur sélectionne dans chaque catégorie un service et ainsi la composition est exécutée. Une fois qu'une exigence de mashup est reçue, la prédiction de pertinence de la catégorie de service commence. Combinant l'apprentissage automatique et le filtrage collaboratif, l'approche décompose les exigences de mashup et prédit explicitement les catégories de services pertinentes. Enfin, sur la base d'un framework d'apprentissage automatique distribué, les services sont recommandés sous la forme de "catégories par listes de classement des services candidats". Par conséquent, la question de classement des services sans signification est surmontée.

Maaradji et al. (Maaradji et al., 2010) ont introduit le framework SoCo qui s'appuie sur les connaissances récupérées des réseaux sociaux pour modéliser les interactions des utilisateurs afin d'aider les utilisateurs finaux à sélectionner les services Web lorsqu'ils entreprennent interactivement des tâches de composition de services Web. Grâce à une interface graphique offrant une fonctionnalité glisser-déposer pour les mashups, les utilisateurs finaux reçoivent une liste triée des services candidats pertinents pour être le successeur de l'actuel. SoCo fournit des recommandations dynamiques grâce à un système de recommandation social intégré. Ses recommandations sont basées sur le profil de l'utilisateur (contenant ses intérêts, ses préférences et l'historique de ses interactions avec le système) et la proximité sociale extraite des réseaux sociaux ainsi que sur les compositions précédemment construites. Cette information est utilisée pour estimer une recommandation de confiance. Les services Web recommandés sont les plus fiables, c'est-à-dire qu'ils ont des valeurs de confiance recommandées élevées.

Xu et al. (Xu et al., 2013) ont utilisé des relations sociales multidimensionnelles parmi les utilisateurs, les sujets, les mashups et les services pour recommander les

services nécessaires à la création de mashups. En utilisant un algorithme de factorisation couplé, les services requis pour la construction de mashup actuelle sont prédits, classés et livrés à l'utilisateur final en une fois. Les services recommandés ne sont pas fonctionnellement similaires. Ils sont plutôt l'ensemble des services nécessaires et ils sont fournis aux utilisateurs finaux, à la fois en fonction des spécifications fonctionnelles des utilisateurs et des exigences implicites qui sont déduites par le modèle de sujet. Les utilisateurs doivent simplement sélectionner les services appropriés et les composer en effectuant des actions "glisser-déposer".

5. Discussion

Dans cette section, nous proposons un tableau comparatif des différents travaux issus des différentes approches de composition des services et la façon dont chacun de ces travaux traite les exigences que nous nous sommes fixées, dans le cadre de cette thèse, qui doivent être satisfaites, pour qu'un processus de composition soit considéré comme réussi (Automatisation, dynamicité, qualité de service, évolutivité, préférences des utilisateurs et l'indépendance au domaine) (voir table 3.1).

Travaux	**Automatisation**	**Dynamicité**	**Qualité de service**	**Evolutivité**	**Préférences des utilisateurs**	**Indépendance au domaine**
Approches orientés Workflow						
(Paik et al., 2014)	✓	✓	✓	✓		Non vérifiée
(Majithia et al., 2004)	✓	✓			✓	Non vérifiée
Approches basées sur les modèles						
[(Berardi et al., 2005a), (Berardi et al., 2005b), (Berardi et al., 2005c)]	✓	✓				Non vérifiée
(Pathak et al., 2008)	✓	✓			✓	Non vérifiée
(Khoumsi, 2013)	✓	✓				Non vérifiée
(Xia et al., 2013)	✓	✓				Non vérifiée
(Cheng et al., 2014)	✓	✓				Non vérifiée
(Wang et al., 2014)	✓	✓			✓	Non vérifiée
Approches mathématiques						
(Yang et al., 2014)	✓	✓		✓	✓	Non vérifiée
(Li et al., 2016)	✓	✓	✓		✓	Non vérifiée
(Ba, 2016)	✓	✓				Non vérifiée

Approches orientées Intelligence Artificielle (IA)						
(Zeng et al.,2008)	✓	✓				Non vérifiée
(Rao et al., 2006)		✓				Non vérifiée
(Sohrabi & McIlraith, 2009)	✓	✓			✓	Non vérifiée
Approches basées sur la recommandation						
(Kang et al., 2012)	✓	✓	✓		✓	Non vérifiée
(Yao et al., 2013)	✓	✓			✓	Non vérifiée
(Xia et al. , 2015)	✓	✓				Non vérifiée
(Xu et al., 2013)	✓					Non vérifiée

Table 3.1 : Comparaison des différents travaux issus de différentes approches de composition des services Web.

En général, nous pouvons conclure que, bien que les approches de composition basées sur les workflow aient évolué, passant des méthodes de composition manuelles et statiques à l'automatisation et à la dynamicité, les tâches résultantes se limitent à des schémas simples tels que l'exécution séquentielle et parallèle. Dans certains cas l'automatisation n'est prise en charge que pendant l'exécution et l'adaptation du processus métier, tandis que le processus de conception de worklfow est manuel. Ce problème a été résolu en combinant des méthodes basées sur le workflow avec des techniques de planification d'IA.

Le problème de la composition automatisée des services a été abordé en utilisant les techniques de planification de l'intelligence artificielle plus que toute autre technique, mais tous les problèmes ne peuvent pas être modélisés par la planification. Par exemple, la plupart des techniques de planification supposent que le comportement des services est déterministe et, par conséquent, ces approches échouent lorsque des événements inattendus se produisent. De plus, ils ont tendance à négliger le contexte dans lequel se déroule la synthèse de la composition en anticipant à l'avance l'interaction des services. De plus, si les plans générés par les algorithmes de l'intelligence artificielle ne sont pas capables d'atteindre un objectif cible donné, la seule façon de résoudre ce problème est de replanifier ce qui peut générer un résultat trop tard. Par conséquent, la planification peut ne pas être la meilleure option pour traiter le problème de la composition de services dans ce cas. De plus, les techniques classiques de planification ne peuvent générer que des séquences d'actions linéaires et le résultat de chaque action doit être connu à l'avance.

La principale différence entre les deux approches est que les techniques basées sur la planification d'IA ne font aucune hypothèse concernant une vue abstraite du service composite. En revanche, cette vue abstraite est un élément clé des approches basées sur le workflow.

Les approches mathématiques et celles basées sur les modèles bien qu'elles soient rigoureuses ne sont pas les approches dominantes, bien que ces dernières années elles gagnent de plus en plus la confiance de la communauté des chercheurs dans le domaine de la composition des services Web.

La composition des services Web basée sur la recommandation est la technique dominante à l'heure actuelle. Largement inspirée des techniques de recommandation des produits en se basant surtout sur les réseaux sociaux qui sont en vogue ces dernières années. Nous pouvons vérifier que la recommandation des services Web peut d'une manière ou d'une autre satisfaire les critères que nous nous sommes fixés au préalable pour vérifier qu'une composition de services Web est réussie. La recommandation basée sur les réseaux sociaux de services Web est l'approche que nous avons retenue pour développer une approche innovante pour la composition des services Web personnalisés.

6. Conclusion

La composition dynamique des services Web émerge comme une nouvelle façon d'autonomiser le commerce électronique. Le développement de services Web composites permet d'économiser du temps et de l'argent pour développer de nouvelles applications et améliorer l'interopérabilité et la collaboration entre les partenaires du commerce électronique. La composition homogène des services Web offre un énorme potentiel de rationalisation des processus métier et d'intégration des applications d'entreprises. Les applications peuvent déterminer non seulement quels services existants peuvent être liés ensemble, mais aussi comment ils doivent être exécutés lors de l'exécution afin de répondre aux demandes complexes des utilisateurs. Cependant, la composition des services Web est une tâche compliquée et difficile en soit, en raison du nombre potentiellement important de services qui fournissent les mêmes fonctionnalités, ainsi que l'hétérogénéité et la nature changeante des services.

Dans ce chapitre nous avons présenté pratiquement les techniques les plus importantes pour la composition des services Web introduites dans la littérature. Nous nous sommes fixés des exigences, dans le cadre de cette thèse, qui doivent être satisfaites, pour qu'un processus de composition soit considéré comme réussi. Une étude comparative des différents travaux issus des différentes approches, nous a permis de consolider l'hypothèse que la composition des services Web basée sur la recommandation est la technique la mieux adaptée pour remplir les exigences que nous sommes fixées (Automatisation, dynamicité, qualité de service, évolutivité, préférences des utilisateurs et l'indépendance au domaine). Le chapitre suivant sera consacré à détailler notre contribution.

Chapitre 4

Intégration flexible des services Web: une nouvelle approche sociale personnalisée

Sommaire

1. Introduction 94
2. Architecture proposée pour la composition des services Web sociaux personnalisés 95
3. Communautés de services Web 97
4. Construction du réseau social de services Web 100
   - 4.1 Parametres qualité de services (QoS) considérés 100
   - 4.2 Fonction objective 101
   - 4.3 Associations de collaboration pour la composition 103
   - 4.4 Associations de collaboration pour la substitution 104
   - 4.5 Associations de recommandation pour la composition 105
5. Sélection des services Web atomiques pour la composition 105
6. Composition des services Web sociaux personnalisés 108
   - 6.1 Construction des préférences utilisateur 108
     - 6.1.1 Attributs statiques 108
     - 6.1.1 Attributs acquis 108

6.2 Algorithme pour la composition de services Web sociaux personnnalisés 109
7. Tolérance aux pannes dans le cadre de l'approche proposée 110
7.1 Associations de recommandation pour la substitution 110
7.2 Algorithme de sélection des services Web pour la substitution 111
8. Implémentation et évaluation 113
8.1 Approche de composition des services Web personnalisés 113
8.2 Approche de substitution des services Web 118
9. Conclusion 119

1. Introduction

L'objectif de ce chapitre est de (1) présenter l'approche proposée pour la composition des services Web sociaux personnalisés ainsi qu'une évaluation préliminaire (2) son applicabilité à l'aide d'exemples (c'est-à-dire, comment modéliser un problème de composition de service en utilisant notre approche), et (3) les complexités de calcul en termes de temps et d'espace nécessaires pour générer un service Web composite avec des graphes associés à chaque cas, (4) présenter l'approche proposée pour la substitution des services Web en cas d'échec ou d'indisponibilité de ces derniers, ainsi qu'une évaluation préliminaire. Dans la communauté de recherche sur les approches de compositions automatisées des services Web, il n'existe pas d'algorithmes de référence ou d'ensembles de tests publics connus que nous pouvons utiliser pour faire une comparaison avec notre approche. Ainsi, nous choisissons des exemples couramment utilisés par la communauté de recherche sur la composition des services pour valider nos résultats de recherche.

De nos jours, la concurrence ne se limite pas aux produits, les services Web ne sont pas non plus exclus de cette compétition. Les fournisseurs développent plusieurs services Web qui pourraient offrir des fonctionnalités similaires comme la réservation d'hôtels et la location de voitures. Pour faciliter et améliorer le processus de découverte de services Web dans un environnement ouvert comme Internet, il est suggéré de regrouper les services Web similaires en groupes appelés communautés [(Benatallah et al., 2003), (Medjahed & Bouguettaya, 2005)]. Bien que les services Web soient

intensivement étudiés, les problèmes suivants liés aux communautés de services Web n'ont pas encore été résolus: comment initier, et spécifier une communauté de services Web, la fonctionnalité d'un service Web est le seul facteur qui conduit à l'établissement d'une communauté, et comment spécifier et gérer les services Web qui résident dans une communauté?

L'approche proposée dans le cadre de cette thèse tire parti de la puissance des réseaux sociaux et des principes des systèmes de recommandation pour définir les interactions entre les services Web dans une même communauté. La construction de tels réseaux a besoin d'être détaillée. Les graphes associés au réseau social de services Web également. Le processus de composition des services Web proposé est basé sur ces interactions et sur la définition d'un modèle de qualité de service (QoS).

2. Architecture proposée pour la composition des services Web sociaux personnalisés

La tâche de composition des services Web est définie comme le processus de sélection, de combinaison et d'exécution des services Web pour atteindre un objectif donné (Martin et al., 2005). L'approche est développée dans cette optique. Le processus de composition des services Web est basé sur l'architecture proposée, illustrée par la figure 4.1.

Un client envoie une requête via le module interface. Sa requête est ensuite analysée et la communauté associée à celle-ci est extraite. Nous rappelons que nous sommes dans le contexte du Web 2.0. La structure arborescente (présentée au paragraphe 3) est utilisée pour représenter les communautés et faciliter la tâche de création et d'identification des différentes communautés. Nous supposons d'abord que pour chaque requête correspond une et une seule communauté. Sachant que pour chaque communauté est associé un seul arbre. Une interface conviviale aide les fournisseurs de services Web à les insérer dans les communautés existantes ou à créer de nouvelles communautés dans le réseau social de services Web. Les services Web sont censés être publiés dans des registres UDDI.

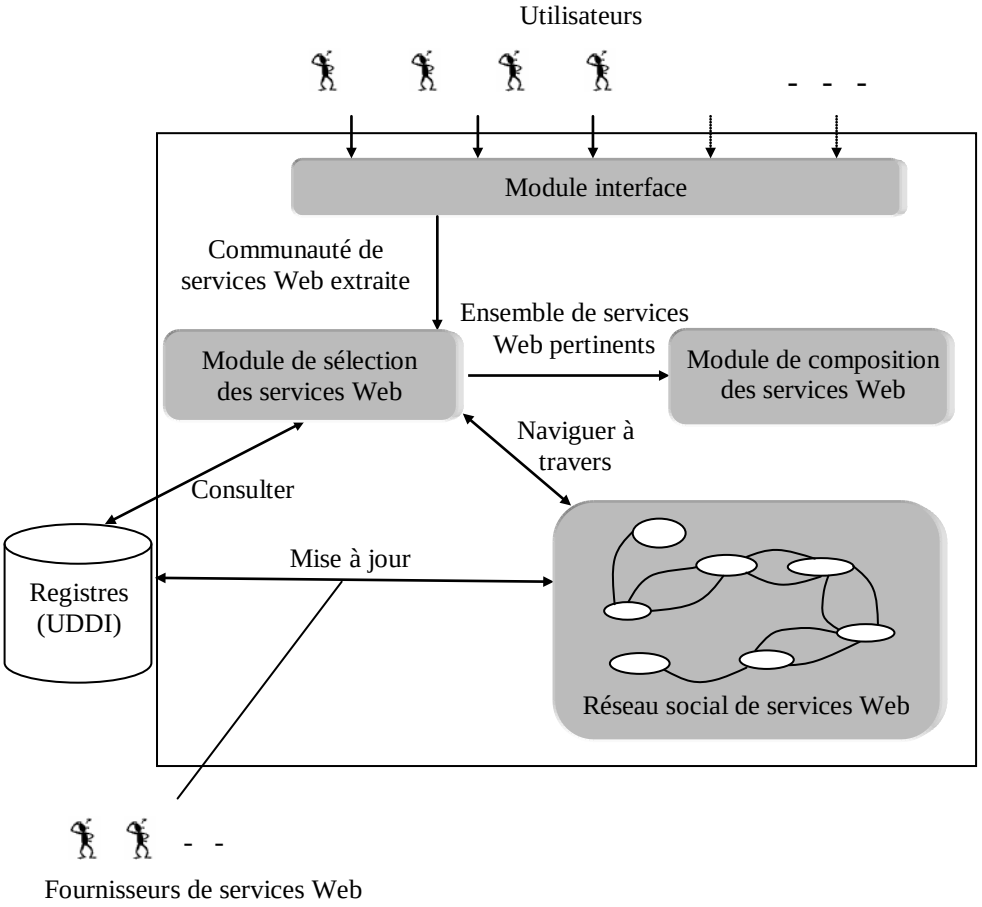


Figure 4.1 - Architecture pour la composition des services Web sociaux personnalisés.

