

On the Reputation of Communities of Web Services

Hamdi Yahyaoui¹, Zakaria Maamar², Jamal Bentahar³
Nabil Sahli⁴, Said Elnaffar⁵, and Philippe Thiran⁶

¹Sharjah University, Sharjah, UAE

²Zayed University, Dubai, UAE

³Concordia University, Montreal, Canada

⁴Telematica Instituut, Enschede, The Netherlands

⁵UAE University, Al-Ain, UAE

⁶University of Namur, Namur, Belgium

ABSTRACT

Web services communities are virtual clusters that agglomerate Web services with the same functionality. However, selecting the best community to deal with is challenging to both users and providers. Reputation has been widely used for evaluating and ranking candidates. In this paper, we introduce a reputation-based Web services community architecture and define some of the performance metrics that are needed to assess the reputation of a Web service community as perceived by the users and providers.

Categories and Subject Descriptors

J.4 [Computer Applications]: SOCIAL AND BEHAVIORAL SCIENCES

General Terms

Performance, Reliability

Keywords

Community, QoS, Reputation, Web Service, Agent

1. INTRODUCTION

La technologie du World Wide Web a énormément changé la manière avec laquelle nous accomplissons nos activités quotidiennes, allant de la vérification des marchés boursiers jusqu'à la réservation de chambres d'hôtel. Un nombre grandissant de personnes se connectent 24 heures par jour, 7 jours par semaine sur des sites Web à la recherche des dernières nouveautés et des meilleures opportunités. Toutefois, qu'est ce qui pousse ces personnes à revenir pour continuer d'utiliser presque les mêmes sites Web? La *réputation* est une des réponses clés à cette interrogation. Brièvement, on définit la réputation comme un processus continu qui est

fondé sur plusieurs facteurs comme la satisfaction, la fiabilité, l'efficacité, et les expériences antérieures.

Les services Web sont parmi les technologies prometteuses dans le développement de processus d'affaires inter-entreprises (souvent désignés par applications B2B). De nos jours, il existe une grande variété de services Web capables de répondre aux différents types de requêtes des utilisateurs. En plus, plusieurs organisations proposent pratiquement le même ensemble de services Web (e.g., prévision météo via le service du National Digital Forecast Database XMLWeb¹ et du National Weather Service Forecast Office²) en espérant que les utilisateurs choisissent leurs services Web et continuent à le faire dans le futur. Ce choix régulier est souvent lié à la réputation. Bui et Gacher notent que malgré les différences qui peuvent exister entre les services Web, leurs fonctionnalités sont suffisamment bien définies et assez homogènes pour permettre une compétition sur le marché [5]. Des services Web avec des fonctionnalités similaires constituent ce qui est souvent désigné par *communautés* [3, 9, 19]. En accord avec d'autres définitions [2, 12, 13], on définit une communauté comme un moyen de rassembler des services Web similaires indépendamment de leurs types de développement, emplacements, ou fonctionnements [9].

La réputation des services Web est un domaine de recherche très actif [7, 11, 17, 20, 21]. En raison du nombre croissant des communautés offertes en ligne, la compétition est devenue significative. Cependant, à date, peu d'efforts ont été consentis à classer les communautés des services Web selon un modèle de réputation approprié. Nous suggérons alors qu'un modèle de réputation propre aux communautés soit examiné aussi bien du point de vue fournisseur qu'utilisateur. En effet, d'une part, un fournisseur doit choisir la communauté dans laquelle il va inscrire son service Web. D'autre part, un utilisateur doit choisir, entre autres, avec quelle communauté interagir avant d'identifier le service Web (qui réside dans cette communauté) qu'il invoquera plus tard. L'objectif de notre projet de recherche est de définir un modèle de réputation pour les communautés des services Web, tout en mettant l'accent sur les éléments suivants : (i) comment structurer ce modèle, (ii) comment rendre ce modèle accessible, et (iii) comment garder ce modèle à jour. Il est à noter qu'une communauté de services Web est dynamique par nature : de nouveaux services Web

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

NOTERE 2008 June 23-27, 2008, Lyon, France

Copyright 2008 ACM 978-1-59593-937-1/08/0003 ...\$5.00.

¹<http://www.weather.gov/xml>.

²<http://www.srh.noaa.gov/mfl>.

s'ajoutent, d'autres services Web quittent, certains services Web deviennent temporairement indisponibles, d'autres services Web redevenant opérationnels après suspension, etc. Tous ces évènements doivent être surveillés de près pour éviter les conflits. Ces conflits se produisent par exemple lorsqu'un service Web quitte une communauté sans préavis. Les autres membres qui croient que ce service Web fait encore partie de leur communauté pourraient à tort lui attribuer des tâches.

Le reste de cet article est organisé comme suit. La Section 2 présente quelques définitions de base et donne un aperçu sur des travaux connexes. La Section 3 décrit l'architecture et le fonctionnement d'une communauté. Notre modèle de réputation pour les communautés de services Web est discuté dans la Section 4. Enfin, la Section 5 conclut l'article et décrit nos travaux futurs.

