A User-oriented Approach to Assessing Web Service Trustworthiness[1]

Weinan Zhao, Hailong Sun, Zicheng Huang, Xudong Liu, Xitong Kang,

School of Computer Science and Engineering, Beihang University,
Beijing 100191, China
{zhaown, sunhl, huangzc, liuxd, kangxt}@act.buaa.edu.cn

Abstract. Trustworthiness is a synthetic characteristic of Web services, which not only involves objective attributes of software quality, but also subjective perception of users. However accurate assessment for service trustworthiness from user's perception is a difficult problem, since user perception of a particular Web service varies in terms of users, application scenarios and time. In this paper, we present a user oriented Web service trustworthiness assessment approach based on collecting and aggregating user feedbacks. The difference from other research work is that in our approach we consider application and user specific factors including the similarity between contexts of the evaluator and feedback reporter, the timeliness of feedback and the evaluator's preferences on quality properties of a Web service. A prototype of our approach is implemented based on ServiceXchange, a Web service repository and search engine developed by our research team. Experimental results demonstrate that our approach has a significant advantage over other approaches that treat feedbacks equally and ignore the difference among user preferences.

Keywords: SOA; Web service; trustworthiness assessment; user feedback

1 Introduction

Service oriented architecture (SOA) has been widely accepted in both industry and academia. As an important type of SOA realization technologies, Web services have been successfully adopted in e-commerce, finance, telecommunication and other fields. In practice, individual Web services provide limited functions that are difficult to meet complex business requirements. Therefore by selecting and compositing existing Web services in accordance with a certain business process, service composition becomes the main approach to service-oriented software development.

Among all the challenges faced by service composition, trustworthiness assurance is especially critical to service-oriented software due to highly complicated software structure and uncertain network environments. Typically, trustworthy software means its behaviors and execution results are consistent with user expectations [1], which usually include both functional and non-functional requirements. Therefore, software

[1] This work was partly supported by China 863 program under Grant No. 2007AA010301 and partly supported by China 973 program under Grant No. 2005CB321803.

trustworthiness not only involves objective attributes of software quality, but also subjective perception of users [2]. The most important prerequisite of developing trustworthy software based on service composition is to select individual trustworthy Web services. Hence, an effective Web service trustworthiness assessment approach that can help to locate trustworthy Web services is of paramount importance to service oriented software development.

There have been a lot of research works on software trustworthiness. However, most of them are focusing on the different attributes of software quality while they ignore the user perception of software quality. For example, [3,4,5] study trustworthiness issue from the perspective of automatic monitoring and measuring quality of service (QoS). Compared with trustworthiness of web service, QoS only focus on the objective quality attributes. Those methods are appropriate for measuring the quality properties of a Web service like response time and reliability, but inappropriate for user oriented quality properties like usability and price.

Reputation-based approaches utilize user feedbacks to evaluate a software system including Web services [6,7,8]. The reputation of a Web service is the aggregation of user feedbacks that contain ratings on various software quality properties. Through the use of feedback, these reputation based approaches assess Web service trustworthiness from the user perspective. However, most existing work treats all the user feedbacks equally and ignores the difference among users and application scenarios, which can make the assessment result inaccurate. The feedbacks of a particular Web service from different users may vary greatly due to the following reasons: (a) The quality properties of Web services are context sensitive. In this paper, the *context* denotes the input parameters and invocation environment of Web service. For example, a Web service may have shorter response time to LAN users than Internet users. (b) The user preferences on quality properties can vary greatly. A time-critical user may give higher overall rating to an expensive real-time Web service than other users who are concerned more about price. (c) The quality properties of Web services are dynamically changing due to upgrade and the changing of underlying infrastructure. Therefore, assessment of a Web service by simply aggregating feedbacks without considering specific contexts, user preferences and the timeliness of feedbacks cannot obtain accurate results.

To address these problems, we propose a user oriented approach to assessing Web service trustworthiness. Our approach collects feedbacks that contain reporter's contextual information and multi-dimensional ratings on service quality properties. When a user (*evaluator*) wants to learn the trustworthiness of a specific Web service, he is required to submit his context and preferences. The assessment procedure is divided into two steps. First, quality properties are predicted by aggregating the feedbacks, that are weighted by the timeliness of feedbacks and the context similarity between evaluator and feedback reporter. Second, the predicted quality properties are aggregated into an overall trustworthiness using evaluator's preferences as weight. Finally, we implement a prototype on the basis of our ServiceXchange [9], a Web service repository and search engine developed by our R&D team at Beihang University. The major contributions of this paper are as follow:

- We propose a user oriented approach to assessing Web service trustworthiness from user's perspective, with consideration of the difference among users' contexts and the difference among user preferences on quality properties as well as the timeliness of user feedbacks.