Module interface : Le traitement de la requête du client sort du cadre de cette thèse. Plusieurs techniques innovantes pourraient être employées, entre autre celle découlant du domaine de traitement automatique de la langue. Dans notre contexte, nous supposons que les requêtes des clients sont analysées et les communautés associées à ces requêtes sont extraites et transmises au module de sélection des services Web.

Module de sélection des services Web : ce module permet la sélection proprement dite des services Web atomiques. Il reçoit en entrée la communauté de services Web retenue par le module interface. Nous rappelons que nous supposons qu'à chaque requête est associée une et une seule communauté. Une communauté de services Web peut contenir plusieurs sous-communautés. Dans la figure 4.2, la communauté "Procédure de construction" englobe les sous communautés *WS-Taxe_Paiement, WS-*

Permis_de_Contruire. Ainsi si l'utilisateur souhaite obtenir un permis de construire, Le paiement des taxes lui est proposé également sont qu'il ne soit explicitement demandé par ce dernier, et l'utilisateur a le choix d'accepter ou rejeter la proposition.

Module de composition des services Web : Le module de composition des services Web reçoit en entrée les services Web sélectionnés pendant la phase précédente de sélection des services Web atomiques, retenus par le module de sélection de ces derniers. Si les services Web correspondent aux préférences du client stockées dans son profil et plus précisément les attributs acquis de ce dernier, le processus de composition est enclenché, sinon il est indiqué au client que les services Web sélectionnés sont mieux cotés que ceux stockés dans son profil. Ainsi le client a le libre choix, soit de les retenir ou d'exécuter comme-même ceux stockés dans son profil.

Réseau social de services Web : regroupe l'ensemble des services Web qui sont répertoriés en communautés. La structure du réseau social de services Web est détaillée le long de ce chapitre. Le réseau social de services Web est en interaction permanent avec le module sélection des services Web. Les services Web sélectionnés sont ceux figurant dans le réseau social de ces derniers.

3. Communautés de services Web

Dans cette section, nous décrivons la notion de communauté de services Web comme une relation entre services Web caractérisées par des poids (construction de maisons, réservation d'hôtels, par exemple). Un service Web peut appartenir à différentes communautés. Une communauté réuni un ensemble de services Web répondant aux mêmes exigences fonctionnelles [(Metrouh & Mokhati, 2013), (Metrouh & Mokhati, 2016), (Metrouh & Mokhati, 2018)]. Ainsi, une communauté fournie un niveau d'abstraction entre les applications clientes et les services qu'elles souhaitent utiliser. L'objectif principal proposé dans le cadre de cette thèse est de fournir un cadre pour la composition de services Web avec un ensemble de services Web sélectionnés répondant aux exigences fonctionnelles et aux meilleures exigences non-fonctionnelles.

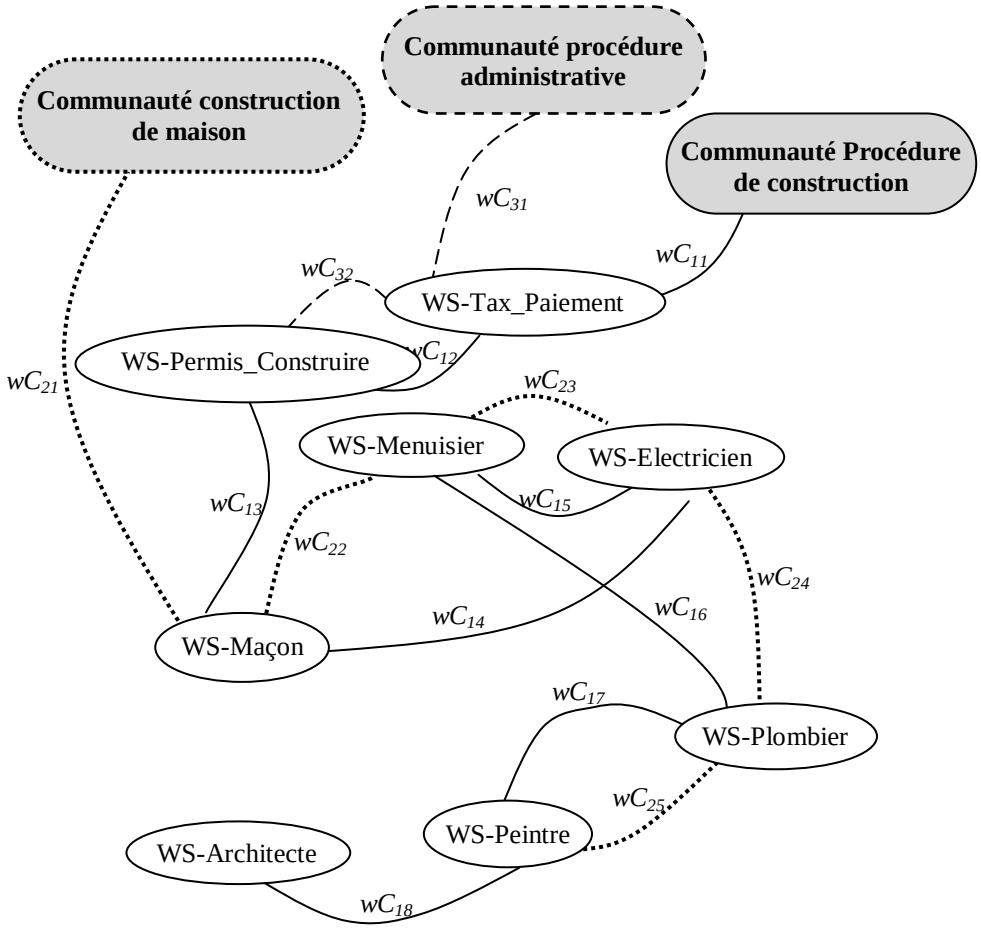
Une communauté de services Web a un contenu dynamique: de nouveaux services Web se rejoignent, d'autres partent, certains deviennent temporairement indisponibles, etc. Tous ces événements doivent être surveillés de près, sinon des conflits surviennent.

Par exemple, si un service Web n'est plus membre d'une communauté, ses pairs peuvent supposer que le service Web est toujours dans la communauté. De plus, les services Web n'exposent pas toujours une attitude coopérative lorsqu'ils rejoignent une communauté. Premièrement, ils peuvent rivaliser sur les ressources informatiques, ce qui peut affecter leurs performances. Deuxièmement, ils peuvent annoncer des détails non fonctionnels pour améliorer leurs opportunités de participation dans les services Web composites.

Un service Web peut appartenir à une ou plusieurs communautés; Dans la Figure 4.2, les services Web *WS-Peintre*, *WS-Maçon*, *WS-Electricien*, *WS-Plombier* et *WS-Menuisier* appartiennent simultanément aux deux communautés "*Construction de maison*" et "*Procédure de construction*".

Le fonctionnement d'une communauté s'articule autour du développement de la communauté elle-même, de l'attraction et de la rétention des services Web et du déploiement de ces communautés dans le réseau social de services Web. Une communauté de services Web est principalement établie pour rassembler les services Web ayant les mêmes fonctionnalités. C'est une activité suivi par un concepteur. La première étape consiste à définir la fonctionnalité, par exemple, "Construction de maison". Cette liaison est cruciale car les fournisseurs de services Web utilisent des terminologies différentes pour décrire les fonctionnalités de leurs services Web respectifs. Par exemple, "Construction de maison" et "*Home's construction*" ont la même fonctionnalité.

La deuxième étape de l'établissement d'une communauté consiste à déléguer l'insertion du service Web dans la communauté, au fournisseur de ce dernier. La survivabilité d'un service Web dépend, dans une certaine mesure, du statut de ce dernier dans cette communauté. Si les fournisseurs de services Web tentent d'insérer leurs services Web respectifs un peu partout dans des communautés qui ne sont pas forcément la leur ; Ils risquent d'être sevrèrent sanctionnés, chaque fois qu'ils sont rejetés par les clients. Ceci sera détaillé au paragraphe suivant.



wC_{ij} : Le poids de collaboration du service Web (SW_j) *dans la communauté i* défini par l'équation (6)

$wC_{11} >= wC_{12} >= wC_{13} >= wC_{14} >= wC_{15} >= wC_{16} >= wC_{17} >= wC_{18}$

$wC_{21} >= wC_{22} >= wC_{23} >= wC_{24} >= wC_{25}$

$wC_{31} >= wC_{32}$

Figure 4.2 - Exemple de définition de communautés de services Web dans un réseau social de services Web.

Le démantèlement d'une communauté de service Web est également une activité pilotée par un concepteur, dans un premier temps, durant la phase de création du réseau social de services Web, et se produit à son appréciation. On pourrait envisager, dans des travaux futurs, de déléguer cette tâche à service Web maitre, une fois que le réseau social de services Web a atteint une taille relativement importante. Ce dernier supervise tous les événements dans une communauté tels que l'arrivée de nouveaux services Web,

le départ de certains services Web, le suivi des sanctions sur les services Web en cas de mauvaise conduite, etc. Si en premier lieu, le nombre de services Web dans la communauté est inférieur à un certain seuil et, en second lieu, le nombre de demandes de participation dans les services Web composites émanant des clients sur une certaine période est inférieur à un autre seuil, la communauté pourrait être démantelée. Les deux seuils sont définis par le concepteur. Les services Web qui seront éjectés d'une communauté sont invités à se joindre à d'autres communautés, sous réserve d'évaluer la similarité de leurs fonctionnalités avec les fonctionnalités des communautés existantes.

Bien que les services Web dans une communauté soient en concurrence, ils exposent une attitude coopérative. L'attraction et la rétention des services Web éclairent un autre scénario, qui concerne les services Web appelés à quitter une communauté, si à différentes occasions, il n'arrive pas à participer aux services Web composites en raison de problèmes récurrents.

4. Construction du réseau social de services Web

4.1. Paramètres de qualité des services (QoS) considérés

Ce travail n'est pas limité à un ensemble particulier de paramètres QoS. Tout modèle de QoS pourrait être considéré. La seule contrainte, dans le cadre de cette thèse, est que les paramètres choisis soient collectés sur un système automatisé sans aucune intervention de l'utilisateur, afin de préserver l'aspect automatique de la sélection des services Web. Les paramètres de QoS considérés sont à la fois faciles à comprendre et à mesurer [(Metrouh & Mokhati, 2016), (Metrouh & Mokhati, 2018)].

Pour mesurer les paramètres de QoS considérés, quatre valeurs de temps sont mesurées. Le temps est mesuré en millisecondes:

- *t1*: Temps d'envoi de la requête client.
- *t2*: Temps de réception de la requête par le fournisseur.
- *t3*: Temps d'envoi de la réponse par le fournisseur.
- *t4*: Temps de réception de la réponse par le client.

- **Temps de Réponse (*RT*):** Le temps nécessaire pour traiter une demande du client à partir du moment de son expédition jusqu'à la réception de la réponse.

Le temps de réponse peut être divisé en temps de communication et temps d'exécution [(Rosenberg et al., 2006), (El Saddik, 2006)].

$$RT = t4 - t1 \quad (\mathbf{1})$$

- **Temps d'Exécution (*ET*)**: Temps nécessaire au service pour exécuter la requête.

$$ET = t3 - t2 \quad (\mathbf{2})$$

- **Temps de Communication (*CT*)**: Le temps nécessaire pour transférer la requête et sa réponse.

$$CT = RT - ET \quad (\mathbf{3})$$

- **Coût du service (*SC*):** Le coût est le prix que les utilisateurs doivent payer pour l'invocation du service Web. Le cout constitue le souci majeur pour les utilisateurs. Il est déterminé ou modifié par les fournisseurs de services en fonction de la capacité du service, la propriété marketing ou la politique financière du fournisseur.
- **Disponibilité du service (*SA*):** la probabilité que le service soit disponible.

$$SA = NQ / TQ \quad (\mathbf{4})$$

NQ: Nombre de requêtes réussies.

TQ: Nombre total de requêtes.

4.2. Fonction objective

Un défi critique dans la sélection des services est de savoir comment mesurer la qualité de bout en bout du service composite. La valeur agrégée d'un attribut QoS doit prendre en compte la valeur d'attributs QoS des services individuels participant au service composite. Dans (Jaeger et al., 2004) et son extension (Jaeger et al., 2005), Jaeger et al., ont proposé des fonctions d'agrégation pour certains attributs de qualité de service tels que le temps d'exécution, le coût et le débit, prenant en charge un ensemble complet de modèles structurels pouvant être trouvés dans les processus métiers. D'autres fonctions d'agrégation ont également été développées [(Canfora et al., 2004), (Ardagna & Pernici, 2007), (Ritcher et al., 2012)].

Il existe une hypothèse générale dans la littérature selon laquelle un client a une idée claire de l'importance d'un attribut QoS qui lui permet d'attribuer un poids scalaire à chaque critère de qualité de service. Mais cela peut ne pas être réaliste, d'autant plus que le nombre d'attributs QoS impliqués dans les critères de sélection augmente. Certains chercheurs ont contesté cette hypothèse. Wang (Wang, 2009) a proposé un processus de résolution pour déterminer les poids des attributs QoS basés sur les préférences d'un groupe de participants. Yu et Bouguettaya (Yu & Bouguettaya, 2012) ont proposé deux algorithmes pour calculer l'horizon de service. La détermination de l'horizon d'un ensemble de données nécessite une comparaison par paire de tous les membres de l'ensemble de données, ce qui peut être très coûteux en termes de temps de calcul et d'utilisation de la mémoire. Les algorithmes proposés dans (Yu & Bouguettaya, 2012) exploitent les indices des opérations de service pour calculer l'horizon plus efficacement. Les horizons calculés garantissent d'inclure les meilleurs fournisseurs de services souhaités par l'utilisateur sans aucune intervention de l'utilisateur.

Dans notre cas, les attributs considérés ne sont pas nombreux. Chaque service Web a une disponibilité et un temps de réponse mesurés la dernière fois, qu'il a été sélectionné ; il a également un coût fixe. Le but du client est de maximiser la disponibilité des services Web sélectionnés WS_i, minimiser le temps de réponse et le coût de ces derniers. La fonction objective est définie comme suit:

$$F_{SS} = \omega_{rt}\left(1-\frac{RT_{WSi}-avg_{rt}}{\sigma_{rt}}\right)+\omega_{sa}\left(\frac{SA_{WSi}-avg_{sa}}{\sigma_{sa}}\right)+\omega_{sc}\left(1-\frac{SC_{WSi}-avg_{sc}}{\sigma_{sc}}\right) \qquad (5)$$

ω_{rt} : poids du temps de réponse, $0< \omega_{rt} <1$.

