

TrustWS: A Trust Management System for Web Services

Loubna Mekouar
University of Waterloo
Waterloo, Canada
lmekouar@bcr.uwaterloo.ca

Youssef Iraqi
Khalifa University
Sharjah, UAE
Youssef.Iraqi@kustar.ac.ae

Abstract

In this paper, we propose a trust management system for Web services. The proposed system helps service consumers select the appropriate Web services based on the feedbacks gathered from past transactions. We describe the main components involved in collecting the feedbacks and managing the trust data. Simulation results confirm that the proposed trust management system increases users' satisfaction.

1 Introduction

A Web service (WS) is a self-describing software application that can be advertised, located and used across the Web using a set of standards (WSDL, UDDI and SOAP) [5]. Service consumers are facing the challenge to select the most appropriate Web services among those offering the same functionality. They also need to know in advance the expected service that they will receive.

Reputation is needed since the open nature of Web services opens the door to some Web services to act maliciously. These WS may fail to fulfill their service agreements by providing low quality, time-consuming, unreliable and/or expensive services. Reputation systems allow to capture the maliciousness of WS and inform the service consumers how reliable is a Web service in providing a specific functionality.

In Web services environments, the followings are the main players [5]:

- **Web service:** a Web service is a software application identified by a URI (Uniform Resource Identifier). A Web service represents a set of operations.
- **Service Providers:** the entity that provides the functionality. This entity may be a government, a business, a non profit organization, etc. A service provider (SP) may provide different Web services. A SP may be trusted in a specific context and not trusted in another

context. In this paper, we focus on the reputation of the Web services and not that of the Service Providers. Their reputation can be concluded from the Web services they are responsible for.

- **Service Consumers¹:** are the entities that invoke the Web service. They consume the service.

While e-commerce, peer-to-peer, and multi-agents systems have received tremendous attention from researchers in addressing the trustworthiness of users, peers, and agents respectively, only few research works have addressed the trust issue in Web services [1, 4, 6]. In Web Services, there is a need for trust management systems to help users distinguish among Web services and hence, select the appropriate ones to fulfill their requests. Recent research papers on Web services emphasize on the importance of incorporating trust in the WS environments. However, most works address trust only from the context of security like authentication, authorization, encryption, integrity and non-repudiation [2].

In this paper, we propose to address the reputation of a WS in terms of fulfillment of the required service as advertised. Several QoS metrics can be used to capture the behavior of the WS and evaluate the received functionality. The quality of a Web service refers to its performance (e.g., processing time), dependability (e.g., availability, reliability), security (e.g., authentication, confidentiality) and other application-specific metrics (e.g., metrics related to a specific domain) [8].

The paper is organized as follows. Section 2 presents the motivation behind this work. Section 3 describes the proposed trust system and the different components involved in managing the trust data. Section 4 shows the required steps to compute the Web service's reputation while Section 5 deals with Web services communities. Section 6 describes the performance evaluation conducted and presents the results that confirm the good performance of the proposed trust management system. Finally, Section 7 concludes the paper.

¹Throughout the paper, we use the terms users and service consumers interchangeably

2 Motivation

In service-oriented environments, Chang *et al.* [3], define trust as “*The belief the trusting agent has in the trusted agent’s willingness and capability to deliver a mutually agreed service in a given context and in a given time slot*”.

Trust management is of paramount importance to reduce the risk involved in transactions between Web services and service consumers.

For a Web service, we propose to address the reputation of a WS in terms of fulfillment of the required service as agreed (e.g., according to a Service Level Agreement (SLA)). We adapt our trust management framework [6] that was initially designed for P2P systems to the context of Web services.

Incorporating trust in Web Services environments will help providing the following advantages:

- Reduce the risk involved in the transactions
- Boost performance and improve efficiency
- Increase users’ satisfaction
- Help in the management of WS communities (i.e. a set of Web services that share the same functionality) as explained in Section 5.

3 The proposed Trust Management: TrustWS

The survey of different reputation systems reveals the important mechanisms used to achieve good reputation management [7]. In this section, we will describe the components involved in the reputation management process.

3.1 Submitting the Feedback

As explained in Section 1, several metrics can be used to capture the behavior of a WS. We propose to summarize all these QoS aspects in one binary value feedback. We assume that users can assess if the delivered quality of service of a WS is as agreed or not. A value of 1 means that the user is satisfied from the transaction and 0 expresses the disappointment of the user from the received service.