- We design and implement a prototype of our approach base on our previous work, namely ServiceXchange.
- We demonstrate the assessment accuracy of our approach through experimental evaluation.

The remainder of this paper is organized as follows. Section 2 introduces the related work. Section 3 describes the proposed approach in detail. Section 4 describes the prototype of our approach. Section 5 presents experiments and results. Finally we conclude our work in Section 6.

2 Related Work

Generally speaking, the study of Web service trustworthiness assessment is performed along two technical directions, i.e. test-based and reputation-based approaches.

Test-based approaches use monitored QoS attributes to assess Web service trustworthiness. Shao et al [3] propose a user-perceived service availability metric on the basis of service status. The service status is described as *stable up*, *transient down*, and *persistent down*. By analyzing the service-invocation records, the service status feature is recognized and used to measure Web service trustworthiness. Zhang et al [4] present the design and implementation of a Web services testing platform. By utilizing of mobile agent technology, the total testing time and network traffic are reduced. Bai et al [5] propose an approach to generate Web services test cases automatically based on the WSDL (Web Services Description Language), which carries the basic information of a service including its interface operations and the data transmitted. These approaches effectively and efficiently facilitate Web service testing, but are limited to get network related quality metrics like response time and availability.

The reputation-based approaches measure trustworthiness by collecting and aggregating user feedbacks. Zaki et al [8] propose a model to compute the reputation of a Web service according to other users' feedbacks, and the user credibility is taken into consideration in the proposed method. Surya et al [6] present a composite service oriented reputation system. When a rating is given to a composite service, the system will distribute it to the component Web services fairly. Zaki et al [7] propose two techniques to resolve the reputation bootstrap issue, with which the initial reputation is set up to a new comer service in a fair and accurate manner. Reputation system is also extensively studied in other fields such as E-commerce and P2P networks [10,11,12]. However those works use feedback with overall rating that ignores the difference among user preferences, and aggregates user feedbacks without considering the context sensitivity and timeliness of feedbacks.

Different from the existing methods, we propose a user oriented Web service trustworthiness assessment approach. By collecting feedback with multidimensional ratings on quality properties instead of an overall rating, the misleading effect caused by the difference among user preferences is eliminated. By using context similarity as weight to aggregate feedbacks, we only make use of feedbacks that have similar context with evaluator to predict quality properties of Web services. Additionally, the timeliness of feedbacks is also taken into account so as to differentiate the time effectiveness of user feedbacks.

3 User Oriented Trustworthiness Assessment Approach

3.1 Overview

Fig.1 gives an overview of our approach to assessing Web service trustworthiness. Firstly the feedbacks are collected as a basis. Then the assessment process is divided into two steps: (a) Predict multidimensional quality properties, using feedbacks that are weighted by the timeliness of feedback and the context similarity between evaluator and feedback reporter. (b) Aggregate the predicted quality properties into an overall trustworthiness incorporating evaluator's preferences.

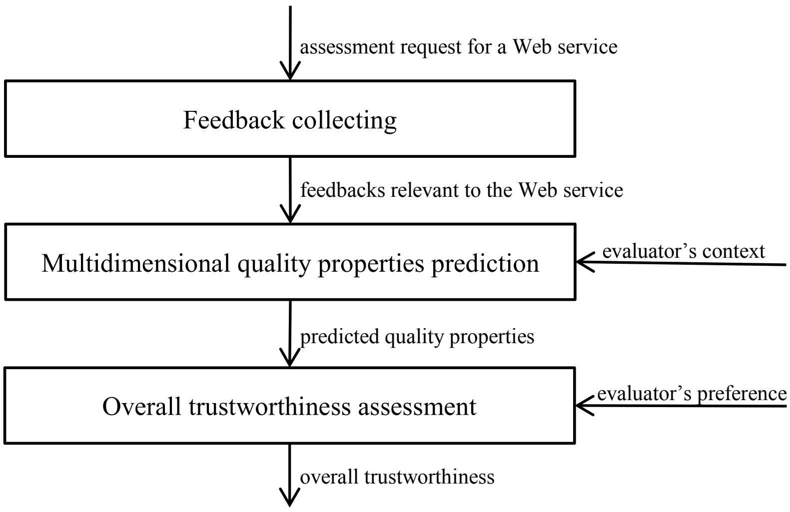

Fig. 1. The overview of our approach.