2. PRÉLIMINAIRES

2.1 Définitions

Réputation. C'est "*l'opinion (plus techniquement, une évaluation sociale) publiée envers une personne, un groupe de personnes, ou une organisation. C'est un facteur important dans plusieurs domaines tels que le monde des affaires, les communautés en ligne ou les statuts sociaux*" (traduction de l'encyclopédie en ligne Wikipédia). Toutefois, les opinions et les classements déterminés par les utilisateurs ne sont plus suffisants pour évaluer la réputation des systèmes informatiques comme Elnaffar le mentionne dans [6]. Les opinions et classements des utilisateurs sont en effet subjectifs et peuvent être facilement falsifiés. Un système de réputation qui compte seulement sur la perspective temporelle des humains (i.e., clients) peut exposer le système, volontairement ou involontairement, à des évaluations malhonnêtes.

Service Web. C'est "*une application logicielle identifiée par un URI dont les interfaces et l'enveloppe (binding) sont capables d'être définies, décrites, et retrouvées par des artefacts XML et qui supporte les interactions directes avec d'autres applications logicielles par l'utilisation de messages XML via des applications Internet*" (traduction de la définition du W3C). Un service Web est associé à une fonctionnalité (e.g., réservation dans un hôtel, validation d'une demande de prêt) qui correspond au type de "service" que ce service Web offre aux utilisateurs et aux autres services Web. La composition est l'un des points forts des services Web puisqu'elle permet le traitement des requêtes utilisateurs (qui ne peuvent être réalisées par un seul service Web) en combinant plusieurs services Web [4].

Communauté. C'est "*un groupe de personnes qui vivent ensemble et/ou sont unies par des intérêts communs, une religion, une nationalité, etc.*" (traduction du Dictionnaire Longman). Appliquée aux services Web, Benatallah et al. définissent une communauté comme un ensemble de services Web qui possèdent une même fonctionnalité même si ces services Web ont des propriétés non fonctionnelles distinctes [2]. Quant à Medjahed et Bouguettaya, ils considèrent une communauté comme un moyen d'offrir une ontologie des services Web ayant le même domaine d'intérêt [13]. Enfin, Maamar et al. définissent une communauté par la fonctionnalité d'un service Web représentatif de cette communauté, i.e., sans faire référence explicitement à quelconque service Web concret dans la communauté [9]. Dans ce travail, nous utilisons la dernière définition. Un exemple

d'utilisation de communauté est reporté dans [12] où Medjahed et Atif adoptent des communautés pour implanter des techniques, basées sur des règles, pour la comparaison des politiques contextuelles des services Web.

2.2 Quelques travaux sur la réputation des services Web

Dans [7], Jurca et Faltings affirment que la plupart des services Web ont besoin d'avoir des contrats capables de garantir un certain niveau de qualité de service (QoS). S'assurer du respect de ces contrats peut se faire par le biais d'un mécanisme de réputation. Ce dernier identifierait par exemple les fournisseurs qui ne font pas d'efforts pour maintenir la qualité de service de leurs services Web. La réputation est basée ici sur les commentaires rétroactifs des utilisateurs après élimination des commentaires malhonnêtes et conflictuels. D'autres travaux qui réclament la nécessité d'évaluer les déviations entre la qualité de service annoncée et effective, sont reportés dans [8] et [14]. Dans [11], Maximilien et Singh notent qu'un modèle de réputation pour les services Web peut être offert par des parties indépendantes et de confiance connues sous le nom d'agences ou intermédiaires (d'autres chercheurs utilisent le terme de gestionnaires de réputation). Le rôle d'une agence est de rassembler la bonne information sur la qualité d'un service Web et de présenter cette information dans un format approprié aux utilisateurs potentiels pendant la phase de sélection de services Web.

Un modèle de réputation facilite non seulement la sélection des meilleurs services Web mais permet aussi de distinguer les bons des mauvais services [20]. Dans un environnement aussi dynamique qu'Internet, il est peu probable que tous les services Web puissent être étiquetés d'un niveau de réputation. Les services Web apparaissent et disparaissent, reprennent leur activité après suspension, se renouvellent et sont mis à jour sans préavis. Un bon service Web peut être qualifié comme étant celui qui satisfait les besoins des utilisateurs, qui utilise des ressources comme convenu, qui ne révèle pas des données privées à un tiers, qui ne crée pas ses propres certificats de sécurité, et enfin qui n'essaie pas d'altérer ou faire mauvais usage des données ou des opérations de ses pairs. Étant donné ces critères, spécifier ce qu'est un mauvais service devient maintenant un simple exercice. Ce qui est important par contre, ce sont les mécanismes qui permettraient de gérer les mauvais services Web en les mettant par exemple en "quarantaine", en réduisant leur niveau de réputation, ou en les dénonçant à une autorité régulatrice.