ω_{sa} : poids de la disponibilité du service, $0< \omega_{sa} <1$.

ω_{sc} : poids du coût du service, $0< \omega_{sc} <1$.

RT_{WSi} : Temps de Réponse du service Web WS_i.

SA_{WSi} : Disponibilité du service Web WS_i.

SC_{WSi}: Coût du service Web WS_i.

σ_{rt}: Ecart type du temps de réponse des services Web appartenant au graphe de substitution du service Web WS_i.

σ_{sa}: Ecart type de la disponibilité des services Web appartenant au graphe de substitution du service Web WS_i.

σ_{sc}: Ecart type des coûts des services Web appartenant au graph de substitution du service Web WS_i.

avg_{rt}: temps de réponse moyen des services Web appartenant au graphe de substitution du service Web WS_i.

avg_{sa}: disponibilté moyenne des services Web appartenant au graphe de substitution du service Web WS_i.

avg_{sc}: coût moyen des services Web appartenant au graphe de substitution du service Web WS_i.

Les services Web sont les seuls constituants du réseau social et correspondent aux nœuds. Nous utilisons trois types de liens ou d'associations entre les services Web: associations basées sur la collaboration pour la composition (*C*), associations basées sur la collaboration pour la substitution (*S*) et associations basées sur la recommandation pour la composition (*Rc*). Ces associations sont inspirées de nos travaux [(Metrouh et al., 2012), (Metrouh & Mokhati, 2013), (Metrouh & Mokhati, 2016) et (Metrouh & Mokhati, 2018)].

4.3. Associations de collaboration pour la composition

Ces associations définissent la notion de communauté de services Web. Chaque communauté est organisée comme un arbre du réseau social de services Web. L'intérêt d'une telle organisation est de permettre un accès rapide à la communauté associée à la requête du client. Nous supposons d'abord que pour chaque requête client est associée une et une seule communauté. Ces associations sont évaluées par l'équation suivante:

$$wC_{ij} = |WS_j\ selection| \ / \ |WS_j\ participation| \qquad (\mathbf{6})$$

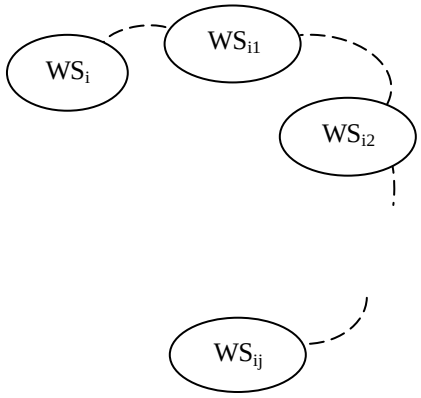
- wC_{ij} est le poids du service Web WS_j dans la communauté i.
- $|WS_j\ selection|$ représente le nombre de fois que WS_j a réellement participé à des scénarios de composition, ainsi sélectionné par le client.
- $|WSj\ participation|$ est le nombre de fois que WS_j a été désigné pour participer à ces scénarios de composition.

L'arbre associé à chaque communauté est trié par ordre décroissant des poids associés aux associations de collaboration (Figure 4.2). La tâche d'intégration de services Web dans le réseau social de services Web ou la création de nouvelles

communautés de services Web est laissée à l'appréciation des fournisseurs de services Web. L'importance de considérer les associations évaluées dans la définition des associations de collaboration est d'éviter que les fournisseurs de services Web ne procèdent inutilement à insérer leurs services Web partout, même dans les communautés qui ne correspondent pas à la catégorie de leurs services Web, espérant maximiser leurs chances de sélection par les clients. Un service Web inséré inutilement dans une communauté qui ne lui correspond pas sera décalé à chaque fois qu'il est rejeté par les utilisateurs à la fin de l'arborescence associée à la communauté. Il sera donc ignoré par le processus de sélection une fois que le réseau social aura atteint une dimension importante. Les associations de services Web avec des poids plus élevés seront préférées.

4.4. Associations de collaboration pour la substitution

Ces associations sont organisées en arbre (Figure 4.3). Une telle organisation permet une identification rapide des substituts possibles pour les services Web défaillants ou indisponibles. Il est inutile que chaque service Web possède son propre arbre de substitution. Une telle redondance augmente considérablement la taille du réseau social. Il doit juste être engagé dans un et un seul arbre de substitution. Ces associations ne sont pas caractérisées pas des poids.

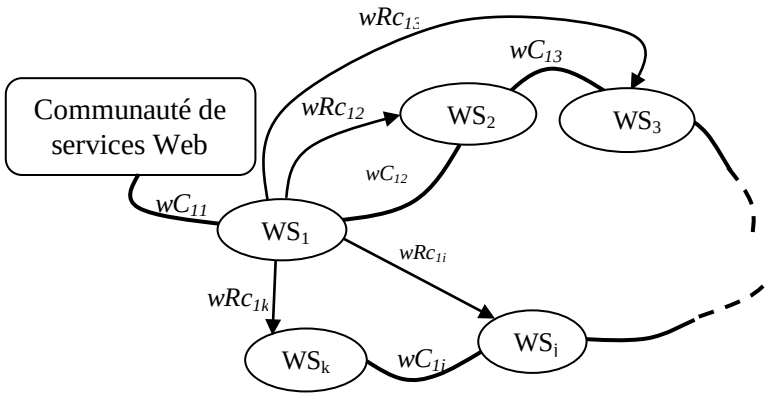


WS_{i1}, WS_{i2}, ..., WS_{ij} sont les substituts probables de WS_i.

Figure 4.3 - Exemple de graphe de substitution d'un service Web.

4.5. Associations de recommandation pour la composition

Dans une même communauté de services Web, un service Web pourrait proposer que de nouveaux services Web résultant d'associations basées sur des recommandations fassent partie d'une composition (voir Figure 4.4).



Association de collaboration pour la composition

Associations de recommandation pour la composition

wC_{ij} : Le poids de la collaboration du service Web (WS_j) dans la communauté *i* défini par l'équation (6).

wRc_{ij} : Le poids de l'association de recommandation pour la collaboration du service Web (WS_i)défini par l'équation (5).

Figure 4.4 – Associations de recommandation et de collaboration pour la composition.

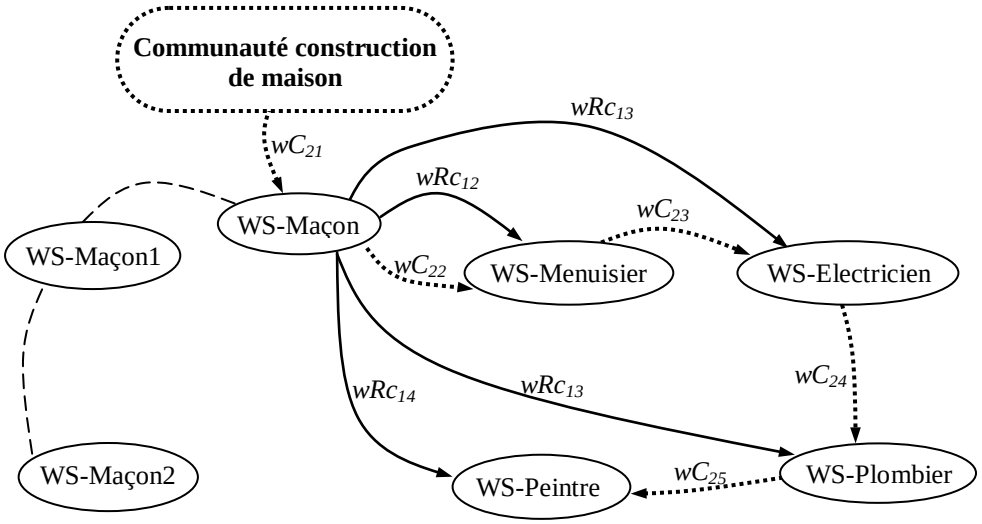
Les services Web résultant d'associations basées sur des recommandations peuvent être nécessaires pour satisfaire une requête utilisateur. Il est souvent rare qu'une requête client puisse être satisfaite par un seul service Web. Ces associations s'appellent des associations basées sur la recommandation pour la composition (*Rc*).

Le poids de l'association basée sur la recommandation pour la composition wRc_{ij} (où WS_i recommande WS_j) est donné par l'équation (6) pour le service Web WS_j.

5. Sélection des services Web atomiques pour la composition

Formellement, nous définissons le graphe *G* (Figure 4.5) associé au réseau social de services Web comme un quadruple (*V*, *C*, *S*, *R*), où *V* est l'ensemble des sommets (services Web), *C* l'ensemble des arcs de *C*ollaboration, *S* l'ensemble des arêtes de *S*ubstitution et *R* l'ensemble des arcs de *R*ecommandation pour la composition. De plus, le graphe direct formé par *V* et *C*

est appelé *CG* (*G*raphe de *C*ollaboration) qui définit les communautés associées aux services Web dont les associations sont évaluées par l'équation dans (6).



Association de *C*ollaboration pour la composition

Association de *C*ollaboration pour la substitution

Association de *R*ecommandation pour la composition

wC_{ij} : Le poids de la collaboration du service Web (WS_j) dans la communauté i défini par l'équation (6).

wRc_{ij} : Le poids de *R*ecommandation *pour la collaboration du* service Web (WS_j) definie par l'équation (5).

$wC_{21} >= wC_{22} >= wC_{23} >= wC_{24} >= wC_{25}$

Figure 4.5 - Exemple de réseau sociaux de services Web.

Le graphe non orienté formé par *V* et *S* s'appelle *SG* (Graph of Substitution) qui définit les substituts probables pour les services Web échoués ou indisponibles, dont les associations n'ont pas de poids. Le graphe direct formé par *V* et *R* est appelé *RCG* (*G*raphe de *R*ecommandation pour la *C*omposition) caractérisé par des poids définis par la fonction objective dans (5). Le sous-graphe formé par la même communauté dans le graphe *RCG* est un graphe orienté complet.

ALGORITHME 1: Sélection des services Web atomique pour la composition

INPUT: *cT (*arbre associé à la communauté extraite de la requête de l'utilisateur dans le *G*raphe de *C*ollaboration *CG*), *RCG (G*raphe des associations de *R*ecommandation pour la *C*omposition), $sT_{(WSi)}$ (arbre de substitution du service Web WS_i dans le graphe de substitution *SG*), *rTime* (Temps de réponse maximal exigé par le client pour sa requête)

OUTPUT: WS_{Sel} (Ensemble de services Web sélectionnés pour la composition)

1: **DEBUT**
2: Marquer *z* dans le graphe *RCG* /* *z* le premier sommet dans *cT* */
3: WS_{Sel} , $x \leftarrow z$
4: **TANT QUE** ∃ un sommet *y* non marqué adjacent à *x* dans $\{RCG - WS_{Sel}\}$ **ET NON** *rTime* **FAIRE**
/* La condition d'arrêt est définie comme suit: soit il n'y a plus de sommets non marqués dans RCG, soit le temps de réponse requis par le client pour sa requête a expiré.*/
5: Marquer *y* dans le graphe *RCG*
6: **SI** *(x, y)* est l'arc sortant du poids le plus élevé **ET** $y \in cT$ **ALORS**
/* Sélectionnez l'arc ayant le poids le plus fort depuis de x dans le graphe *RCG* puis s'assuré que le service Web associé appartient à la communauté extraite */
7: $WS_{Sel} \leftarrow WS_{Sel} \cup \{y\}$
8: **FINSI**
9: $x \leftarrow y$
10: **FIN TANT QUE**
11: $i \leftarrow$ o
12: $j \leftarrow 1$
/* Supprimer les services Web redondants appartenant aux arbres de substitution du service Web déjà sélectionné dans WS_{Sel} */
13: **TANT QUE NON** fin WS_{Sel} **FAIRE**
14: **TANT QUE NON** fin WS_{Sel} **FAIRE**
15: **SI** $WS_{Sel}(j) \in sT(WS_{Sel}(i))$ **ALORS**
16: $WS_{Sel} \leftarrow \boldsymbol{WS_{Sel}} - WS_{Sel}(j)$
17: **FINSI**
18: $j \leftarrow j + 1$
19: **FIN TANT QUE**
20: $i \leftarrow i + 1$
21: **FIN TANT QUE**
22: **RETOUNER** WS_{Sel}
23: **FIN**.

L'algorithme présenté ci-dessous permet de définir le processus de sélection des services Web atomiques en vue de leur composition. Cet algorithme exploite les arbres associés à chaque communauté de service Web pour réduire leur espace de recherche (Figure 4.2). Rappelons qu'à chaque requête est associée à une et une seule communauté. L'algorithme proposé augmente progressivement le nombre de services Web sélectionnés. Les services Web ayant le poids est le plus élevé sont les seuls retenus. Les services Web appartenant à l'arborescence de substitution de tout service Web déjà sélectionné sont ignorés. Cela nous évitera des redondances inutiles. Le client peut affecter la fonction objective qui définit les poids des arcs de recommandation dans le graphe de recommandation pour la composition *RCG*, en augmentant ou diminuant les poids associés à chaque paramètre (disponibilité, coût et temps d'exécution). L'algorithme 1 est largement inspiré du célèbre algorithme de recherche en largeur (BFS), en théorie des graphes. Dans notre cas, le graphe est un arbre et l'algorithme est beaucoup plus simple.

6. Composition des services Web sociaux personnalisés

6.1. Construction des préférences utilisateurs

Le modèle proposé consiste en un ensemble d'attributs. Deux types d'attributs sont proposés: les attributs statiques et les attributs acquis.

6.1.1. Attributs statiques

Définition 6.1.1.1 (Vecteur des attributs statiques): Le tuple VS_i=<*nom*, *prénom*, *date de naissance*> constitue les attributs statiques qui ne varient pas au fil du temps.

Les attributs statiques sont généralement renseignés explicitement par les utilisateurs et sont très utiles pour les identifier. Ils sont représentés par un ensemble de couples (attribut, valeur).

6.1.2. Attributs acquis

Définition 6.1.2.1 (Vecteur des attributs acquis):
le tuple VA_i=<$communaute_1$,...,$communaute_p$> constitué par chaque communauté de service Web et qui forme les préférences de l'utilisateur où *p* est le nombre de communautés pour l'utilisateur *i*.

Les attributs acquis représentent les intérêts des utilisateurs qui évoluent dans le temps en fonction de leurs interactions avec le système. Ils sont également représentés par un ensemble de couples (attribut, valeur)

6.2. Algorithme pour la composition des services Web sociaux personnalisés

L'algorithme présenté ci-dessous exploite tous les concepts déjà définis et détaillés : réseaux sociaux de services Web, communautés de services Web, profil utilisateur et ses attributs statiques et acquis et l'algorithme de sélection des services Web atomiques pour présenter une approche innovante pour la composition des services Web sociaux personnalisés.