Then we propose our user oriented assessment approach in detail. First, the consideration about feedback collecting is discussed. Second, the method to predict quality properties under a specific context is presented. Third, the timeliness of feedback is taken account into the quality properties prediction. Finally, we introduce the calculation of overall trustworthiness incorporating user preference.

3.2 Feedback Collecting

To assess the Web service trustworthiness under the evaluator's specific context and preference, we collect feedbacks that contain contextual information and multidimensional ratings on quality properties into a *feedback repository*. The collecting of multidimensional ratings instead of an overall rating will clear away the preferences of feedback reporters. And by recording reporter's contextual information and comparing it to evaluator's context, we can know whether a feedback should be involved in the assessment. In addition, throughout this paper we will utilize *context of feedback* and *context of feedback reporter* interchangeably to denote the context in which the Web service is invoked, and *context of evaluator* to denote the context in which the evaluator want to use the Web service.

To collect multidimensional feedbacks, a feedback model should be defined first. A feedback model describes what information should be collected while users submit feedbacks.

We give the formal definition of feedback model and user feedback as follows.

Definition 1.A *feedback model* is defined as $<Context, Quality>$, where $Context=<ctx_1,ctx_2,...,ctx_i>$ defines a set of context factor and $Quality=<qlt_1,qlt_2,...,qlt_i>$ defines a set of quality properties.

Definition 2. A *user feedback* reported by user U at time T about Web service S is defined as $<U,S,T,C,Q>$, where $C=<c_1,c_2,...,c_i>$ defines a set of context factor values and $Q=<q_1,q_2,...,q_i>$ defines a set of quality property values. Without loss of generality, the domain of quality property value is set to between 0 and 1. And in this paper we only consider the case where context factors have discrete or categorical values.

In practice, users focus on different quality properties considering services in different categories. Hence a domain specific feedback model should be given for each service category, and this is usually completed by domain experts.

For instance, considering services in *search engine category*, the feedback model may be $<Context, Quality>$ where $Context=<User\ Location, Search\ Language>$ and $Quality=<Response\ time, Recall, Precision>$. Then a user feedback of a service in the category may be $<U,S,T,C,Q>$ where $C=<RPC, English>$ and $Q=<0.6,0.7,0.7>$.

3.3 Predict Quality Properties under a Specific Context

Based on multidimensional user feedbacks from the feedback repository, the quality properties under a specific context can be predicted. Assuming that a user requests to assess the trustworthiness of a Web service S under the specific context C, we firstly fetch feedbacks related to S from feedback repository into a feedback set FS and for each feedback F in FS, the similarity between C and context of F is calculated. Then we use the similarity as weight to aggregate the feedbacks in FS to predict the multidimensional quality properties of S.

The core part of the prediction is the context similarity calculation. Then we give the consideration and design of the calculation method as follows.

We call two contexts is similar to a Web service iff the difference between service's quality property values under the two contexts is less than a threshold. In practice, the similarity between two contexts is sensitivity to quality properties. For instance, considering the quality property values of a search engine Web service under two context C_1 and C_2 , where $C_1=<UK, Chinese>$ and

C_2= <*UK, English*>, the values of quality property *Response Time* may be close since the values of context factor *User Location* are same while the values of quality property *Precision* may be obviously different due to the values of context factor *Search Language* vary (e.g. Google performs better while is used to search English information than Chinese). Therefore C_1 and C_2 are similar to the search engine Web service considering the property *Response Time* and not similar considering the property *Precision*. Based on the above discussion, the similarity between two contexts should be calculated for each quality property individually.

The formal definition and calculation method of context similarity is given as follows.

Definition 3. *The distance between two context factor values* c_i *and* c_j *considering quality property* qlt_k *of service S* is $dis^S_{qlt_k}(c_i, c_j)$ which is defined as:

$$dis^S_{qlt_k}(c_i, c_j) = \left|\overline{q^S_k(c_i)} - \overline{q^S_k(c_j)}\right|,\ c_i, c_j \in domain(ctx_m). \quad \textbf{(1)}$$

Where:

$$\overline{q^S_k(c_i)} = \frac{1}{n}\sum_{F_i \in FS'} q_k, FS' = \{F_j | S \in F_j \wedge C \in F_j \wedge c_i \in C\}, n = |FS'|. \quad \textbf{(2)}$$