In a composite service, several services are combined to achieve an expected service. As an example, planning a trip requires the flight booking, the hotel booking, the sightseeing in addition to other features to satisfy the users needs. Combining different services is still considered as one service provided by this WS and will be rated by users as explained earlier.

3.2 Trust Manager

We gather the feedbacks from users. A trust management service is used to handle and manage all the trust data, called *Trust Manager*. A user may send a reputation query to the *Trust Manager* regarding potential Web services. Dealing with highly reputable Web services will increase the users satisfaction.

To collect the feedbacks, two alternatives are possible:

- Getting the feedbacks from all the users that invoked a specific Web service.
- Choosing only a random selection of users from all these users.

While the first alternative incurs additional overhead, the second alternative will reduce the generated overhead. However, the more information is gathered, the more accurate is the reputation value of a Web service.

The *Trust Manager* is used as a feedback collector rather than a monitor of the services provided to users. Monitoring will increase the burden of this entity by checking on a regular basis the performance of the Web services. Moreover, this approach will not reflect the experience as perceived by the users.

3.3 The Impact of Dishonest Users

In a competitive market, malicious users may send wrong feedbacks to impact the reputation of some Web services. A dishonest user may send a wrong feedback to decrease the reputation of competitive WS or increase the reputation of an ally. To protect the trust data, we suggest to emphasize and give more weight to honest users while minimizing the impact of liar users.

To minimize the impact of wrong feedbacks, the following are some mechanisms that have been proposed in the literature. These mechanisms could be integrated in our proposed trust management to increase the accuracy of the received ratings:

- Keeping track of past feedbacks: it may be difficult to satisfy a user who always sends negative feedbacks regardless of the high quality provided.
- The multivariate outlier detection technique [10]: this technique is used to detect users that send wrong feedbacks. Outlier detection is an important task in data analysis. The outliers describe the abnormal data behavior which means data that is deviating from the natural data variability.
- The use of trustworthy users: the users that are regular customers are considered as trustworthy users. The

ratings of these users can be given more weights in the reputation computation than the other ratings.

- A voting system: collecting different ratings and using a majority vote to eliminate the false reports.

It is important to investigate how to assess users' credibility, however, this will be addressed in future research.

4 Web Services' Reputation

Let A_k be the feedback of a user k after a transaction. We assume that a transaction involves one WS (a service is realized by a single WS or by a composite of many WS). The user evaluates if the quality of the service received is as agreed or not. If satisfied then a positive feedback is sent to the *Trust Manager* otherwise a negative feedback is sent. If the transaction is satisfactory, we set $A_k = 1$, otherwise, we set $A_k = 0$. In this later case, the delivered quality of service was not as agreed.

Each Web service WS_i in the system has the following reputation data (REP_{WS_i}), stored by the *Trust Manager* entity:

1. N_i^+ : Number of satisfactory transactions,
2. N_i^- : Number of unsatisfactory transactions,

We suggest to use the number of times a Web service has been involved in the transactions, we get the following operation:

If $A_k = 1$ then $N_i^+ ++$, else $N_i^- ++$.

To compute the reputation of a WS_i , we propose to take into consideration the difference between N_i^+ and N_i^- and also the sum of these values as follows:

$$R_i = \frac{N_i^+ - N_i^-}{N_i^+ + N_i^-} \quad \text{if } (N_i^+ + N_i^-) \neq 0 \quad (1)$$

$$R_i = 0 \quad \text{otherwise}$$

The reputation value is a real number between -1 (if $N_i^+ = 0$) and 1 (if $N_i^- = 0$).

When using this reputation scheme, a user can do one of the following scenarios:

1. Choose the WS_i with the maximum value of R_i , or
2. Choose the set of WS such that $R_i \geq R_{threshold}$, where $R_{threshold}$ is a parameter set according to the users requirements which may include other aspects like the cost.