ALGORITHME 2: Composition des services Web sociaux personnalisés

Etape 1: Identifier le client et identifier ses attributs statiques.

Etape 2: Récupérer les attributs acquis relatifs au client.

Etape 3: Vérifiez si la communauté extraite est dans les attributs acquis du client.

Etape 4: Exécuter **Algorithme 1** pour les services Web atomiques sélectionnés.

Etape 5: Comparer les services Web sélectionnés avec ceux répertoriés dans les attributs acquis du client et conserver uniquement les services Web avec le poids le plus élevé.

Etape 6: Exécuter les services Web.

Une fois que la requête du client est introduite, elle est analysée et la communauté relative à celle-ci est extraite. Les informations relatives au profil du client sont extraites. Le client est identifié par ses attributs statiques et ses préférences sont extraites de ses attributs acquis puis communiqués au module de sélection des services Web atomiques. L'algorithme 1 de sélection des services Web atomiques est lancé, et les services Web retournés par ces derniers sont comparés à ceux stockés dans son profil et plus précisément ses attributs acquis. Ceux ayant le poids le plus fort sont retenus. Le client a la possibilité de garder ceux stockés dans ses attributs acquis. Les services Web sont ensuite exécutés.

7. Tolérance aux pannes dans le cadre de l'approche proposée

Au paragraphe précédent l'approche proposée pour la composition des services Web sociaux personnalisés utilise une solution pour la substitution des services Web en cas d'échec ou d'indisponibilité de ces derniers qui ne garantit pas un substitut de qualité. Les substituts sont choisis arbitrairement.

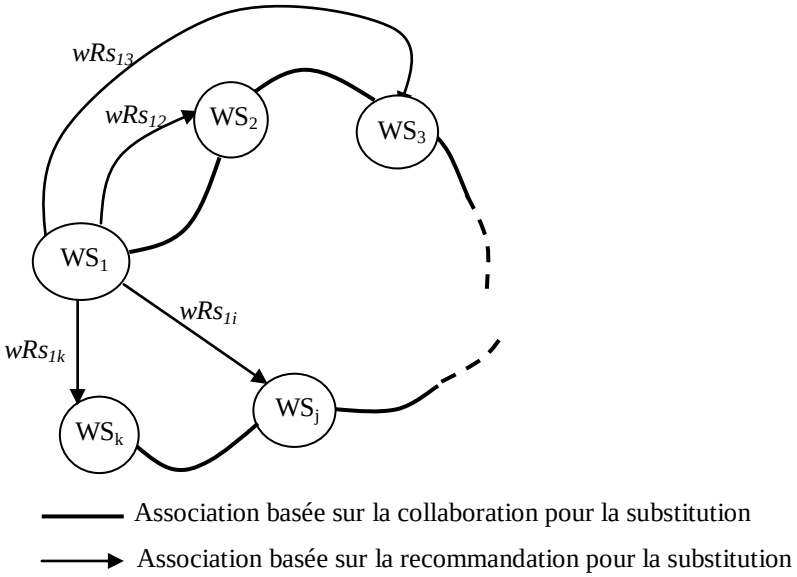
Dans ce paragraphe nous présentons notre approche pour la sélection des services Web pour la substitution de ces derniers, qui garantit le meilleur substitut disponible au service Web défaillant ou indisponible, et ainsi remédier aux insuffisances qui pourraient nuire aux attentes de l'utilisateur et au bon fonctionnement de l'approche proposée pour la composition des services Web sociaux personnalisés (Metrouh et al., 2016).

Nous détaillons dans ce qui suit la structure du graphe *RSG* (Graphe basé sur la Recommandation pour la Substitution). Après cela, nous détaillons l'algorithme proposé pour la sélection de services Web dans le cas de la substitution. Nous utilisons deux types de liens ou d'associations entre services Web: les associations basées sur la collaboration pour la substitution (*S*) et les associations basées sur la recommandation pour la substitution (*Rs*). Toujours dans le même contexte c'est-à-dire le réseau social de services Web. Ces associations sont inspirées de nos travaux antérieurs [(Metrouh et al., 2012), (Metrouh & Mokhati, 2013)].

7.1. Associations de recommandation pour la substitution

Dans le même arbre de substitution (présenté au paragraphe 4.4), un service Web pourrait proposer que de nouveaux services Web résultant d'associations fondées sur des recommandations puissent être utilisés en remplacement en cas de défaillance ou d'indisponibilité de ces derniers (voir la figure 4.6). Ces associations sont appelées associations basées sur la recommandation pour la substitution (*Rs*).

Le poids de l'association basée sur la recommandation pour la substitution wRs_{ij} (où WS_i recommande WS_j) est donné par l'équation (5) pour le service Web WS_j.



wRs_{ij}: Le poids de l'association de recommandation pour la substitution du service Web (WS_j) défini par l'équation (5).

Figure 4.6 – Associations de recommandation et de collaboration pour la substitution.

7.2. Algorithme de sélection des services Web pour la substitution

Formellement, nous définissons dans le même graphe *G* associé aux réseaux sociaux des services Web (figure 4.5) *Rs* l'ensemble des arcs de recommandation pour la substitution. *S* étant l'ensemble des arêtes de substitution déjà défini dans le graphe *G* (figure 4.3).

Le graphe indirect formé par *V* l'ensemble des nœuds (services Web) et *S* l'ensemble des arêtes de substitution est *SG* (Graphe de Substitution) qui définit les substituts probables des services Web suite à un échec ou indisponibilité. Ses associations n'ont pas de poids. Le graphe direct formé par *V* et *Rs* est appelé *RSG* (Graphe de Recommandation pour la Substitution) caractérisé par des poids définis par la fonction objectif dans (5). Le sous-graphe associé à chaque service Web dans le graphe *RSG* est un graphe orienté complet.

La substitution est envisagée en cas de défaillance ou d'indisponibilité d'un service Web. Un service Web adjacent résultant de l'association basée sur la recommandation pour la substitution dans le graphe *RSG* est sélectionné (voir Algorithme 3). Les services Web adjacents du service Web en échec ou indisponible dans *RSG* sont triés

par ordre décroissant en fonction du poids de l'arc sortant. Le service Web avec le poids le plus élevé est testé. S'il est disponible et non défaillant, il est sélectionné. Sinon, le suivant est sélectionné et testé, ainsi de suite.

ALGORITHME 3: Sélection des services Web pour la substitution

1 : **INPUT**: WS_i (service Web en echec ou indisponible), sT_i (L'arbre de substitution de WS_i dans le graphe de substitution *SG*), *RSG* (Graphe des associations de recommandation pour la subtitution)

2 : **Résultat**: WS_j (Service Web sélectioné)

3 : **DEBUT**

4 : $x \leftarrow WS_i$

5 : $i \leftarrow 0$

6 : TANT QUE ∃ ***UN SOMMET*** *y* non marqué et adjacent à *x dans RSG* **FAIRE**

7 : *Queue*(*i*) ← *y* /* dans Queue les services Web sont tries dans l'ordre décroissant selon le poids de l'arc sortant */

8 : Marquer *y*

9 : $i \leftarrow i + 1$

10: **FIN TANT QUE**

11: $i \leftarrow 0$

12: **TANT QUE NON** fin *Queue* **FAIRE**

13: **SI** *Queue*(*i*) disponible et fonctionne **ALORS**

14: $WS_j \leftarrow y$

15: **GOTO** :19

16: **FIN SI**

17: $i \leftarrow i + 1$

18: **FIN TANT QUE**

19: **RETURN** WS_j

20: **FIN**.

---

8. Implémentation et évaluation

8.1. Approche de composition des services Web personnalisés

Dans cette section, nous décrivons les résultats expérimentaux que nous avons obtenus. L'algorithme est implémenté en Java EE v6 et fonctionne sous le système d'exploitation Windows 7. Les temps CPU ont été obtenus sur un i3 à 3,33 GHz avec 6 Go de RAM. Le tableau suivant (Tableau 4.1) montre le temps requis par l'algorithme proposé pour retourner l'ensemble des services Web sélectionnés. Le graphe *CG* choisi est aléatoire dans le sens où le nombre de sommets et les poids des associations de composition pour la composition sont choisis arbitrairement. Il en est de même pour les poids des associations basées sur la recommandation pour la composition dans le graphe *RCG*, les données sont requis au démarrage du système pour éviter le problème du "démarrage à froid". Plusieurs approches ciblant différents domaines cherchent à résoudre ce problème [(Golbeck, J., 2009), (Klein et al., 2009)]. Cette problématique est traitée plus loin dans cette section. Le nombre de communautés créées n'affecte pas le temps d'exécution. L'interface utilisateur est initialement définie de manière simple afin d'exploiter les communautés existantes qui sont indexées au fur et à mesure de leur création.

L'implémentation fournie dans le cadre de cette thèse utilise trois listes d'adjacence pour les graphes *CG*, *RCG* et *SG*. Le temps est mesuré en millisecondes. La liste d'adjacence associée au graphe *CG* a une structure simple. Elle est utilisée pour vérifier que le service Web appartient effectivement à la communauté sélectionnée. Le nœud de la liste d'adjacence contient le nom du sommet qui fait référence à un service Web, son parent et le poids de l'arc entre le parent et ce dernier, donné par l'équation (6). La liste d'adjacence associée au graphe *SG* a également une structure simple. Elle est utilisée pour définir les substituts probables d'un service Web. Le nœud de la liste d'adjacence contient le nom du sommet qui fait référence à un service Web et à son parent. La structure de la liste d'adjacence associée au graphe *RCG* est définie comme suit: Le nœud de liste d'adjacence contient le nom du sommet qui se réfère à un service Web, ses parents, le coût, la disponibilité et le temps d'exécution de ce dernier.

Supposons que *rTime* (temps de réponse maximum requis par le client pour sa requête) soit très important (donc sans aucune influence sur l'exécution de la requête) et calculons la complexité de l'espace et la complexité temporelle dans le cas de notre algorithme.

- **Complexité spatiale:** comme le nombre de sommets est connu d'avance, et que des structures de données supplémentaires sont utilisées pour déterminer quels sommets ont déjà été ajoutés à la file d'attente, la complexité de l'espace peut être exprimée comme suit: $O(/V/)$, où $/V/$ est la cardinalité de l'ensemble des sommets, et $/M/$ le nombre moyen de sommets par arbre de substitution ($/M/ < /V/$). Le graphe est représenté par une liste d'adjacence qui occupe $O(2/V/)$ espace de mémoire.
- **Complexité temporelle:** la complexité du temps est exprimée par l'expression $O(/V/^2)$. Chaque sommet et chaque arc sont explorés.

Calcul du temps d'exécution:

Temps total = Temps pour construire la liste d'adjacence associée au graphe *RCG* + Temps pour parcourir le graphe *RCG* et calculer les poids dans le sous-graphe associé à la communauté extraite de la requête client, donnée par l'équation (5) + temps pour vérifier que le sommet appartient effectivement au sous-graphe formé par la communauté dans le graphe *CG* + temps pour vérifier que le sommet n'appartient à aucun graphe de substitution *SG* associé aux services Web déjà sélectionnés + temps pour vérifier si la communauté extraite est dans les attributs acquis du client + temps pour comparer les services Web sélectionnés avec ceux listés dans les attributs acquis du client et conserver uniquement les services Web avec les poids les plus élevés.

No	Données						Temps enregistré en (ms)
	Graphe *G*		Graphe *CG*	Graphe *RCG*	*cT*		
	Nœud	Arête	Arc	Arc	Nœud	Arc	
1	450	1854	477	980	25	100	242
2	600	3216	651	2350	35	400	925
3	800	6123	982	4500	45	900	2245
4	1100	9265	1265	8100	55	1600	4015
5	1400	18456	1601	17950	65	2500	6241
6	1500	40213	1724	27650	75	3600	9158
7	1800	51244	2013	37250	85	4900	11453
8	2000	60213	2207	46900	95	6400	16592
9	2200	71523	2315	56550	105	8100	20157
10	2300	81245	2501	66250	115	10000	25164
11	2400	90245	2645	75850	125	12100	31258
12	2600	101234	2804	85250	135	14400	35149
13	2800	110251	3002	94400	145	16900	42587

Table 4.1 - Temps nécessaire pour renvoyer l'ensemble des services Web sélectionnés pour la composition.

En calculant le temps, seul le temps de construction de l'arbre a été pris en considération. Certaines sections de code, telles que l'impression de la liste des arêtes et certaines vérifications des erreurs d'entrée du client, ont été mises en commentaire avant d'exécuter le code sur l'ensemble des données.

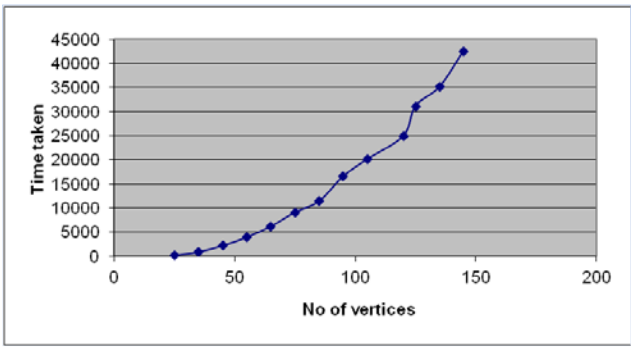


Figure 4.7 - Temps vs Nombre de sommets dans l'arbre associé à la communauté extraite *cT*.

.

À partir du graphe de la figure 4.7, nous pouvons facilement conclure que l'algorithme proposé prend plus de temps une fois que le nombre de sommets (services

Web) augmente dans l'arbre associé à la communauté extraite de la requête client. Le graphe forme la courbe n^2.

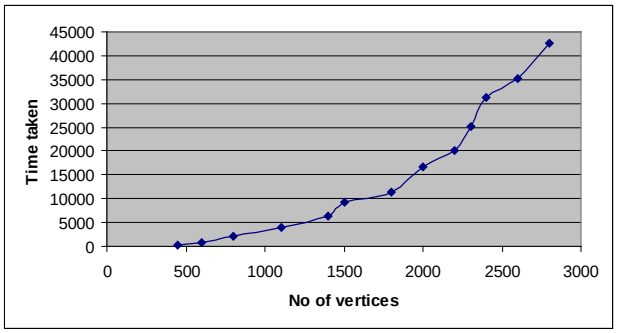


Figure 4.8 - Temps Vs Nombre de sommets dans le graphe de collaboration *CG*.

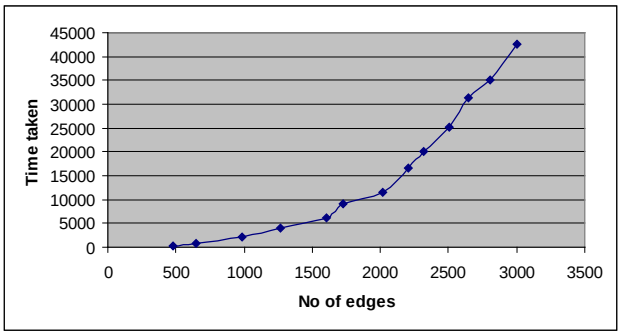


Figure 4.9 – Temps Vs Nombre d'arcs dans le graphe de collaboration *CG*.