Dans [16], Shaikh Ali et al. utilisent la réputation pour identifier des services Web dans une configuration en grille. En effet, il est supposé qu'un environnement riche en services Web verra le jour, ce qui nécessite la sélection entre des services Web similaires mais offerts par des fournisseurs indépendants. Toutefois, les technologies actuelles telles que UDDI n'aident pas à localiser les services Web selon leurs capacités et comportements. Les travaux de Shaikh Ali et al. permettent aux utilisateurs de trouver un service Web de bonne réputation qui répond à leurs besoins. Il permet également de créer et partager facilement des cursus d'activité complexes et de haute qualité. En appui aux remarques faites par Shaikh Ali et al. concernant l'inadéquation des technologies actuelles pour une sélection efficace des services Web, Xu et al. ajoutent que les détails

non fonctionnels des services Web sont contestables [21]. Pour ces fins, ils suggèrent un modèle de réputation qui combine un registre UDDI amélioré et un gestionnaire de réputation qui attribue des points aux services Web selon les commentaires rétroactifs des clients. Même si notre revue de littérature révèle que bon nombre de travaux sur les services Web et réputation ont été proposés, examiner les communautés du point de vue réputation fait encore défaut. Ceci permettrait par exemple d'identifier les meilleures communautés et d'analyser la réputation des services Web *versus* la réputation des communautés et comment l'une affecte l'autre.

3. COMMUNAUTÉS DES SERVICES WEB

Les travaux de recherche que nous avons entamés dans [9] et [19], n'incluent pas la réputation comme un critère ni pour structurer ni pour gérer des communautés. Par conséquent, les sections 3.1 et 3.2 présentent brièvement ces travaux. La formation et gestion des communautés en prenant en compte le concept de réputation sont décrites dans la section 4.

3.1 Architecture

La Figure 1 présente l'architecture que nous avons développée pour illustrer des communautés de services Web et la manière avec laquelle elles sont reliées aux fournisseurs des services Web et aux registres UDDI (ou tout autre type de registre tel que ebXML). Une communauté est assemblée et désassemblée selon certains protocoles que nous résumons dans la Section 3.2. De plus, une communauté est dynamique; les services Web peuvent entrer et quitter à leur convenance.

Dans la Figure 1, un service Web promu au grade de Maître (*Master*) dirige la communauté; les autres services Web de la communauté étant libelés maintenant comme Esclaves (*Slaves*). Les interactions entre Maître et Esclaves sont formulées selon des protocoles bien définis (section 3.2). Parmi les responsabilités du service Web Maître, nous identifions : attirer les services Web vers sa communauté en utilisant des récompenses (la réputation de la communauté n'est pas utilisée comme facteur d'attraction), convaincre les services Web de rester plus longtemps dans sa communauté, et nommer les services Web qui participeront aux scénarios de composition.

3.2 Fonctionnement

Le fonctionnement d'une communauté est divisé en trois parties : (1) gestion de la communauté, (2) attraction et rétention des services Web et (3) sélection des services Web pour la composition. Pour les besoins de cet article, nous mettons l'emphase juste sur les deux premières parties. La sélection des services Web en vue d'une composition est essentiellement basée sur le protocole Contract-net [18].

3.2.1 Gestion d'une communauté

Une communauté regroupe des services Web ayant la même fonctionnalité. Son assemblage est initiée par un concepteur selon les étapes suivantes. Initialement, le concepteur définit la fonctionnalité de la communauté (e.g., "réservation de vol"). Ensuite, il déploie le service Web Maître qui va diriger la communauté, ce qui veut dire entre autres, inviter des services Web à s'inscrire dans sa communauté et vérifier leurs certificats (e.g., QoS, mécanismes de protection) avant de les accepter de façon définitive. Il

est important de souligner que notre étude de la littérature révèle qu'il n'y a aucun travail qui prend en compte la réputation comme étant un facteur qui pourrait motiver les services Web à faire partie d'une certaine communauté.

Le désassemblage d'une communauté est aussi mené par le concepteur et se produit à la demande du service Web Maître. Ce dernier surveille tous les événements qui se déroulent dans sa communauté, comme par exemple l'arrivée de nouveaux services Web, le départ de services Web existants, l'identification des services Web qui feront partie d'une composition de services, et les sanctions envers les services Web dans le cas d'un mauvais comportement. Si le service Web Maître remarque que (i) le nombre de services Web dans la communauté est inférieur à un certain seuil et que (ii) le nombre de requêtes de participation dans des scénarios de compositions provenant des utilisateurs sur une certaine période de temps est aussi inférieur à un autre seuil prédéfini, alors la communauté sera désassemblée. Les deux seuils sont déterminés par le concepteur. Les services Web éjectés de cette communauté sont invités à rejoindre d'autres communautés. Comme l'on peut noter de cette description, la réputation de la communauté n'a pas été adoptée comme facteur pouvant motiver le désassemblage de la communauté. Une mauvaise réputation aurait pu être un argument valide pour désassembler la communauté, mais ceci n'est pas le cas ici.

3.2.2 Attraction et rétention des services Web

Attirer de nouveaux services Web et retenir ceux déjà existants dans la communauté relèvent des responsabilités du service Web Maître. Le service Web Maître consulte régulièrement les registres UDDI à la recherche de nouveaux services Web. Lorsqu'un service Web candidat est identifié par sa fonctionnalité, le service Web Maître interagit avec son fournisseur. Parmi les arguments qui sont utilisés durant ces interactions, le taux élevé de participation des services Web existants dans les compositions et l'efficacité des services Web dans le traitement des requêtes utilisateurs. Encore une fois, la réputation de la communauté n'a pas été exploitée pour attirer de nouveaux services Web.