5 Web Services Communities

Web services communities are virtual clusters that agglomerate Web services with the same functionality [9]. In

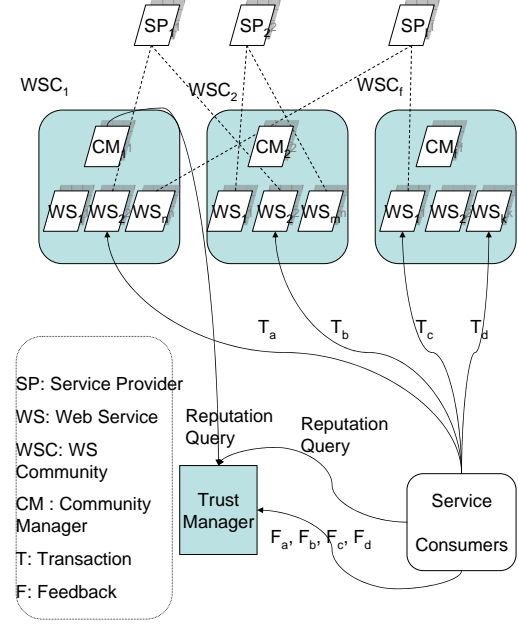


Figure 1. Trust Framework for WS Communities

[9], the authors argued that selecting the best community to deal with is challenging for both the service providers and the users.

Our proposed work can be adapted to rate the communities to facilitate the selection process for both the service providers and service consumers. Incorporating trust in these communities may be achieved as follows:

- A service provider may choose to publish its WS within the community that has a high reputation value.
- A service consumer selects a highly reputable community first and then invokes a Web service within this community.
- A WS community manager will use the trust data to decide the integration of a WS in the community or its dismissal from it due to its unsatisfactory performance.

We propose to compute a WS community reputation (WSC) as follows:

$$WSC = \frac{\sum_i n_i \times R_i}{\sum_i n_i} \quad (2)$$

Where R_i represents the reputation of a WS_i that belongs to the community WSC and n_i represents the number of times WS_i was invoked.

Figure 1 presents the proposed trust framework for Web services communities. A service provider may publish its

Category	Percentage	Probability of providing agreed QoS
WS1	40%	0.95
WS2	30%	0.5
WS3	10%	0.4
WS4	20%	0.2

Table 1. WS Behavior Distribution

WS within different communities. After conducting a transaction with a WS community, a service consumer sends a feedback to the *Trust Manager*. The feedbacks received are based on the quality of service as perceived by the users and based on the agreement with the WS communities. The collected feedbacks will be used to update the reputation of Web services and WS communities. *Reputation queries* to inquire about an entity's reputation are sent to the *Trust Manager*.

6 Performance Evaluation

In the performance evaluation section, we simulate the proposed trust management system (*TrustWS*). We compare the proposed scheme to a system without trust management, where the selection of a WS is, in this sense, random. This is named the *Random* algorithm.

6.1 Simulation Parameters

We use the following simulation parameters:

- The number of Web services is 2000.
- The number of service consumers is 10000.
- WS behavior distribution is as depicted in Table 1.
- We simulate 40000 requests. The simulations were repeated several times over which the results are averaged.

We consider a significant percentage of Web services that may fail to deliver the agreed service to assess the performance of our proposed trust management system.

6.2 Performance Metrics

In this simulations, we focus on the following performance metrics:

- The percentage of unsatisfactory QoS: computed as the number of unsuccessful transactions over the total number of all transactions.
- The users satisfaction: computed as the difference of successful and unsuccessful transactions over the total number of all transactions.

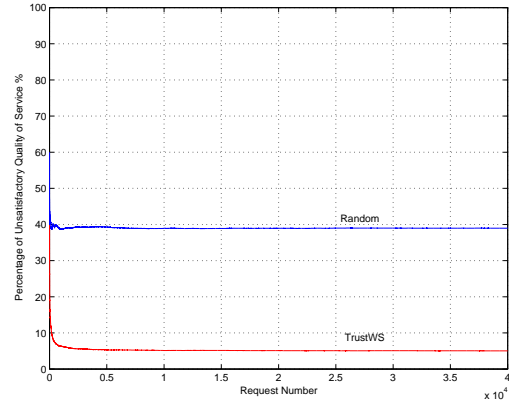


Figure 2. Percentage of Unsatisfactory Quality of Service

6.3 Simulation Results

Figure 2 depicts the percentage of unsatisfactory QoS achieved by the two considered schemes. The *X* axis represents the number of requests while the *Y* axis represents the percentage of unsatisfactory QoS in terms of number of transactions. According to this figure, the proposed trust management scheme *TrustWS* provides better results. The scheme outperforms the *Random* scheme in terms of satisfactory QoS provided to service consumers.