On remarque également sur la figure 4.8 que le nombre total de sommets dans le graphe de collaboration *CG* n'a aucune influence sur le temps de parcours; vu sa structure simple. Chaque communauté est associée à un et un seul arbre (voir figure 4.2). Le nombre total d'arcs dans le graphe *CG* (voir figure 4.9) est également sans influence sur le temps d'exécution. Les arcs considérés dans le graphe *CG* sont uniquement ceux appartenant à l'arbre associé à la communauté extraite.

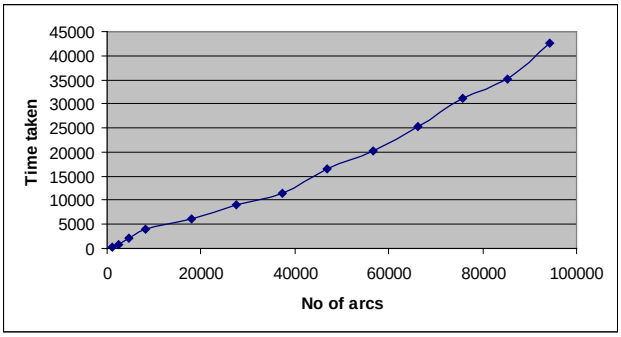


Figure 4.10 - Temps Vs Nombre d'arcs dans le graphe *RCG*.

Sur la figure 4.10, le nombre total d'arcs dans le graphe des associations basées sur la recommandation pour la composition *RCG* est également sans influence sur le temps d'exécution. Les sommets considérés dans le graphe *RCG* sont uniquement ceux appartenant à l'arbre associé à la communauté extraite et leurs arcs dans le graphe *RCG*.

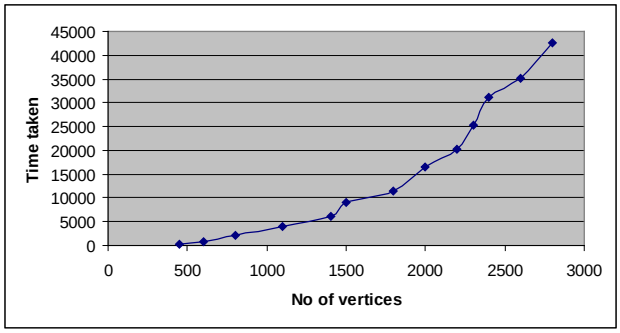


Figure 4.11 - Temps Vs Nombre de sommets dans le graphe *G* associé au réseau social de services Web.

Nous pouvons conclure que le nombre total de sommets dans le graphe *G* associé au réseau social des services Web n'a aucune influence sur le temps d'exécution (voir figure 4.11). Le nombre total d'arêtes dans *G* (voir figure 4.12) est également sans influence sur le temps d'exécution. Le concept "communauté de services Web" est très utile et a pu réduire considérablement l'espace de recherche des services Web.

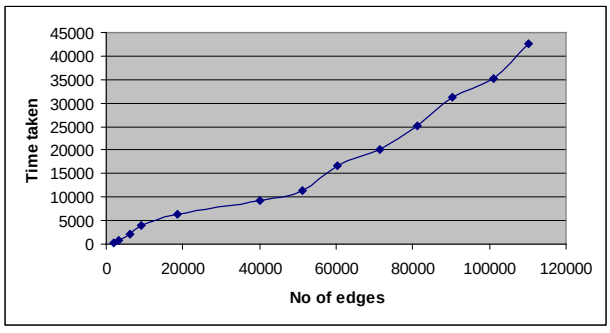


Figure 4.12 - Temps Vs Nombre d'arêtes dans le graphe *G* associé au réseau social de services Web.

Nous devrions également faire face à un autre problème, le "démarrage à froid" qui se produit au début de l'utilisation du système, dans des situations critiques où le

système manque de données pour effectuer un filtrage personnalisé de qualité. Dans notre contexte, il existe deux (02) situations:

- **Le réseau social des services Web est en construction :** Il faut d'abord envisager de créer le graphe des associations basées sur la collaboration pour la composition (*CG*) ainsi que le graphe des associations basées sur la collaboration pour la substitution (*SG*), s'ils n'existent pas encore. Nous recommandons dans ce cas de les créer de manière incrémentielle, de façon à inclure la notion de communauté dès le début. Ceci pourrait être demandé aux fournisseurs au moment de l'enregistrement de leurs services Web.
- **Pour des graphes basés sur les associations de collaboration existants (*CG* et *SG*) :** Les graphes des associations basées sur la recommandation pour la composition *RCG* posent toujours un problème. Nous pourrions considérer les arcs de poids zéro dans les graphes *RCG*, et fixer un seuil pour le nombre de services Web sélectionnés. La notion de poids le plus élevé n'aurait pas de sens au début de l'utilisation du système. Les sommets (services Web) pourraient être choisis arbitrairement.

8.2. Approche de substitution des services Web

Dans cette section, nous décrivons les résultats expérimentaux que nous avons obtenus. L'algorithme de sélection des services Web dans le cas de leur substitution est implémenté en Java EE v6 et fonctionne sous le système d'exploitation Windows 8.1. Les temps CPU ont été obtenus sur un i3 à 3,33 GHz avec 6 Go de RAM. Le tableau suivant (Table 4.2) montre le temps requis par notre algorithme pour retourner les services Web sélectionnés (Metrouh & Mokhati, 2016).

N°	Noeud	Arcs	Temps requis (ms)
1	450	1854	242
2	600	3216	925
3	800	6123	2245
4	1100	9265	4015
5	1400	18456	6241
6	1500	40213	9158
7	1800	51244	11453
8	2000	60213	16592
9	2200	71523	20157
10	2300	81245	25164
11	2400	90245	31258
12	2600	101234	35149
13	2800	110251	42587

Table 4.2 - Temps nécessaire pour renvoyer les services Web sélectionnés (substituts).

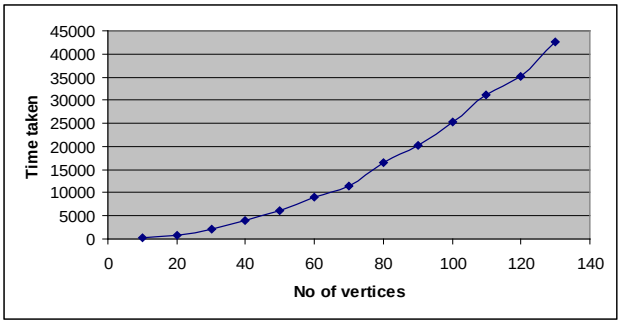


Figure 4.13 - Temps Vs nombre de sommets dans le graphe de substitution.

À partir du graphe de la figure 4.13, nous pouvons facilement conclure que l'algorithme proposé prend plus de temps à mesure que le nombre de sommets augmente dans le graphe de substitution. Le graphe forme la courbe "n^2".

9. Conclusion

L'architecture présentée dans ce chapitre pour la composition dynamique des services Web personnalisés, est fondée sur une approche innovante de sélection et de

composition de services basée sur des attributs fonctionnels et non fonctionnels et des préférences individuelles. Les services Web sont regroupés en communautés. Une communauté rassemble des services Web ayant des fonctionnalités similaires. Nous avons abordé plusieurs aspects, y compris établir / démanteler une communauté nouvelle / existante, attirer de nouveaux services Web vers une communauté existante et conserver les services Web existants dans une communauté.

Nous avons défini trois types d'association dans un réseau social de services Web : (1) associations de collaboration pour la composition formées à partir de communautés de services Web, (2) associations basées sur des recommandations pour la composition au sein de ces mêmes communautés et (3) associations basées sur la collaboration pour la substitution. L'approche proposée exploite ces trois types d'associations dans le cadre de deux algorithmes : Algorithme 1 pour la sélection de services Web dynamiques et Algorithme 2 pour la composition de services Web personnalisés.

Une série d'expériences montre que cette approche de recommandation des services Web obtient un taux de sélection intéressant et un meilleur score dans le but de leur composition. A fin d'améliorer l'approche proposée, nos travaux futurs doivent se concentrer sur les trois points suivants: 1) Améliorer le modèle de profil de l'utilisateur utilisé et dériver plusieurs instances d'attributs acquis en fonction des éléments contextuels; 2) Prendre en considération la mise à jour dynamique du vecteur des attributs acquis de l'utilisateur dans le contexte des réseaux sociaux de l'utilisateur; 3) Appliquer notre approche dans un contexte réseau, et prendre en compte les retards de communication.

Les substituts probables d'un service Web en cas d'échec ou d'indisponibilité sont choisis arbitrairement, dans le cadre de la solution proposée pour la composition des services Web personnalisés. Une solution innovante basée toujours sur les réseaux sociaux de services Web pour améliorer la sélection des substituts probable d'un service est proposée. Nous avons défini des associations basées sur la recommandation pour la substitution dans le même réseau social de services Web. Les associations basées sur la collaboration pour la substitution étant déjà définies. L'exploitation combinée de ces deux types d'associations dans le cadre d'un algorithme de sélection de services Web en cas de substitution nous a permis de réduire considérablement cette tâche.

Un aspect important de l'approche proposée pour la composition des services Web personnalisés est son extensibilité. Avec un minimum d'effort et en réutilisant les concepts déjà définis, nous sommes arrivés à traiter la problématique relative à l'indisponibilité ou à l'échec d'exécution d'un service Web en cours d'exécution.

Le travail actuel pourrait être étendu en introduisant la notion de l'aspect temporel dans la définition des paramètres QoS des services Web. Nous pourrions également revoir les associations basées sur la recommandation et suggérer que les utilisateurs participent davantage au processus de recommandation pour la substitution.

Conclusion générale

Sommaire

1. Bilan 122
2. Orientations des futures recherches 126

1. Bilan

Les services Web sont des programmes autonomes qui peuvent être exécutés via des protocoles standards d'Internet (Atzori et al, 2010). Il existe de nombreux services sur le Web et chacun a une fonctionnalité limitée (Alrifai & Risse, 2009). Dans de nombreux cas, un seul service Web n'est pas suffisant pour répondre à une requête d'un utilisateur, plusieurs services doivent être combinés selon un modèle de composition pour atteindre un objectif spécifique (Argawal et al, 2005). Le sujet de recherche abordé dans le cadre de cette thèse est celui de tenter de conformer les préférences des utilisateurs en obtenant leurs désirs de base pour le processus de composition. Afin de fournir une solution optimale, seuls les services Web atomiques ont été pris en considération.

La composition des services Web personnalisés est un problème difficile en théorie et en pratique, car il s'agit d'un domaine émergent, comme en témoigne le manque de systèmes automatisés de composition de services publics déployés dans le monde commercial. Bien qu'il existe des approches pour la sélection de produits, les services ont des propriétés non fonctionnelles ou des attributs QoS qui influencent leur crédibilité et leur consommation. Les propriétés non fonctionnelles ou les attributs QoS des services Web sont difficiles à déterminer et à contrôler pour l'utilisateur. Les utilisateurs sont généralement réticents à décrire leurs préférences complètes, et sont encore moins enclins à attribuer des pondérations relatives à leurs préférences, en particulier lorsque les effets et les conséquences de leurs contributions sont inconnus ou indéterminés. De plus, les utilisateurs peuvent même ne pas être conscients de leurs préférences implicites. La capture et la spécification des préférences utilisateur constituent l'un des problèmes les plus complexes du processus de composition et de recommandation des services Web.

Dans le cadre de cette thèse, nous proposons une nouvelle approche pour la composition des services Web personnalisés basés sur les réseaux sociaux de services Web, dans un environnement dynamique. Cette approche utilise les réseaux sociaux de services Web pour la sélection et la composition de ces derniers. Le travail présenté diffère des approches existantes dans trois aspects majeurs. Premièrement, il a été montré que les services Web peuvent socialiser ! Et que les réseaux sociaux de services diffèrent des réseaux sociaux conventionnels. Les réseaux sociaux conventionnels sont basés sur la coopération absolue et l'assistance mutuelle entre leurs membres (c'est-à-dire sans concurrence). Par contre, les services Web dans les réseaux sociaux sont particulièrement compétitifs. Les services Web sont les seuls constituants du réseau social. La plupart des approches proposées dans la littérature pour la composition des services Web se concentrent sur la description des services Web eux-mêmes et négligent leurs interactions. Notre approche tire parti de la puissance des réseaux sociaux et des principes des systèmes de recommandation pour définir les interactions entre les services Web. La construction de tels réseaux nécessite une représentation des différentes interactions dans lesquelles les services Web sont intervenus. Le processus de composition des services Web proposé est basé sur ces interactions et sur la

définition d'un modèle de QoS. Deuxièmement, l'espace de recherche des services Web est considérablement réduit en définissant des communautés dans le réseau social de services Web. La notion de communauté de services Web est définie comme étant une relation entre des services Web caractérisés par des poids. Un service Web peut appartenir à différentes communautés. Une communauté réunit un ensemble de services Web répondant aux mêmes exigences fonctionnelles. Troisièmement, l'approche proposée se distingue par l'implication directe des fournisseurs de services dans la création des réseaux sociaux de services Web.

La terminologie et les définitions relatives aux services Web utilisés tout au long de cette thèse sont présentées. Les bases des technologies des services Web ont été abordées, en commençant par introduire les paradigmes qui gagnent de plus en plus d'élan dans le milieu universitaire et l'industrie, à savoir l'informatique orientée services (SOC) et les architectures orientées services (SOA). Nous avons également discuté en détail les trois technologies piliers pour les services Web: WSDL, SOAP et UDDI. Un aperçu des technologies clés: les services Web, les réseaux sociaux basés sur le Web, les systèmes de recommandation et la personnalisation des services Web sont ensuite présentées.

Le problème de la sélection et de la recommandation des services Web sociaux est étudié, trois technologies clés qui peuvent être intégrées pour y remédier sont identifiées : les services Web, les réseaux sociaux de services Web et les systèmes de recommandation. Un aperçu sur la sélection des services sociaux et du problème de la recommandation, mettant en évidence les défis actuels est présenté. Une revue de la littérature relative à l'approche proposée pour la sélection des services Web est présentée. Plus précisément trois approches de sélection et de recommandation de services sociaux: réputation, recommandation et renvoi. Nous proposons également une étude comparative des différentes approches de sélection des services Web et la façon dont chacun des défis (démarrage à froid, parcimonie, confidentialité, évolutivité et saisie des préférences des utilisateurs) a une incidence sur chacune des approches de sélection des services Web.