Par ailleurs, retenir les services Web dans la communauté pour une longue période de temps est un bon indicateur des éléments suivants (pas de référence à la réputation non plus ici) : les services Web affichent une attitude³ coopérative bien qu'ils soient en compétition, et les services Web (fournisseurs) sont dans une certaine mesure satisfaits de leur taux de participation dans les compositions. L'attraction et la rétention des services Web mettent en évidence un troisième scénario concernant les services Web auquel une demande de quitter la communauté leur a été formulée. Un service Web Maître peut émettre une telle demande après avoir évalué les critères suivants : le service Web a une nouvelle fonctionnalité qui ne concorde pas tout à fait avec celle de la communauté, et le service Web est non fiable. Dans différentes occasions, ce service Web ne réussit pas à participer dans des compositions à cause de problèmes opérationnels récurrents ou d'une faible qualité de service.

³Voir [19] pour un exemple de "comportement coopératif" quand des services Web supportent les uns les autres lors d'un cas de substitution.

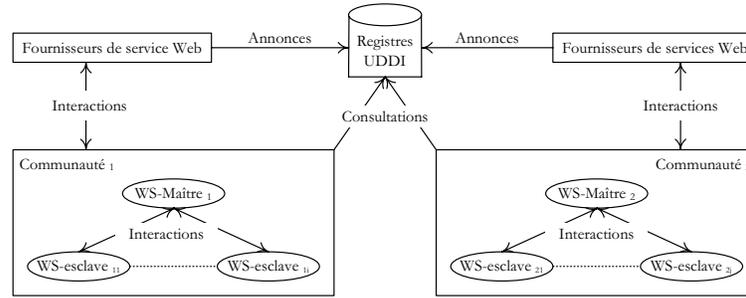


Figure 1: Architecture illustrant des communautés de services Web

4. RÉPUTATION DES COMMUNAUTÉS DE SERVICES WEB

4.1 Architecture

La Figure 2 décrit la nouvelle architecture de communautés de services Web qui cette fois-ci prend en compte la notion de réputation. Comme illustré, l'architecture a trois composantes principales :

- **UDDI Élargi.** Un registre UDDI traditionnel ne contient que des entrées qui décrivent des services Web individuels. Nous suggérons l'extension de ce registre en inscrivant des entrées additionnelles en relation avec les communautés.
- **Agent Client.** C'est un proxy entre le client et les autres parties, à savoir le registre UDDI élargi, les communautés des services Web et le système de réputation.
- **Agent Fournisseur.** Semblable à l'agent client, l'agent fournisseur est un proxy entre le fournisseur, les autres parties, à savoir le registre UDDI élargi, les communautés des services Web et le système de réputation.
- **Système de Réputation.** Il est au coeur de l'architecture et assure deux fonctions principales : (i) maintenir un répertoire de données opérationnelles et temps réel, désignées par *Logs* qui sont nécessaires pour calculer les métriques de performance de la communauté, et (ii) classer les communautés selon leur réputation en utilisant un algorithme de classement qui est en cours de développement (Section 4.2).

Développer des clients et des fournisseurs comme des parties indépendantes est indispensable pour intercepter en temps réel des données fiables sur chaque interaction avec les services Web. Afin de pouvoir calculer les *métriques de performance* nécessaires à l'évaluation de la réputation d'une communauté, du point de vue des clients et des fournisseurs, les agents clients doivent mettre à jour les *Logs* horodatés pour chaque événement qui survient durant l'interaction avec la communauté. Ces *Logs* sont :

1. *Log des Soumissions:* traque toutes les requêtes envoyées à la communauté, qui ne sont pas nécessairement suivies par une invocation d'un service Web. Comme convention, $submissions_{C_i}$ désignera

le nombre des requêtes envoyées à la communauté C_i tandis que $submissions_{C_i}^{WS_j}$, désignera le nombre de celles envoyées à la communauté C_i et déléguées par la suite au service Web WS_j .

2. *Log des Temps de Réponse :* traque le temps que met le service Web Maître d'une communauté donnée pour identifier un service Web Esclave afin que celui-ci prenne en charge la requête d'un client.
3. *Log des Invocations :* traque toutes les exécutions des requêtes envoyées au service Web qui a été sélectionné par la communauté.
4. *Log des Succès :* traque les services Web dont l'exécution a réussi.
5. *Log des Échecs :* traque les services Web dont l'exécution a échoué.
6. *Log des Recouvrements :* traque les services Web qui ont fait l'objet de ré-exécution après échec.
7. *Log des Frais :* traque les frais chargés par les fournisseurs des services Web.
8. *Log des Pos :* traque les votes positifs que les clients des services Web soumettent après interaction avec les communautés.
9. *Log des Neg :* traque les votes négatifs que les clients des services Web soumettent après interaction avec les communautés.