Definition 4. *The similarity of context* C_1 *and* C_2 *considering property* qlt_k *of service S* is $sim^S_{qlt}(C_1,C_2)$ which is defined as:

$$sim^S_{qlt_k}(C_1,C_2) = 1 - \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left[dis^S_{qlt_k}(c^1_i,c^2_i)\right]^2}, c^j_i \in C_j, n = |C_j|. \quad \textbf{(3)}$$

Note that $dis^S_{qlt_k}(c_i, c_j) \in [0,1]$, and larger value of it implies greater change of value of quality property qlt_k when the value of context factor ctx_m vary from c_i to c_j. Then $sim^S_{qlt_k}(C_1,C_2) \in [0,1]$, and larger value of it implies higher similarity between C_1 and C_2 considering property qlt_k of *S*.

For instance, $dis^S_{Precision}$ (*Chinese, English*) is the distance between context factor values *Chinese* and *English*, and $sim^S_{Precision}(C_1,C_2)$ is the similarity between contexts C_1 and C_2 considering quality property *Precision* of service *S*.

To predict quality properties of service *S* under evaluator's specific context *C*, we average values of the quality properties in feedbacks related to S, using similarity between *C* and feedback reports' context as weight. A threshold ∂ is used to filter out the feedbacks that have context not similar to *C*. The following equation gives the formally definition of the prediction, where $pq_k(C)$ denotes the predicted value of quality property qlt_k under context *C*.

$$pq_k(C) = \frac{\sum_{F_i \in FS} sim'^S_{qlt_k}(C_i,C) \times q^i_k}{\sum_{F_i \in FS} sim'^S_{qlt_k}(C_i,C)}, C_i, Q_i \in F_i, q^i_k \in Q_i. \quad \textbf{(4)}$$

Where:

$$sim'^S_{qlt_k} = \begin{cases} sim^S_{qlt_k} & , if\ sim'_{qlt_k} > \partial. \\ 0 & , otherwise. \end{cases} \quad \textbf{(5)}$$

It is worth noting that $dis^S_{qlt_k}(c_i, c_j)$ reflects the Web service's inherent characteristic that is relative stable, so it could be calculated offline and stored in a lookup table to facilitate online context similarity calculation.

3.4 Timeliness Consideration of Feedback

To cope with the dynamic feature of service quality properties, we give less weight to old feedback than more recent ones. A fading factor is introduced as weight to weaken the old feedbacks' importance on prediction [10]. Suppose that the current time is T_{now}, the fading factor of a feedback F is defined as follows.

$$\gamma(T_i,T_{now})=\tau^{T_{now}-T_i},T_i \in F,\tau \in [0,1]\,. \qquad \textbf{(6)}$$

By adjusting the value of τ, different fading effect can be achieved. It's complete fading when $\tau=0$, that only the latest feedbacks are counted and all previous feedbacks are completely swept out. And having $\tau=1$ is equivalent to not having a fading factor.

Based on the introduced fading factor, we give the formal definition of the quality properties prediction that takes timeliness of feedback into account as follows.

$$pq_k(C,T_{now})=\frac{\sum_{F_i\in FS} sim'^S_{qlt_k}(C_i,C)\times\gamma(T_i,T_{now})\times q^i_k}{\sum_{F_i\in FS} sim'^S_{qlt_k}(C_i,C)\times\gamma(T_i,T_{now})},T_i,C_i,Q_i \in F_i\,,q^i_k \in Q_i\,. \quad (7)$$

3.5 Incorporation of User Preference

By collecting and aggregating the multidimensional feedbacks, the quality properties of a Web service are predicted with the feedback reporter's preference eliminated. The predicted quality properties can be delivered to evaluator as multidimensional trustworthiness, but typically the comparison among candidate services' trustworthiness is conducted to help selecting the most trustworthy Web service, so an overall trustworthiness is required as a sorting index. We give a method to assess the overall trustworthiness of a Web service base on the incorporation of the evaluator's own preference into the predicted quality properties.

First we give the formal definition of user preference on multidimensional quality properties as follows.

Definition 5. *A user preference on quality properties of a Web service* is defined as $P=<p_1,p_2,...p_k>,p_k \in [0,1], \sum p_k=1$, where p_k is the importance to the quality property qlt_k from the user perspective.