Le problème de la composition des services Web est présenté en détail. Le plein potentiel des services Web ne peut être atteint que si une collaboration est envisagée

entre les fournisseurs et les consommateurs de services Web, qui interagissent pour atteindre certains objectifs. Le problème de la composition de services est décrit en détail, ainsi que les différents types d'approches de composition qui existent dans la littérature. Une étude comparative des différentes approches est présentée. Les avantages et les inconvénients de chaque approche sont discutés. Nous fournissons une classification structurée de toutes les approches en fonction de leur logique, de sorte que les approches appartenant à une classe spécifique ne diffèrent que pour des aspects mineurs (principalement techniques). Ensuite, pour chaque classe, nous discutons de l'idée fondamentale sous-jacente et nous dressons la liste des différentes faiblesses. Enfin, nous soulignons les principales limites de l'état de l'art et nous affirmons qu'une notion douce de recommandation (basée principalement sur les réseaux sociaux de services Web) se cache derrière de telles faiblesses. En conséquence, nous préconisons le besoin d'une nouvelle approche pour la composition des services Web fondée sur les réseaux sociaux de ces derniers.

Le cœur des contributions dans le cadre de cette thèse est la proposition d'une approche innovante de composition des services Web sociaux et dynamiques basée sur les attributs fonctionnels et non fonctionnels de ces derniers et les préférences individuelles des utilisateurs. Nous avons décrit comment créer les réseaux sociaux de services Web, ainsi que les différents graphes associés à ces derniers. À cette fin, nous avons introduit une définition de la notion de communauté de services Web qui repose sur l'aspect fonctionnel de ces derniers, ainsi qu'un modèle de QoS. Le problème de la composition des services Web est formellement présenté et sa solution discutée. Dans cette approche, la composition des services Web est liée aux réseaux sociaux et prend en compte l'historique d'invocation des services par les utilisateurs. Les expérimentations de l'approche proposée pour la composition dynamique des services Web personnalisés sont également présentées, ainsi qu'une brève analyse de la complexité de calcul des algorithmes proposés.

Enfin, une approche prometteuse pour la sélection des services Web dans le but de leur substitution en cas de panne ou d'indisponibilité de ces derniers est proposée. Des évaluations expérimentales de l'approche proposée ont également étaient fournies.

2. Orientation des futures recherches

En ce qui concerne l'efficacité de l'approche de composition proposée dans le cadre de cette thèse, plusieurs extensions pourraient être envisagées. Un de ces travaux consiste à étudier comment représenter de manière efficace des profils utilisateurs riches en utilisant des structures de données adéquates. Nous croyons également que les algorithmes proposés pourraient être améliorés et optimisés. Il convient de noter que la motivation derrière de nombreuses approches de composition de service souffre de devenir une réalité en raison de l'absence de "*benchmarks*" de services Web existants, ou d'algorithmes de référence pour fournir une façon standard de comparer une technique de composition de services avec une autre. Par conséquent, nous n'avons aucun moyen de déterminer la qualité de notre approche par rapport aux approches proposées dans la littérature, au cours des processus d'évaluation présentés au chapitre 4. Nous avons dû nous appuyer sur certains exemples fréquemment utilisés dans le domaine des services Web. Par conséquent, un travail futur serait de créer une plate-forme uniforme où notre approche de composition pourrait être comparée avec les autres techniques existantes. Plus précisément, il sera très intéressant de comparer notre approche dans sa mise en œuvre complète et ses capacités à certaines approches de composition basées sur la recommandation. En outre, les exemples utilisés dans le cadre de cette thèse se limitent au domaine du commerce électronique. Notre approche pourrait être pleinement utilisée dans le domaine de la bioinformatique et l'énergie électrique, par exemple.

Enfin, nous avons montré comment les réseaux sociaux de services Web associés aux principes des systèmes de recommandation pourraient être utilisés pour résoudre le problème de la composition des services Web personnalisés. Cependant, il existe de nombreuses autres voies de recherche où les réseaux sociaux de services Web pourraient être exploités pour résoudre certains problèmes de génie logiciel par exemple. Une conséquence directe de la motivation dérivée de ce travail serait d'appliquer ces derniers au problème de la découverte et la composition de l'Internet des objets et la composition des micro-services qui sont nouveaux dans le domaine du génie logiciel.

Publications découlant de cette thèse

- **Journaux internationaux avec comité de lecture**

1. Metrouh, A., Mokhati, F. (2018). Flexible Web services integration: A novel personalised social approach. *Journal of Experimental & Theoretical Artificial Intelligence*. Taylor & Francis Group. Vol. 30, no. 3, pp. 441–456. doi: 10.1080/0952813X.2018.1430862.

- **Conférences internationales avec comité de lecture**

1. Metrouh, A., Seridi-Bouchelaghem, H., Mokhati, F. (2012). Web Services Discovery - A Novel Social Networks Approach based on Communities. Proceedings of the 14th International Conference of the Enterprise Information Systems (ICEIS), Wroclaw, Poland, 28 June - 1 July (2, pp. 316–319). SciTeSeer, ISBN: 978-989-8565-11-2,
   doi: 10.5220/ 0004094103160319.

2. Metrouh, A., Mokhati, F. (2013). Social Web Services Discovery: A Community-Based Approach. Proceedings of the 15th International Conference on Information Integration and Web-based Applications & Services (iiWAS), Vienna, Austria, December 02-04 (pp. 275–279). ACM, ISBN: 978-1-4503-2113-6, doi: 10.1145/2539150.2539257.

3. Metrouh, A., Monfort V. (2013). Selecting Web services on the fly "According to Dynamic Social Communities Creation". Proceedings of the the 2nd International Workshop on Web Intelligence WEBI (ICEIS 2013), Angers, France, July 04-07 (pp. 82–89). SciTeSeer, ISBN: 978-989-8565-63-1, doi:10.5220/0004605500820089.

4. Metrouh, A., Mokhati, F. (2016). A Novel Social Networks Approach Based on QoS for Web Services Selection. Proceedings of the Mediterranean Conference on Pattern Recognition and Artificial Intelligence (MedPRAI), Tebessa, Algeria, November 22 - 23 (pp. 92–97). ACM, ISBN: 978-1-4503-4876-8. doi: 10.1145/3038884.3038899.

Bibliographie

Adomavicius, G., Tuzhilin, A. (2005a). Personalization technologies: a process oriented perspective. *Communications of the ACM*, vol. 48, no. 10, pp. 83-90.

Adomavicius, G. and Tuzhilin, A. (2005b). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions*, vol. 17, no. 6, pp. 734-49.

Agarwal, V., Chae, G., Dasgupta, K., Karnik, N., Kumar, A., Mittal, S., Srivastava, B. (2005). Synthy: A system for end to end composition of Web services. *Web Semantics Science Services and Agents on the World Wide Web*, vol. 3, no. 4, pp. 311-339.

Akkiraju, R., Verma, R., Goodwin, R., Doshi, P., Lee, J. (2004). Executing Abstract Web Process Flows. Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS) Workshop on Planning and Scheduling for Web and Grid Services, Whistler, British Columbia, Canada, June 3-7 (pages 9-15).

Alrifai, M. and Risse, T. (2009). Combining global optimization with local selection for efficient qos-aware service composition. Proceedings of the 18th international conference on World Wide Web, Madrid, Spain, April 20-24, (pp. 881–890).

Ardagna, D. Pernici, B. (2007). Adaptive Service Composition in Flexible Processes. *IEEE Trans. on Software Eng.*, vol. 33, no. 6, pp. 369-384.

Arroyo, L., De Meo, P., Ursino, D. (2010). Trust and Reputation in Social Internetworking Systems. Proceedings of the Workshop on Adaptation in Social and Semantic Web, Hawaii, USA, June 21st, AACHEN: RWTH, vol. 590 CEUR Workshop Proceedings, pp. 42-53, ISSN: 1613-0073.

Atzori, L., Iera, A., Morabito, G. (2010). The internet of things: A survey. *Computer networks*, vol. 54, no. 15, pp. 2787-2805.

Ba, C. (2016). An Exact Cover-Based Approach for Service Composition. Proceedings of the IEEE International Conference on Web Services (ICWS), San Francisco, USA, Jun 27 – July 2 (pp. 631-636).

Baida, Z., Gordijn, J., Omelayenko, B., Akkermans, H. (2004). A shared service terminology for online service provisioning. Proceedings of the Sixth International Conference on Electronic Commerce (ICEC04), Delft, Netherlands, October 25-27 (pp.1–10)

Bajaj, S., Box, D., Chappell, D., Curbera, F., Daniels, G., Hallam-Baker, P., Hondo, M., Kaler, C., Maruyama, H., Nadalin, A., Orchard, D., Prafullchandra, H., von Riegen, C., Roth, D., Schlimmer, J., Sharp, C., Shewchuk, J., Vedamuthu,

A., Yalcinalp, U., (2006):Web services policy 1.2 attachment (ws-policyattachment). http://www.w3.org/Submission/WS-PolicyAttachment/

Battle, S. (Hewlett Packard) et al. Semantic Web Services Framework (SWSF) Overview, W3C Member Submission. https://www.w3.org/Submission/ SWSF/, Sept 2005. [Online, accessed on April 7, 2018].

Bellwood, T.: UDDI version 2.04 API specification (2002). Available from http://uddi.org/pubs/ProgrammersAPI-V2.04-Published-20020719.htm

Benatallah, B., Sheng, Q. Z., Dumas, M. (2003). The Self-Serv Environment for Web Services Composition. *IEEE Internet Computing*, vol. 7, no. 1, pp.40-48.

Berardi, D., Calvanese, D., De Giacomo, G., Hull, R., Lenzerini, M., Mecella, M. (2005a). Modeling Data & Processes for Service Speci_cations in Colombo. Proceedings of the Open Interop Workshop on Enterprise Modeling and Ontologies for Interoperability (EMOI-INTEROP'05), Co-located with CAiSE'05, volume 160 of CEUR Workshop Proceedings, Porto, Portugal, June 2005.

Berardi, D., Calvanese, D., De Giacomo, G., Hull, M., Mecella, M. (2005b). Automatic Composition of Web Services in Colombo. In Proceedings of the Thirteenth Italian Symposium on Advanced Database Systems, (SEBD), pages 8-15, Brixen-Bressanone (near Bozen-Bolzano), Italy, June 2005.

Berardi, D., Calvanese, D., De Giacomo, G., Hull, M., Mecella, M. (2005c). Automatic Service Composition Based on Behavioural Descriptions. *International Journal of Cooperative Information Systems*, vol. 14, no. 4, pp.333-376.

Berners-Lee, T., Hendler, J., Lassila, O. (2001). The Semantic Web. *Scientific American*, vol. 284, no. 5, pp. 28-37.

Beth, T., Borcherding, M. and Klein, B. (1994). Valuation of trust in open networks. Proceedings of the third European Symposium on Research in Computer Security, Brighton, United Kingdom, November 7-9 (pp. 1-18).

Billionniere, E., Greiman, D., Gosha, K. (2009). A comparison of social service selection techniques. Proceeding of the 8th IEEE international conference on dependable, autonomic and secure computing, Chengdu, China, December June 12-14 (pp. 260–265).

Birukou, A., Blanzieri, E., D'Andrea, V., Giorgini, P., Kokash, N., Modena, A. (2007). IC-service: a service-oriented approach to the development of recommendation systems. Proceedings of the 2007 ACM symposium on Applied computing (SAC'07), Seoul, Republic of Korea, March 11-15, ACM Press New York, NY, USA, pp. 1683-1688.

Blake, M., B., Nowlan, M., F. (2007). A Web Service Recommender System Using Enhanced Syntactical Matching. Proceeding of the International Conference on Web Services (ICWS 2007), Seoul, Republic of Korea, March 11-15 (pp.

575-582).

Booth, D., Haas, H., McCabe, F., NewCorner, E., Champion, M., Ferris, C., Orchard, D. (2004). W3C working group note - Web services architecture, February 2004. http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/.

Borzymek, P., Sydow, M., Wierzbicki, A. (2009). Enriching Trust Prediction Model in Social Network with User Rating Similarity. Proceedings of Computational Aspects of Social Networks, IEEE Computer Society, June 24 - 27 (pp. 40-47).

Bruijn, J., D., Bussler, C., Domingue, J., Fensel, D., Hepp, M., Keller, U., Kifer, könig-Ries, M., B., Kopecky, J., Lara, R., Lausen, H., Oren, E., Polleres, A., Roman, D., Scicluna, J., Stollber, M. (2005). Web Service Modeling Ontology (WSMO). http://www.w3.org/Submission/WSMO/#L4036, Jun 2005. [Online, accessed 04-April-2018].

Buyya, R., Broberg, J., Goscinski, A. (2010). Cloud Computing: Principles and Paradigms. John Wiley & Sons, Inc., Hoboken, N. J, 1st edition, 2010.

Canfora, G., Penta, M., D., Esposito, R., Villani, M., L. (2005), An approach for QoS-aware service composition based on genetic algorithms. Proceedings of the conference on Genetic and evolutionary computation, Washington, USA, June 25-29 (pp. 1069–1075).

Cappiello, C., Matera, M., Picozzi, M., Daniel, F., Fernandez, A. (2012): Quality-Aware Mashup Composition: Issues, Techniques and Tools. Proceedings of the Eighth International Conference on the Quality of Information and Communications Technolog, Lisbon, pp. 10-19.

Chen, H., Yu, T., Lin, K.J. (2003). QCWS: An implementation of QoS-capable multimedia Web services. Proceedings of the IEEE Fifth International Symposium, Taichung, Taiwan, December 10-12 (pp. 38-45).

Cheng, J., Li, C., Zhou, M., C., Zeng, Q., Yla-Jaaski, A. (2014). Automatic Composition of Semantic Web Services Based on Fuzzy Predicate Petri Nets. *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 680-689

Cheong, F., Corbitt, B. (2009). A Social Network Analysis of the Co-Authorship Network of the Pacific Asia Conference on Information Systems From 1993 To 2008. Proceedings of PACIS 2009, p. 23.

Chiu, D.K.W., Leung, H., F., Lam, K., M. (2009). On the making of service recommendations: An action theory based on utility, reputation, and risk attitude. *Expert Systems with Applications*, vol. 36, no. 2, pp. 3293-3301.

Chung, J., Rao, V., R. (2012). A General Consumer Preference Model for Experience Products: Application to Internet Recommendation Services. *Journal of Marketing Research*, Vol. 49, No. 3, pp. 289-305.

Curbera, F., Nagy, W., A., Weerawarana, S. (2001). Web services: Why and how.

Proceedings of the Workshop on Object-Oriented Web Services (OOPSLA 2001), Florida, USA, October 14-18 (pp.111–117).

Day, J., Deters, R. (2004). Selecting the best web service. Proceedings of the 2004 Conference of the Centre for Advanced Studies on Collaborative Research (CR 2004), Ontario, Canada, October 04-07 (pp. 293-307).

Dellarocas, C., Wood, C., A. (2008). The sound of silence in online feedback: Estimating trading risks in the presence of reporting bias. *Management Science*, vol. 54, no. 3, pp. 460-476.

Dieberger, A., Dourish, P., Höök, K., Resnick, P., Wexelblat, A. (2000). Social navigation: techniques for building more usable systems. *Interactions,* vol. 7, no. 6, pp. 36-45.