En plus de ces *Logs*, et dans le but de calculer les métriques de performance telle que perçue par le fournisseur, les agents fournisseurs doivent aussi capturer les *Logs* horodatés suivants :

1. *Log des Adhésions :* traque les fournisseurs qui ont inscrit leurs services Web dans la communauté.
2. *Log des Désinscriptions :* traque les fournisseurs qui ont retiré leurs services Web de la communauté.
3. *Log des Réactivations :* traque les service Web réactivés après une certaine suspension imposée par le service Web Maître de la communauté.
4. *Log des Suspensions :* traque les services Web dont l'adhésion a été suspendue par le service Web Maître.

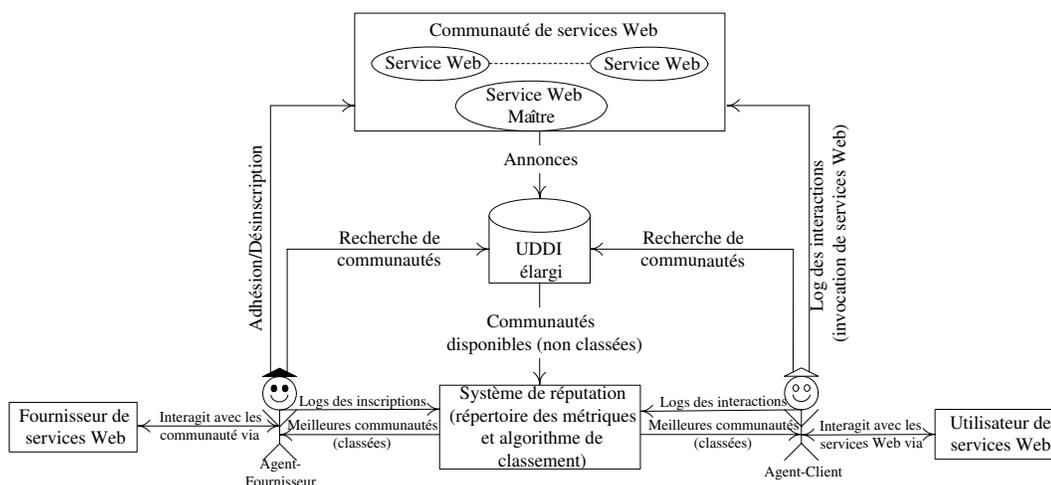


Figure 2: Architecture des communautés de services Web avec réputation

5. *Log des Pos* : traque les votes positifs que les fournisseurs des services Web soumettent après interaction avec les communautés.
6. *Log des Neg* : traque les votes négatifs que les fournisseurs des services Web soumettent après interaction avec les communautés.

La Section 4.2 décrit comment ces *Logs* sont utilisés pour calculer les métriques de performance nécessaires à l'évaluation de la réputation d'une communauté. Les clients, les fournisseurs, et les Maîtres des communautés ne peuvent pas altérer le contenu de ces *Logs* car tout est surveillé et collecté par des agents neutres. Nous supposons également que ce contenu soit enregistré dans des registres à accès publique.

Une communauté de services Web est principalement représentée par son Maître qui joue le rôle de porte d'accès au monde extérieur. Plus précisément, le Maître a pour tâche d'annoncer l'existence de sa communauté auprès de plusieurs registres UDDI élargis. De plus, il est responsable de gérer les adhésions et les désinscriptions que les agents fournisseurs réalisent. Sa tâche principale consiste cependant à gérer les requêtes des services Web qui sont relayées par les agents clients. Du point de vue des clients, ceux-ci espèrent être servis par des communautés qui fournissent une fiable et haute qualité de service. Lorsqu'un client (une application) a besoin d'un service Web, il délègue sa requête à un agent qui prend en charge la recherche de la communauté capable de fournir les fonctionnalités désirées (e.g., calcul prix action) et la qualité de service attendue (e.g., minimum de temps de réponse). L'agent cherche dans l'UDDI élargi la liste (pas encore triée) de toutes les communautés de services Web possibles qui satisfont les exigences du client. Ensuite, pour filtrer les meilleurs communautés possibles, la liste non triée est envoyée au système de réputation qui se charge de classer les communautés selon leur réputation en utilisant l'algorithme de classement. La liste des communautés les plus réputées est finalement retournée au client via son agent. Quant aux fournisseurs, ils se préoccupent surtout de s'engager avec des communautés réputées en espérant ainsi améliorer leur visibilité via-à-vis des utilisateurs. Pour

y parvenir, les agents fournisseurs sont toujours à l'affût des communautés compatibles avec les fonctionnalités des fournisseurs (e.g., **Réservation de vols**). Chaque agent cherche dans l'UDDI élargi et passe la liste des communautés sélectionnées au système de réputation. Ce dernier classe les communautés selon leur réputation. Enfin, la liste triée des communautés est renvoyée au fournisseur à travers son agent afin d'en choisir les meilleures auxquelles il peut adhérer. Notons que la définition de réputation telle que perçue par les clients est différente de celle perçue par les fournisseurs. La prochaine section détaille cette différence.