Using *TrustWS*, it is possible to distinguish between the Web services and selecting only the ones that fulfill the QoS requirements. The low performance of the *Random* scheme is justified by the fact that WS are selected independently of their performance in previous transactions. Using the *Random* scheme, 39% of the transactions were not successful and users had very low satisfaction. *TrustWS* has a very low percentage of unsuccessful transactions.

Figure 3 depicts the difference of successful and unsuccessful transactions over the total number of all transactions. In this figure, the *X* axis represents the number of requests while the *Y* axis represents the service consumers satisfaction value. The maximum satisfaction that can be reached is 1 while the minimum value is -1. A positive value shows that successful transactions surpass the unsuccessful ones. *TrustWS* scheme reaches a higher value (0.9) compared to *Random* scheme (only 0.22). Overall, taking the users' feedbacks into account in computing the WS reputation assesses better the expected QoS of a particular WS. Predicting the results of future transactions based on previous ones allows users to make appropriate WS selection leading to a significant increase in users' satisfaction.

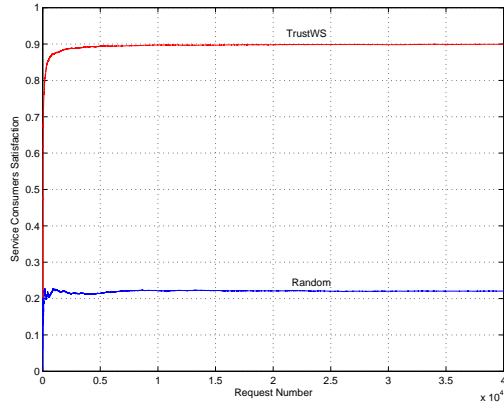


Figure 3. Service Consumers' Satisfaction

7 Conclusion

In this paper, we propose a trust management system for Web services environments. We describe the different components involved in gathering the feedbacks and computing the WS reputation. Performance evaluation results show that the proposed trust management system is able to identify the Web services that do not satisfy the service consumers requirements and consequently, will not be selected to serve users in future requests, hence, increasing users' satisfaction and loyalty.

References

- [1] ebay. <http://www.ebay.com/>.
- [2] Ws-Trust. www.ibm.com/developerworks/webservices/.
- [3] E. Chang, T. Dillon, and F. K. Hussain. *Trust and Reputation for Service-Oriented Environments*. Wiley, 2006.
- [4] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The EigenTrust Algorithm for Reputation Management in P2P Networks. In *International Conference on World Wide Web*, pages 640–651, 2003.
- [5] Z. Malik and A. Bouguettaya. RATEWeb: Reputation Assessment for Trust Establishment among Web Services. *VLDB Journal*, 18(4):885–911, 2009.
- [6] L. Mekouar, Y. Iraqi, and R. Boutaba. Peer-to-Peer Most Wanted: Malicious Peers. *Computer Networks Journal, Special Issue on Management in Peer-to-Peer Systems: Trust, Reputation and Security*, 50(4):545–562, 2006.
- [7] L. Mekouar, Y. Iraqi, and R. Boutaba. *Handbook of Peer-to-Peer Networking*, chapter Reputation Management in Peer-to-Peer Systems: Taxonomy and Anatomy. Springer, 2009.
- [8] Y. Wang and J. Vassileva. Toward Trust and Reputation Based Web Service Selection: A Survey. *International Transactions on Systems Science and Applications Journal, Special Issue on New tendencies on Web Services and Multi-agent Systems*, 3(2):118–132, 2007.

- [9] H. Yahyaoui, Z. Maamar, J. Bentahar, N. Sahli, S. Elnaffar, and P. Thiran. On the Reputation of Communities of Web Services. In *International Conference on New Technologies in Distributed Systems*, pages 1–8, 2008.
- [10] Y. Zhang and Y. Fang. A Fine-Grained Reputation System for Reliable service Selection in Peer-to-Peer Networks. *IEEE Transactions on Parallel and Distributed Systems*, 18(8):1134–1145, 2007.