Eijk, P., V., Diaz, M. (1989). Formal Description Technique Lotos: Results of the Esprit Sedos Project. Elsevier Science Inc., New York, NY, USA, 1989.

El Saddik A. (2006). Performance measurements of Web services-based applications. *IEEE Transactions on Instrumentation and Measurement*, vol. 55, no. 5, pp. 1599-1605.

Erl, T. (2016). Service-Oriented Architecture: Analysis and design for services and microservices. Prentice Hall, PearsonPTR, 2nd edition, 2016.

Estivill-Castro, V., Ferrer-Mestres, J. (2013). Path-finding in Dynamic Environments with PDDL-Planners. Procceding of the 16th International Conference on Advanced Robotics (ICAR'13), Engeneering School-Universidad de la Republica (UdelaR), Montevideo, Uruguay, November 25-29 (pages 1-7).

Fauvet, M., -C., Duarte, H. , Dumas, M., Benatallah, B. (2005). Handling Transactional Properties in Web Service Composition. Proceeding of the 6th International Conference on Web Information Systems Engineering, New York, USA, November 20-22 (pp. 273–289).

Fielding, R., T. (2000). Architectural styles and the design of networkbased software architectures. PhD thesis, University of California, Irvine. https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf [Online, accessed Mai 18 2018]

Ferrara, A. (2004). Web Services: A Process Algebra Approach. Proceedings of the 2nd International Conference on Service-Oriented Computing (ICSOC '04), New York, NY, USA, November 15 - 18 (pp. 242-251).

Ferrein, A., Schiffer, S., Lakemeyer, G. (2009). Embedding Fuzzy Controllers in Golog. Proceeding of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'09), Jeju Island, Korea, Aug 20-14 (pp. 894-899).

Girard, J., Y. (1987). Linear Logic. *European Association for Theoretical Computer Science (EATCS)*, vol. 50, no. 1, pp. 1-102.

Golbeck, J. (2009). Trust and nuanced profile similarity in online social networks. *ACM Transactions on the Web (TWEB)*, vol. 3, no. 4, pp. 1–33.

Gürsel, A. and Sen, S. (2009). Producing timely recommendations from social networks through targeted search. Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems, Budapest, Hungary, May 10-15 (pp. 805–812).

Jaeger, M., C., Muhl, G. (2007). QoS-based Selection of Services: The Implementation of a Genetic Algorithm. Proceeding of KiVS (Kommunikation in Verteilten Systemen) in Workshop: Service-Oriented Architectures und Service Oriented Computing, Bern, Switzerland, 26 Febrary - 2 March (pp. 359–370).

Jaeger, M., C., Rojec-Goldmann, G., Muhl, G. (2004). QoS aggregation for Web service composition using workflow patterns. Proceeding of the Eighth IEEE International Enterprise Distributed Object Computing Conference (EDOC), Monterey, CA, USA, September 24-24, (pp. 149–159)

Jaeger, M., C., Rojec-Goldmann, G., Muhl, G. (2005). QoS aggregation in Web service compositions. Proceeding of the IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE '05), Hong Kong, China, March 29 – 1 April (pp. 181–185).

Jøsang, A., (2008). Online reputation systems for the health sector. *electronic Journal of Health Informatics*, vol. 3, no. 1., pp. 1–10

Jøsang, A., Ismail, R., Boyd, C. (2007). A survey of trust and reputation systems for online service provision. *Decision Support Systems*, vol. 43, no. 2, pp. 618–644.

Hamadi, R., Benatallah, B. (2003). A Petri Net-based Model for Web Service Composition. Proceedings of the 14th Australasian Database Conference, volume 17 of ADC '03, pages 191-200, Darlinghurst, Australia, 2003. Australian Computer Society, Inc., Adelaide, Australia.

Hashemian, S., V., Mavaddat, F. (2005). A Graph-Based Approach to Web Services Composition. Procceding of the IEEE/IPSJ International Symposium on Applications and the Internet (SAINT '05), Trento, Italy, Jan 11-15 pp. 183-189. IEEE Computer Society.

Herlocker, J., Konstan, J., Terveen, L., Riedl, J. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 5-53.

Huhns, M., N., Singh, M., P. (2005). Service-Oriented Computing: Key Concepts and Principles. *IEEE Internet Computing, vol. 9*, no.1, pp. 75-81.

Hull, R., Su, J. (2005). Tools for composite web services: a short overview. *SIGMOD Record*, vol. 34, no. 2, pp. 86-95.

Kalepu, S., Krishnaswamy, S., Loke, S., W. (2004). Reputation = f (user ranking, compliance, verity). Proceedings of the IEEE International Conference on Web Services (ICWS.2004), California, USA, July 6-9, Published by the IEEE Computer Society (pp. 200-207).

Kang, G., Liu, J., Tang, M., Liu, X., Cao, B., Xu, Y. (2012). AWSR: Active Web Service Recommendation Based on Usage History. Proceedings of the International Conference on Web Services (ICWS), Honolulu, HI, USA, June 24-29 (pp. 186–193).

Kavitha Devi, M., Venkatesh, P. (2010). Kernel based collaborative recommender system for e-purchasing. *Sadhana*, vol. 35, no. 5, pp. 513-524.

Kerrigan, M. (2006). Web service selection mechanisms in the web service execution environment (WSMX). Proceedings of the 2006 ACM symposium on Applied computing (SAC'06), Dijon, France, April 23-27 (pp. 1664-1668).

Khosravifar, B., Bentahar, J., Thiran, P., Moazin, A., Guiot, A. (2009). An Approach to incentive-based reputation for communities of Web services. Proceedings of the 7th International Conference on Web Services (ICWS), IEEE Computer Society, Los Angeles, California, USA, July 6-10(pp. 303–310).

Khoumsi, A. (2013). A Simple Formal Method to Synthesize an Orchestrator in Web Service Composition. Proceeding of the In American Control Conference (ACC), Washington, USA, June 17-19 (pp. 107-112)

Klein, T.,J., Lambertz, C., Spagnolo, G., Stahl, K., O. (2009). The actual structure of eBay's feedback mechanism and early evidence on the effects of recent changes. *International Journal of Electronic Business*, vol. 7, no. 3, pp. 301–320.

Kleinberg, J. (2008). The convergence of social and technological networks. *Communications of the ACM*, vol. 51, no. 11, pp. 66-72.

Kleinberg, J. (2008). The convergence of social and technological networks. *Communications of the ACM*, vol. 51, no. 11, pp. 66-72.

Kim, J., H., Lee, E., S. (2006). XQuery Pattern Method for Semantic Web based Personalization Recommender Service. *International Journal of Information Technology*, vol. 12, no. 6, pp. 87-95.

Kurshan, R., P. (1994). Computer-Aided Verification of Coordinating Processes: The Automata-Theoretic Approach. Princeton University Press, Princeton, NJ, USA, 1994.

Lee, D., Brusilovsky, P. (2009). Does Trust Influence Information Similarity? Proceedings of the ACM RecSys'09 Workshop on Recommender Systems & the Social Web, New York, USA, October 22-25 (pp. 71-74).

Lloyd, J., W., Shepherdson, J. C. (1991). Partial Evaluation in Logic Programming.

The Journal of Logic Programming, vol. 11, no. 3, pp. 217-242.

Li, S., Chen, H., Chen, X. (2010). A Mechanism for Web Service Selection and Recommendation Based on Multi-QoS Constraints. Proceeding of the 6th World Congress on Services (SERVICES-1) IEEE, Miami, Florida, USA, July 5-10 (pp. 221-228).

Li, Y., M., Kao, C.P. (2009). TREPPS: A Trust-based Recommender System for Peer Production Services. *Expert Systems with Applications*, vol. 36, no. 2, pp. 3263-3277.

Li, J., Yan, Y., Lemire, D. (2016). Full Solution Indexing for top-K Web Service Composition. *IEEE Transactions on Services Computing*, PP(99):1,13.

Limam, N., Boutaba, R. (2008). QoS and reputation-aware service selection. Proceeding of the IEEE/IFIP Network Operations and Management Symposium (NOMS), IEEE, Salvador, Brazil, April 7-11 (pp. 403-410).

Lin, K., J., Lu, H., Yu, T., Tai, C. (2005). A reputation and trust management broker framework for web applications. Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE '05), Hong Kong, China, March 29 – April 1 (pp. 262-269).

Liu, B., Yuan, Z. (2010). Incorporating social networks and user opinions for collaborative recommendation: local trust network based method. Proceedings of the Workshop on Context-Aware Movie Recommendation, Bacelona, Spain, September 30th, ACM, pp. 53-56.

Lo, W., Yin, J., Deng, S., Li, Y., Wu, Z. (2012). Collaborative web service qos prediction with location-based regularization. Proceedings of the 19th International Conference on Web Services (ICWS), Honolulu, HI, USA, June 24-29 (pp. 464–471).

Maamar, Z., Faci, N., Badr, Y., Wives, L.K., dos Santos, P.B., Benslimane, D., de Oliveira, J.P.M. (2011). Towards a framework for weaving social networks principles into web services discovery. Proceedings of the International Conference on Web Intelligence, Mining and Semantics, Sogndal, Norway, May 25-27 (pp. 51-59).

Maamar, Z., Lahkima, M., Benslimane, D., Thirand, P., Sattanathan, S. (2007). Web Services Communities: Concepts & Operations. Proceeding of the 3rd International Conference on Web Information Systems and Technologies, Bacelona, Spain, March 3-6 (pp. 3–6).

Maamar, Z., Wives, L.K., Badr, Y., Elnaffar, S. (2009). Even Web Services Can Socialize: A New Service-Oriented Social Networking Model. Proceedings of the International Conference on Intelligent Networking and Collaborative Systems (INCOS), Barcelona, Spain, November 8-9 (pp. 24–30).

Maaradji, A., Hacid, H., Daigremont, J., Crespi, N. (2010). Towards a Social Network Based Approach for Services Composition. Proceedings of the IEEE

International Conference on Communications, Cape Town, South Africa, May 23-27 (pp. 1-5).

Majithia, S., Walker, D., W., Gray, W., A. (2004). A Framework for Automated Service Composition in Service-Oriented Architectures. In Bussler C., Davies J., Fensel D., and Studer R., editors, The Semantic Web: Research and Applications, volume 3053 of Lecture Notes in Computer Science, pages 269-283. Springer Berlin Heidelberg, 2004.

Malik, Z., Akbar, I., Bouguettaya, A. (2009). Web Services Reputation Assessment Using a Hidden Markov Model. Proceedings of the 7th International Conference on Service Oriented Computing, Stockholm, Sweden, November 24-27 (pp. 576–591).

Manikrao, U., S., Prabhakar, T., V. (2005). Dynamic Selection of Web Services with Recommendation System. Proceeding of the international conference on next generation Web services practices, Seoul, Korea, August 22-26 (pp. 117-121).

Martin, D., Paolucci, M., Mcilraith, S., Burstein, M., Mcdermott, D., Mcguinness, D., Parsia, B., Payne, T., Sabou, M., Solanki, M., Srinivasan, N., Sycara, K. (2005). Bringing semantics to web services: The OWL-S approach. Proceedings of the 1st Work Shop on Semantic Web Services and Web Process Composition (SWSWPC), San Diego, California, USA, July 6. Revised selected papers, LNCS, Springer-verlag ,3387 (pp. 26–42).

Massa, P., Avesani, P. (2004). Trust-aware collaborative filtering for recommender systems. Proceedings of OTM Confederated International Conferences, CoopIS, DOA, and ODBASE, Agia Napa, Cyprus, October 31 - November 4 (pp. 492-508).

Massa, P., Avesani, P. (2007). Trust-aware recommender systems. Proceedings of the ACM conference on Recommender systems RecSys '07, Minneapolis, USA, October 19-20 (pp. 17–24).

Massa, P., Bhattacharjee, B. (2004). Using trust in recommender systems: an experimental analysis. Proceeding of the second international conference, iTrust, Trust Management, Oxford, UK, March 29 - April 1 (pp. 221-235).

Maximilien, E., M., Singh, M., P. (2005). Toward Web services interaction styles. Proceeding of the IEEE International Conference on Services Computing, Orlando, FL, USA, July 11-15 (pp. 147-154).

McIlraith, S., & Son, T. (2002). Adapting Golog for composition of semantic web services. Proceedings of the 8th International Conference on Knowledge Representation and Reasoning, Toulouse, France, April 22-25 (pp. 482–496).

Medjahed, B., Bouguettaya, A. (2005). A Dynamic Foundational Architecture for Semantic Web Services. *Distributed and Parallel Databases*, vol. 17, no. 2, pp. 179–206.

Menascé, D., A. (2002). Qos issues in Web services. *IEEE internet computing*, vol. 6, no. 6, pp. 72–75.

Metrouh, A., Seridi-Bouchelaghem, H., Mokhati, F. (2012). Web Services Discovery - A Novel Social Networks Approach based on Communities. Procedings of the 14th International Conference of the Enterprise Information Systems (ICEIS), Wroclaw, Poland, 28 June - 1 July (2, pp. 316–319). SciTeSeer, ISBN: 978-989-8565-11-2, doi: 10.5220/0004094103160319.

Metrouh, A., Mokhati, F. (2013). Social Web Services Discovery: A Community-Based Approach. Proceedings of the 15th International Conference on Information Integration and Web-based Applications & Services (iiWAS), Vienna, Austria, December 02-04 (pp. 275–279). ACM, ISBN: 978-1-4503-2113-6, doi: 10.1145/2539150.2539257.

Metrouh, A., Monfort V. (2013). Selecting Web services on the fly "According to Dynamic Social Communities Creation". Proceedings of the the 2nd International Workshop on Web Intelligence WEBI (ICEIS 2013), Angers, France, July 04-07 (pp. 82–89). SciTeSeer, ISBN: 978-989-8565-63-1, doi:10.5220/0004605500820089.

Metrouh, A., Mokhati, F. (2016). A Novel Social Networks Approach Based on QoS for Web Services Selection. Proceedings of the Mediterranean Conference on Pattern Recognition and Artificial Intelligence (MedPRAI), Tebessa, Algeria, November 22 - 23 (pp. 92–97). ACM, ISBN: 978-1-4503-4876-8. doi: 10.1145/3038884.3038899.

Metrouh, A., Mokhati, F. (2018). Flexible Web service integration: A novel personalised social approach. *Journal of Experimental & Theoretical Artificial Intelligence*. Taylor & Francis Group. Vol. 30, no. 3, pp. 441–456. doi: 10.1080/0952813X.2018.1430862.

Milner, R. (1999). Communicating and Mobile Systems - The Pi-Calculus. Cambridge University Press, 1999.

Milner, R. (1982). A Calculus of Communicating Systems. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1982.

Mislove, A., Marcon, M., Gummadi, K.P., Druschel, P., Bhattacharjee, B. (2007). Measurement and analysis of online social networks. Proceedings of the 5th ACM/USENIX Internet Measurement Conference, ACM, New York, NY, USA, San Diego, CA (pp. 29-42).