4.2 Modèle de réputation pour les communautés

Avant d'expliquer notre modèle de réputation pour les communautés de services Web, il est nécessaire de faire les deux commentaires suivants :

1. Les recherches sur la réputation ont juste mis l'emphase sur les services Web (Section 2.2).
2. Actuellement, dans le domaine des services Web, l'association (fonctionnalité/communauté) est de type un à un (Figure 1).

La réputation signifie de manière indirecte l'existence de différentes options (ou choix) parmi lesquelles une (parfois plusieurs) est sélectionnée. Par conséquent, le développement d'un modèle de réputation pour les communautés de services Web nécessite la révision du deuxième commentaire en considérant plutôt des associations (fonctionnalité/communauté) de type un à plusieurs, i.e., plusieurs communautés indépendantes ayant des fonctionnalités communes. Dans ce qui suit, C_i désigne la communauté candidate i qui est à considérer par un fournisseur ou utilisateur, alors $|C_i|$ désigne la taille de la communauté qui est en fait le nombre de ses services Web membres à un instant donné.

4.2.1 Réputation vue par l'utilisateur

Un modèle de réputation doit aider les utilisateurs à (i) choisir les communautés appropriées selon leurs besoins (objectif de ce travail) et (ii) les services Web appropriés au sein de ces communautés (plusieurs travaux se sont déjà intéressés à la sélection des services Web comme [1, 10, 22, 15]). Pour chaque service Web requis, un utilisateur aura à identifier une communauté. Le modèle de réputation des communautés va mener au processus de découverte des communautés et plus tard aux services Web en utilisant un ensemble de métriques.

En général, la plupart de ces métriques sont calculées sur la base de la moyenne des dernières observations enregistrées dans un certain $\mathcal{L}og$. Nous appelons ce dernier ensemble de mesures la *fenêtre d'observations* de taille w . Les métriques que nous considérons dans notre modèle sont :

1. **en Demande (*inDemand*)** : cette métrique évalue le nombre d'utilisateurs qui sont intéressés par une certaine communauté plutôt que d'autres. Elle est mesurée par le pourcentage de requêtes qu'une communauté reçoit par rapport au total des requêtes soumises à toutes les communautés. Elle est exprimée par :

$$inDemand_{C_i}^w = \frac{submissions_{C_i}^w}{\sum_{k=1}^M submissions_{C_k}^w} \quad (1)$$

où M désigne le nombre de communautés considérées. Par conséquent, la métrique *en Demande* reflète à quel point les utilisateurs ont tendance à favoriser une communauté particulière pour soumettre leurs requêtes.

2. **Temps de réaction (*responsiveness*)** : c'est l'une des plus importantes qualités que les utilisateurs recherchent dans une communauté. Elle est la métrique qui détermine la rapidité avec laquelle un Maître d'une communauté peut nommer un service Web membre capable de traiter la requête utilisateur. Le temps de réaction d'une communauté C_i , désigné par $responsiveness_{C_i}^w$, est mesuré en calculant la moyenne des temps de réponse aux requêtes émises dans la fenêtre w . Ainsi,

$$responsiveness_{C_i}^w = \frac{1}{n} \sum_{k=1}^n rt_k^w \quad (2)$$

où rt_k^w est le temps de réponse mis pour choisir un service Web qui répondrait à la requête de l'utilisateur numéro k , et n est le nombre de requêtes observées dans la fenêtre w .

3. **Satisfaction** : représente l'opinion subjective des clients qui ont récemment fait affaire avec une communauté sur une fenêtre d'observation w :

$$satisfaction_{C_i}^w = \frac{pos_{C_i}^w}{pos_{C_i}^w + neg_{C_i}^w} \quad (3)$$

où $pos_{C_i}^w$ et $neg_{C_i}^w$ représentent le nombre de votes positifs et négatifs, respectivement, que la communauté C_i a eu de la part de tous les clients dans la fenêtre w .

4.2.2 Réputation vue par le fournisseur

Un modèle de réputation doit aider les fournisseurs de services à identifier les communautés où leurs services Web peuvent s'inscrire. À partir de maintenant nous adoptons un cadre où un service Web est membre d'au plus une communauté⁴. Dans ce qui suit nous énumérons les métriques de performance les plus importantes du point de vue fournisseur, ce qui constitue la réputation de la communauté :

1. **Sélectivité (*selectivity*)** : la sélectivité est le nombre moyen de requêtes qui sont déléguées aux services Web membres d'une communauté. Tout fournisseur espère se joindre à la communauté ayant le maximum de sélectivité en comparaison avec les autres communautés. En supposant que le Maître de la communauté adopte une politique de sélection juste (uniforme) *vis-à-vis* des services Web, la sélectivité d'une communauté C_i est calculée comme suit :

$$selectivity_{C_i}^w = \frac{submissions_{C_i}^w}{|C_i^w|} \quad (4)$$

où $submissions_{C_i}^w$ est le nombre de requêtes de services Web reçues par la communauté C_i dans la fenêtre w , et $|C_i^w|$ est la taille de C_i en termes de nombre de services Web membres dans la même fenêtre.