Then the aggregation method for overall trustworthiness under user specific context *C* and preference *P* at particular time T_{now} is given as follow.

$$T(P,C,T_{now})=\sum p_k\times pq_k(C,T_{now}),p_k\in P. \qquad \textbf{(8)}$$

4 Prototype Design and Implementation

In this section, we will first introduce a Web service repository and search engine that developed by our R&D team, namely ServiceXchange, and then present the implementation of a prototype of the proposed trustworthiness assessment approach based on ServiceXchange.

4.1 ServiceXchange

ServiceXchange is a Web service repository and search engine that provides functions of Web service registering, searching, monitoring, etc. Fig.2 shows the system architecture of ServiceXchange. Web services are collected into ServiceXchange using a crawler, and a UDDI-like registry is also available for software provider to register their Web service into ServiceXchange. By analyzing the WSDL of Web services, the information such as interface operations and data transmitted is stored in Raw Info DB. Further knowledge of Web services such as similarity, connectivity, cluster and the keyword index are discovered leveraging the data mining technology and stored in Mined Info DB. Users can search Web services by keyword or category. To measure QoS of Web services, automatic built clients are used to monitor Web services and measure the response time and availability.

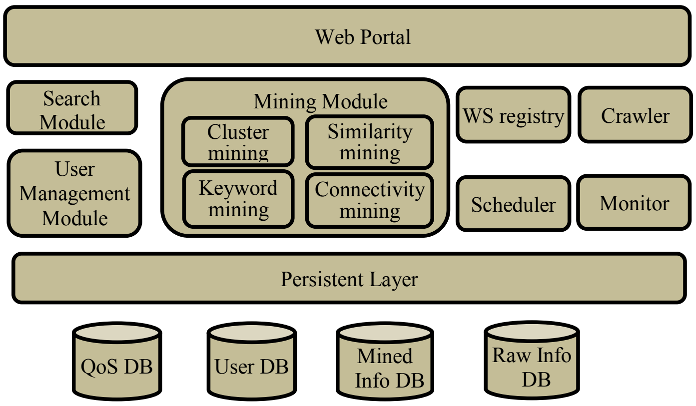

Fig.2. System architecture of ServiceXchange.

4.2 Implementation of a Prototype

On the basis of the design presented in section 3, we implement a prototype integrating our trustworthiness assessment approach with ServiceXchange. Fig. 3 shows the system architecture of the prototype.

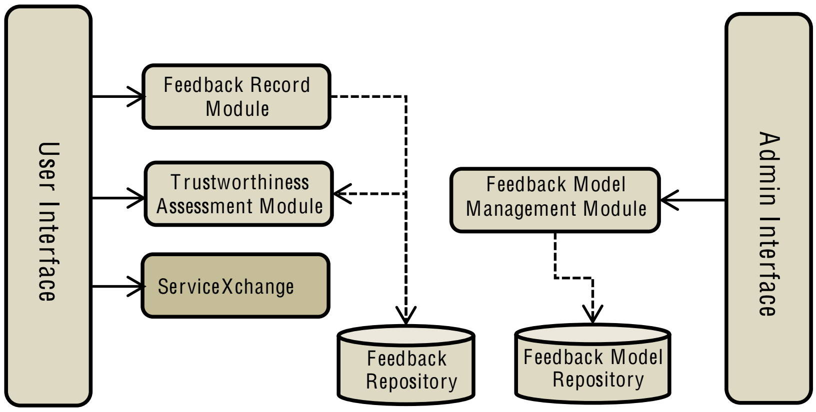

Fig. 3. System architecture of the prototype.

Feedback model management module provides tools to domain experts for building feedback models. For each Web services category in ServiceXchange, a feedback model that contains the contextual factors and quality properties is built leveraging domain knowledge. When a user want to report a feedback about a Web service, the feedback record module automatically generates a HTML form according to feedback model, then the user can report multidimensional feedbacks that will be stored into the Feedback Repository. The trustworthiness assessment module support two interfaces to assessing trustworthiness: (a) Submit a context, then get the assessed multidimensional quality properties of a particular Web service. (b) Submit a context and user preference, then get the assessed overall trustworthiness of a particular Web service or sort the Web services in same category by the overall trustworthiness.

5 Experimental Evaluation

In this section, we present experimental evaluation of our approach for trustworthiness assessment. Since our prototype based on ServiceXchange is not open to public yet, few real-world feedbacks can be used to evaluate our approach. In order to evaluate the assessment accuracy of our proposed methods, we conducted a simulation. The main purpose of this simulation is to measure the performance of our approach in assessing Web service trustworthiness. To clearly show the effectiveness of our approach, we implement the following five different assessment methods and perform comparative analysis of them in terms of assessment accuracy.