Moghaddam, S., Jamali, M., Ester, M., Habibi, J. (2009). FeedbackTrust: using feedback effects in trust-based recommendation systems. Proceedings of the third ACM conference on Recommender systems RecSys '09, ACM, pp. 269-272.

Narayanan, S., McIlraith, S., A. (2002). Simulation, Veri_cation and Automated Composition of Web Services. Proceedings of the 11th International Conference on World Wide Web (WWW '02), Honolulu, HI, USA, May 07-

11 (pp. 77-88).

Nau, D., Au, T., Ilghami, O., Kuter, U., Murdock, J., W., Wu, D., Yaman, F. (2003). SHOP2: An HTN Planning System. Journal of Artificial Intelligence Research, vol. 20, no. 1, pp. 379-404.

Newman, M. (2003). The structure and function of complex networks. SIAM Review, vol. 45, pp. 167–256.

Novet, J. (2015). Amazon Web Services posts $1.8B in revenue in Q2 2015, up 81% from last year. [En ligne] Disponible à: https://venturebeat.com/2015/07/23/amazon-web-services-posts-1-8b-in-revenue-in-q2-2015-up-81-from-last-year/ [20 Avril 2018].

Paik, I., Chen, W., Huhns, M., N. (2014). A Scalable Architecture for Automatic Service Composition. *IEEE Transactions on Services Computing*, vol. 7, no. 1, pp. 82-95.

Papazoglou, M., P. (2008). Web Services: Principles and Technology. Prentice Hall (1rst edition).

Papazoglou, M.,P., Traverso, P., Dustdar, S., Leymann, F. (2008). Service oriented computing: a research roadmap. *International Journal of Cooperative Information Systems*, vol. 17, no. 2, pp. 223-255.

Papazoglou, M., P., Traverso, P., Dustdar, S., Leymann, F. (2007). Service-oriented computing: State of the art and research challenges. *IEEE Computer*, vol. 40, no. 11, pp. 38-45.

Pathak, J., Basu, S., Lutz, R., R., Honavar, V. (2008). MOSCOE: An Approach for Composing Web Services through Iterative Reformulation of Functional Specications. *International Journal on Artificial Intelligence Tools*, vol. 17, no 1, pp. 109- 138.

Peer, J. (2005). A POP-Based Replanning Agent for Automatic Web Service Composition. Proceedings of the Second European Conference on The Semantic Web: Research and Applications (ESWC'05), Berlin, Heidelberg, pp. 47-61. Springer-Verlag.

Picozzi, P., Radolfi, M., Cappiello, C., Matera, M. (2010). Quality-based Recommendations for Mashup Composition. In: Daniel, F., Facca, F.M. (eds.) ICWE 2010. LNCS, vol. 6385, pp. 360–371. Springer, Heidelberg.

Preist, C. (2004). A conceptual architecture for Semantic Web Services. Proceedings of the International Semantic Web Conference (ISWC), Hiroshima, Japan, November 7-11 (pp. 395–409).

ProgrammableWeb (2014). Centre de recherche ProgrammableWeb - Croissance des API Web de 2005 à 2013. [En ligne] Disponible à: http://www.programmableweb.com/api-research [Accéder 18 mars 2018].

Pujol, J., M., Sangüesa, R., Delgado, J. (2002). Extracting reputation in multi agent

systems by means of social network topology. Proceedings of the first international joint conference on Autonomous agents and multiagent systems, Bologna, Italy, July 15–19 (pp. 467-474).

Rana, O., F., Stout, K. (2000). What is scalability in multi-agent systems? Proceedings of the fourth international conference on Autonomous agents (AGENTS '00), Barcelona, Spain, June 03–07 (pp. 56-63).

Rao, J., Dimitrov, D., Hofmann, P., Sadeh, N., M. (2006). A Mixed Initiative Approach to Semantic Web Service Discovery and Composition: SAP's Guided Procedures Framework. Proceedings of the IEEE International Conference on Web Services (ICWS'06), Chicago, Illinois, USA, September 18- 22 (pp. 401-410). IEEE Computer Society.

Rao, J., Kungas, P., Matskin, M. (2004). Logic-Based Web Services Composition: From Service Description to Process Model. Proceedings of the IEEE International Conference on Web Services (ICWS'04), pages 446-453, San Diego, California, USA, Jul 2004. IEEE Computer Society.

Rao, J., & Su, X. (2004). A survey of automated web service composition methods. Proceedings of the First International Conference on Semantic Web Services and Web Process Composition (SWSWPC'04), Springer-Verlag Berlin, Heidelberg, pp. 43–54. doi: http://dx.doi.org/10.1007/978-3-540-30581-1_5.

Resnick, P., Kuwabara, K., Zeckhauser, R., Friedman, E. (2000). Reputation systems. *Communications of the ACM*, vol. 43, no. 12, pp. 45-48.

Resnick, P., Varian, H., R. (1997). Recommender systems. *Communications of the ACM*, vol. 40, no. 3, pp. 56-58.

Richter, J., Chhetri, M., B., Kowalczyk, R., Vo, Q. B. (2012). Establishing composite SLAs through concurrent QoS negotiation with surplus redistribution. *Concurrency and Computation: Practice and Experience*, vol. 24, no. 9, pp. 938-955.

Ripon, S., Salah, M., U., and Barua, A. (2014). Web Service Composition - BPEL vs CCSP Process Algebra. Computing Research Repository (CoRR), abs/1402.5592, 2014.

Rosenberg, F., Platzer, C., Dustdar S. (2006). Bootstrapping performance and dependability attributes of Web services. Proceedings of the IEEE International Conference on Web Services (ICWS'06), Chicago, Illinois, USA September 18-20 (pp. 205–212).

Sabater, J., Sierra, C. (2002). Reputation and social network analysis in multiagent systems. Proceedings of the first international joint conference on Autonomous agents and multiagent systems (AAMAS '02), Bologna, Italy, July 15-19 (pp. 475-82).

Salaün, G., Bordeaux, L., Schaerf, M. (2004). Describing and Reasoning on Web Services Using Process Algebra. Proceedings of the IEEE International

Conference on Web Services, ICWS '04, San Diego, California, USA, Jun 6-7 (pp. 43-50).

Sarwar, B., Karypis, G., Konstan, J., Reidl, J. (2001). Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th international conference on World Wide Web, Hong Kong, China, May 01-05, ACM, (pp. 285-295).

Schafer, J., Frankowski, D., Herlocker, J., Sen, S. (2007). Collaborative filtering recommender systems. The Adaptive Web: methods and strategies of Web personalization, pp. 291-324.

Shao, L., Zhang, J., Wei, Y., Zhao, J., Xie, B., Mei, H. (2007). Personalized qos prediction for web services via collaborative filtering. Proceedings of International Conference on Web Services. Salt Lake City, UT, USA, Jully 09-13 (pp. 439–446).

Shekarpour, S., Katebi, S.D. (2010). Modeling and Evaluation of Trust with an Extension In Semantic Web. *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 8, no. 1, pp. 26–36.

Sherchan, W., Loke, S., W., Krishnaswamy, S. (2006). A fuzzy model for reasoning about reputation in Web services. Proceedings of the 2006 ACM Symposium on Applied Computing (SAC'06), Dijon, France, April 23-27 (pp. 1886-1892).

Skogan, D., Gronmo, R., Solheim, I. (2004). Web Service Composition in UML. Proceedings of the Eighth IEEE International Enterprise Distributed Object Computing Conference, EDOC '04, pages 47-57, Washington, DC, USA, 2004. IEEE Computer Society.

Singh, M., P., Huhns, M., N. (2005), Service-Oriented Computing: Semantics, Processes, Agents, John Wiley & Sons, Ltd, Chichester, West Sussex PO19 8SQ, England.

Sirin, E, Parsia, B., Wu, D., Hendler, J., Nau, D. (2003). HTN Planning for Web Service Composition using SHOP2. Web Semantics: Science, Services and Agents on the World Wide Web, 1(4):377-396, 2004. International Semantic Web Conference 2003.

Sohrabi, S., McIlraith, S., A. (2009). Optimizing Web Service Composition While Enforcing Regulations. Proceedings of the 8th International Semantic Web Conference (ISWC'09), October 25 - 29 (pp. 601-617). Springer-Verlag.

Staab, S., van der Aalst, W., Benjamins, V., R. (2003). Web services: been there, done that? *IEEE Intelligent Systems*, vol.18, no.1, pp. 72–85.

Stollberg, M., Lausen, H., Keller, U., Zaremba, M., Haller, A., Fensel, D., Kifer, M. (2004): D3.3 v0.1 wsmo use case virtual travel agency. Working Draft D3.3v0.1, WSMO. Available from http://www.wsmo.org/2004/d3/d3.3/v0.1/

Tang, M., Jiang, Y., Liu, J., Liu, X. (2012). Location-Aware Collaborative Filtering

for QoS-Based Service Recommendation. Proceedings of the International Conference on Web Services (ICWS), Honolulu, HI, USA, June 24-29 (pp. 202–209).

Vitvar, T., Mocan, A., Kerrigan, M., Zaremba, M., Moran, M., Cimpian, E., Haselwanter, T. and Fensel, D. (2007). Semantically-enabled service oriented architecture: concepts, technology and application. *Service Oriented Computing and Applications*, vol. 1, no. 2, pp. 129-154.

Vu, L., H., Hauswirth, M., Aberer, K. (2005). QoS-based service selection and ranking with trust and reputation management. Proceedings of OTM Confederated International Conferences, CoopIS, DOA, and ODBASE, Agia Napa, Cyprus, October 31 - November 4 (pp. 466-483).

Wang, P. (2009). QoS-aware Web services selection with intuitionistic fuzzy set under consumer's vague perception. *Expert Systems with Applications*, vol. 36, no. 3, pp. 4460-4466.

Wang, P., Ding, Z., Jiang, C., Zhou, M. (2014). Automated Web Service Composition Supporting Conditional Branch Structures. *Enterprise Information Systems*, vol. 8, no. 1, pp. 121-146.

Wang, S., G., Sun, Q., B., Yang, F., C. (2010). Towards Web Service selection based on QoS estimation. *International Journal of Web and Grid Services*, vol. 6, pp. 424–443.

Wang, Y., Vassileva, J. (2007). Toward Trust and Reputation Based Web Service Selection: A Survey. *International Transactions on Systems Science and Applications (ITSSA) Journal*, vol. 3, no. 2, pp. 118–132.

Wang, X., Vitvar, T., Kerrigan, M., Toma, I. (2006). A QoS-aware selection model for semantic Web services. Proceedings of the 4th International Conference on Service Oriented Computing (ICSOC), Springer-Verlag LNCS series, Chicago, USA, December 4-7 (pp. 390–401).

Wang, S., Zheng, Z., Wu, Z., Yang, F., Lyu, M., R. (2014a). Reputation Measurement and Malicious Feedback Rating Prevention in Web Service Recommendation Systems. *IEEE Transactions on Services Computing*, vol. 8, no. 5, pp. 755–767.

Wu, Z., Ranabahu, A., Gomadam, K., Sheth, A., P., Miller, J., A. (2007). Automatic Composition of Semantic Web Services using Process and Data Mediation. Technical report, Kno.e.sis Center, Wright State University, Dayton, Ohio, USA, Feb 2007.

Xia, B., Fan, Y., Tan, W., Huang, K.., Zhang, J., Wu, C. (2015). Category-aware API clustering and distributed recommendation for automatic mashup creation. *IEEE Transactions on Service Computing*, vol. 8, no. 5, pp. 674–687.

Xia, Y., Luo, X., Li, J., Zhu, Q. (2013). A Petri-Net-Based Approach to Reliability

Determination of Ontology-Based Service Compositions. *IEEE Transactions Systems, Man, and Cybernetics Systems*, vol. 43, no. 5, pp. 1240-1247.

Xiong, L., Liu, L. (2004). Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843-857.

Xu, W., Cao, J., Hu, L., Wang, J., Li, M. (2013). A Social-Aware Service Recommendation Approach for Mashup Creation. Proceedings of the 20th IEEE International Conference on Web Services, Santa Clara, CA, USA, June 28-July 3 (pp. 107-114).

Xu, P., Gao, J., Guo, H. (2005). Rating reputaion: a necessary consideration in reputation mechanism. Proceedings of 2005 International Conference on Machine Learning and Cybernetics, Guangzhou, China, August 18-21, vol. 1, IEEE, (pp. 182-187).

Xu, Y., Ma, J., Sun, Y., Hao, J. and Zhao, Y. (2009). Using social network analysis as a strategy for e-commerce recommendation. Proceeding of Pacific Asia Conference on Information Systems (PACIS 2009), Hyderabad, India, July 10-12 (p. 106).

Xu, Z., Martin, P., Powley, W., Zulkernine, F. (2007). Reputation-Enhanced QoS-based Web Services Discovery. Proceedings of the IEEE International Conference on Web Services (ICWS), Salt Lake City, Utah, USA, July 10-12 (p. 249–256).

Yang, J., Zhou, X., Wang, J., Zhu, X. (2014). A Novel Method for Web Service Composition Based on Extended BDI. Procceding of the IEEE 11th International Conference On Networking, Sensing and Control, (ICNSC'14), Miami, FL, USA, April 7-9 (pp. 310-315).

Yao, L., Sheng, Q. Z., Ngu, A. H. H., Yu, J., Segev, A. (2015): Unified Collaborative and Content-Based Web Service Recommendation. *IEEE Transactions on Services Computing*, vol. 8, no. 3, pp. 453-466.

Yao, L., Sheng, Q., Z., Segev, A., Yu, J. (2013). Recommending Web Services via Combining Collaborative Filtering with Content-based Features. Proceedings of International Conference on Web Services (ICWS). Santa Clara, CA, USA, 28 Jun - 03 Jul (pp. 42–49).

Yu, Q., Bouguettaya, A. (2012). Multi-attribute optimization in service selection. *World Wide Web*, vol. 15 no. 1, pp. 1-31.

Zhao, C., Ma, C., Zhang, J., Zhang, J., Yi, L., Mao, X. (2010). HyperService: Linking and Exploring Services on the Web. Proceedings of the IEEE International Conference on Web Services (ICWS), Miami, Florida, USA, July 5-10 (pp. 17–24).

Zheng, Z., Ma, H., Lyu, M., R., King, I. (2009). Wsrec: A collaborative filtering based web service recommender system. Proceedings of IEEE International

Conference on Web Services, Los Angeles, CA, USA, July 7-10, (pp. 437-444).

Zeng, L., Ngu, A., H., Benatallah, B., Podorozhny, R., Lei, H. (2008). Dynamic Composition and Optimization of Web Services. *Distributed and Parallel Databases*, vol. 24, no. 1, pp. 45-72.

Zibin, Z., Hao, Z., Michael R., L., Irwin, K.: WSRec (2009): A Collaborative Filtering Based Web Service Recommender System. Proceeding of the IEEE International Conference on Web Services, Los Angeles, CA, USA, July 6-10 (p. 437–444).

Ziegler, C., N., Golbeck, J. (2007). Investigating interactions of trust and interest similarity. *Decision Support Systems*, vol. 43, no. 2, pp. 460-475.