2. **Market Share** : est la proportion des services Web inscrits à la communauté dans la fenêtre w par rapport au nombre total de services Web inscrits sur toutes les communautés dans la même fenêtre:

$$marketShare_{C_i}^w = \frac{|C_i^w|}{\sum_{k=1}^m |C_k^w|} \quad (5)$$

où $|C_i^w|$ représente le nombre de services Web qui se sont inscrits à la communauté candidate C_i pendant la période w , et $\sum_{k=1}^m |C_k^w|$ représente le nombre total des services Web des inscriptions sur les m fournisseurs dans la même période. Du point de vue fournisseur, le facteur $marketShare_{C_i}^w$ peut être une épée à double tranchant. En effet, plus $marketShare_{C_i}^w$ est grand plus la communauté candidate est forte, ce qui signifie une meilleure chance d'être sélectionné par les clients. Ceci est donc positif. Toutefois, avoir un $marketShare_{C_i}^w$ plus grand signifie aussi une compétition plus féroce pour le fournisseur, ce qui veut dire moins de chance d'être sélectionné quelque soit la politique de sélection (juste ou non) adoptée par le Maître de la communauté.

3. **Équitabilité (*Fairness*)** : les fournisseurs cherchent des communautés où ils seront traités de façon équitable et juste. Dans ce contexte, la plus simple définition d'équitabilité serait : donner presque la même chance de participation à tous les fournisseurs au sein d'une communauté. Ceci peut être évalué en calculant la fréquence de participation de chaque fournisseur. Une grande disparité indiquerait que certains fournisseurs sont favorisés tandis qu'une faible disparité refléterait une politique de sélection équitable

⁴La contrainte "au plus" peut être assouplie en permettant aux services Web de s'inscrire à plusieurs communautés en même temps.

adoptée par le Maître de la communauté. Un fournisseur potentiel favoriserait sûrement une communauté à faible disparité. Un outil statistique typique pour mesurer la disparité est la déviation standard que nous exprimons comme suit :

$$\sigma_{C_i}^w = \sqrt{\frac{1}{|C_i^w|} \sum_{j=1}^{|C_i^w|} (rf_i^{WS_{ij}^w} - \mu_{C_i}^w)^2} \quad (6)$$

où $rf_i^{WS_{ij}^w}$ est la fréquence d'utilisation du service Web WS_{ij} pendant la période w au sein de la communauté C_i , i.e.,

$$rf_i^{WS_{ij}^w} = \frac{\text{submissions}_{C_i}^{WS_{ij}^w}}{\text{submissions}_{C_i}^w} \quad (7)$$

où $\text{submissions}_{C_i}^{WS_{ij}^w}$ désigne le nombre de requêtes envoyées à la communauté C_i qui sont déléguées au service Web WS_{ij} ; et $\mu_{C_i}^w$ est la moyenne de toutes les fréquences relativement à la période w :

$$\mu_{C_i}^w = \frac{\sum_{j=1}^{|C_i^w|} rf_i^{WS_{ij}^w}}{|C_i^w|} \quad (8)$$

σ_{C_i} peut ainsi indiquer au fournisseur concerné à quel point le Maître de la communauté est consistant lorsqu'il s'agit de sélectionner uniformément les services Web membres. Un plus grand σ_{C_i} signifie que le Maître de la communauté favorise certains services Web au dépend des autres. L'équitabilité de la communauté C_i peut donc être définie comme étant l'inverse de σ_{C_i} :

$$\text{fairness}_{C_i}^w = \begin{cases} \infty & \text{si } \sigma_{C_i}^w = 0 \\ \frac{1}{\sigma_{C_i}^w} & \text{si } \sigma_{C_i}^w \neq 0 \end{cases} \quad (9)$$

5. CONCLUSION

Dans cet article, nous avons présenté les grandes lignes de notre projet de recherche sur la réputation des communautés de services Web. Les communautés ayant une bonne réputation sont convoitées aussi bien par les utilisateurs que par les fournisseurs. Les utilisateurs espèrent avoir une bonne qualité de service tandis que les fournisseurs espèrent augmenter leur visibilité et leur portée en vue d'un meilleur revenu (lucratif). Les différentes perspectives de la réputation exigent différentes méthodes de calcul de la réputation d'une communauté. Chacune de ces méthodes nécessite différents ensembles de métriques de performance. Dans cette première étude, nous avons proposé une architecture basée sur la réputation pour les communautés de services Web. L'UDDI élargi, les agents clients et fournisseurs et le système de réputation sont les composantes clés de cette architecture. L'UDDI élargi contient des entrées sur les communautés en plus des traditionnelles entrées sur les services Web. Les agents, quant à eux, jouent un rôle important en tant que parties indépendantes qui intercèpent les Logs nécessaires pour calculer les métriques de performance. Enfin, le système de réputation représente le coeur de l'architecture. Il maintient les données (Logs) requises pour calculer la réputation d'une communauté et exécute

l'algorithme de classement. Actuellement, nous travaillons sur la conception et l'implantation de cet algorithme qui transforme les métriques de performance d'une communauté en une seule valeur représentative de sa réputation.

6. REFERENCES

- [1] S. Agarwal and B. Sprick. Specification of Access Control and Certification Policies for Semantic Web Services. In *Proceedings of The 6th International Conference on Electronic Commerce and Web Technologies (EC-Web'2005)*, Copenhagen, Denmark, 2005.
- [2] B. Benatallah, Q. Z. Sheng, and M. Dumas. The Self-Serv Environment for Web Services Composition. *IEEE Internet Computing*, 7(1), January/February 2003.
- [3] J. Bentahar, Z. Maamar, D. Benslimane, and P. Thiran. Using Argumentative Agents to Manage Communities of Web Services. In *Proceedings of WAMIS'2007*, Niagara Falls, Ontario, Canada, 2007.
- [4] D. Berardi, D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Mecella. A Foundational Vision for e-Services. In *Proceedings of Ubiquitous Mobile Information and Collaboration Systems Workshop (UMICS'2003) held in conjunction with The 15th International Conference On Advanced Information Systems Engineering (CAiSE'2003)*, Klagenfurt/Velden, Austria, 2003.
- [5] T. Bui and A. Gacher. Web Services for Negotiation and Bargaining in Electronic Markets: Design Requirements and Implementation Framework. In *Proceedings of HICSS'2005*, Big Island, Hawaii, USA, 2005.
- [6] S. Elnaffar. Beyond User Ranking: Expanding the Definition of Reputation in Grid Computing. In *Proceedings of CIS²E'2006*, 2006.
- [7] R. Jurca and B. Faltings. Reputation-Based Service Level Agreements for Web Services. In *Proceedings of ICSOC'2005*, Amsterdam, The Netherlands, 2005.
- [8] Z. Maamar, S. Kouadri Mostéfaoui, and Q. H. Mahmoud. On Personalizing Web Services Using Context. *International Journal of E-Business Research, Special Issue on E-Services, The Idea Group Inc.*, 1(3), July-September 2005.
- [9] Z. Maamar, M. Lahkim, D. Benslimane, P. Thiran, and S. Sattanathan. Web Services Communities - Concepts & Operations -. In *Proceedings of WEBIST'2007*, Barcelona, Spain, 2007.
- [10] Z. Maamar, Q. Z. Sheng, and B. Benatallah. Selection of Web Services for Composition Using Location of Provider Hosts Criterion. In *Proceedings of Ubiquitous Mobile Information and Collaboration Systems Workshop (UMICS'2003) held in conjunction with The 15th International Conference On Advanced Information Systems Engineering (CAiSE'2003)*, Velden, Austria, 2003.
- [11] M. Maximilien and M. Singh. An Ontology for Web service Ratings and Reputations. In *Proceedings of OAS'2003*, Melbourne, Australia, 2003.
- [12] B. Medjahed and Y. Atif. Context-based Matching for Web Service Composition. *Distributed and Parallel Databases, Springer*, 21(1), January 2007.

- [13] B. Medjahed and A. Bouguettaya. A Dynamic Foundational Architecture for Semantic Web Services. *Distributed and Parallel Databases*, 17(2), March 2005.
- [14] M. Ouzzani and A. Bouguettaya. Efficient Access to Web Services. *IEEE Internet Computing*, 8(2), March/April 2004.
- [15] M. Paolucci, T. Kawamura, T. R. Payne, and K. Sycara. Importing the Semantic Web in UDDI. In *Proceedings of The Workshop on Web Services, E-business, and Semantic Web (WES'2002) held in conjunction with the 14th International Conference on Advanced Information Systems (CAiSE'2002)*, Toronto, Canada, 2002.
- [16] A. Shaikh Ali, S. Majithia, O. Rana, and D. Walker. Reputation-based Semantic Service Discovery. *Concurrency and Computation: Practice and Experience, John Wiley & Sons*, 18(8), 2006.
- [17] W. Sherchan, S. W. Loke, and S. Krishnaswamy. A Fuzzy Model for Reasoning about Reputation in Web Services. In *Proceedings of the 2006 ACM Symposium on Applied Computing (SAC'2006)*, Dijon, France, 2006.
- [18] R. Smith. The contract Net Protocol: High Level Communication and Control in Distributed Problem Solver. *IEEE Transactions on Computers*, 29, 1980.
- [19] Y. Taher, D. Benslimane, M.-C. Fauvet, and Z. Maamar. Towards an Approach for Web Services Substitution. In *Proceedings of IDEAS'2006*, Delhi, India, 2006.
- [20] Y. Wang and J. Vassileva. A Review on Trust and Reputation for Web Service Selection. In *Proceedings of ICDCSW'07*, Toronto, Ontario, Canada, 2007.
- [21] Z. Xu, P. Martin, W. Powley, and F. Zulkernine. Reputation-Enhanced QoS-based Web Services Discovery. In *Proceedings of ICWS'2007*, Salt Lake City, Utah, USA, 2007.
- [22] L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, and Q. Z. Sheng. Quality Driven Web Services Composition. In *Proceedings of The 12th International World Wide Web Conference (WWW'2003)*, Budapest, Hungary, 2003.