- **Web service trustworthiness assessment using user oriented approach (**$WSTA_{UO}$**).** This approach is proposed in section 3.
- Web service trustworthiness assessment ignoring user preferences ($WSTA_{IUP}$). Compared with $WSTA_{UO}$, this method does not consider the difference among user preferences. It collects the overall ratings incorporating the preference of *feedback reporters* that may be different from the preference of the *evaluator*, and then

aggregates them using the context similarity and fading factor to weighting the feedbacks from different users.

- **Web service trustworthiness assessment ignoring context (*$WSTA_{IC}$*)**. Compared with $WSTA_{UO}$, this approach does not consider user contexts. It first collects multidimensional feedbacks, then predicts quality properties using the fading factor of feedbacks as weight, and finally aggregates the predicted quality values into an overall trustworthiness using evaluator's preference as weight.
- **Web service trustworthiness assessment ignoring timeliness (*$WSTA_{IT}$*).** Compared with $WSTA_{UO}$, this method ignores the timeliness of feedbacks. It collects multidimensional feedbacks, and then predicts quality properties using the context similarity as weight, finally aggregate the predicted quality property values into an overall trustworthiness using evaluator's preference as weight.
- **Web service trustworthiness assessment with overall rating (*$WSTA_{OR}$*)**. This approach collects overall ratings and evaluates trustworthiness simply by averaging the ratings, ignoring all the considerations that are taken into account by $WSTA_{UO}$.

5.1 Simulation Parameters

There are three types of parameters in the simulation:

1. **Simulation environment parameters.** Simulations are run for 100 epochs, and a user assesses a Web service with the probability of 0.5. After a user obtains the assessed trustworthiness value, he invokes the Web service to get the real trustworthiness value and reports a feedback. If the difference between the assessed and real trustworthiness value is less than a threshold ε, the assessment is considered to be accurate. The threshold ε is set to 0.1. The five assessment approaches are used concurrently for each assessment. At the end of each epoch, the ratio of accurate assessment for each approach is record and compared.
2. **User parameters.** The number of users is set to 100. The characteristic of a user consist of two aspects: (a) User's context. A user's context contains two dimensions and each dimension has two possible values, i.e. 0 or 1. Then the space of user context value is $C^*=\{C=<c_1,c_2>|c_i\in[0,1]\}$. (b)User preference on multidimensional service quality properties. The user preference is a tuple whose elements are set to random numbers.
3. **Web service parameters.** Considering the candidate Web service set that contains 20 services, namely service set $S=\{s_1,s_2,...,s_{20}\}$. Each service has three quality properties. When a service is invoked, the user perceived quality properties are determined by the following two factors: (a) The service's inherent feature, which follows normal distribution $N(\mu,\sigma^2)$. The μ and σ is set to a random number between 0 and 1. (b) The context adjustment factor δ_s that is affected by user context. Then the user perceived quality property values of a service s under specific context C is defined as follows.

$$Q_s(C)=<q_1,q_2,q_3>,q_i=N(\mu,\sigma^2)+\delta_s(C). \quad (9)$$

In order to simulate the impact of the context on the quality properties of a service, we divide the services set S into two subset: S_1, S_2 where $S_1 \cap S_2=\emptyset$, and divide the user context space C^* into two subset: C_1^*, C_2^* where $C_1^* \cap C_2^*=\emptyset$, then set the context adjustment factor δ_s as follows.

$$\delta_s(C)=\begin{cases} -0.2, if\ s \in S_1 \wedge C \in C_1^*. \\ 0.2,\ if\ s \in S_1 \wedge C \in C_2^*. \\ 0.2, if\ s \in S_2 \wedge C \in C_1^*. \\ -0.2, if\ s \in S_2 \wedge C \in C_2^*. \end{cases} \tag{10}$$

The above formula implies that the services in S_1 will the performance better under the contexts in C_1^* than under the contexts in C_2^* , and services in S_2 is just the opposite.

Note that the calculated result of q_i according to the equation 9 and 10 may be out of the range 0 to 1, we just set 1 to result values greater than 1 and set 0 to result values less than 0. To simulate the dynamic feature of Web services, each service is set to have a probability of randomly changing its quality properties, and this probability is set to 0.2.

5.2 Simulation Results

At the beginning of the simulation, the accuracy of all approaches is low since few feedbacks are obtained. As the simulation advanced, more feedbacks are reported and used to assess the trustworthiness of Web services. Fig.4 shows the comparison between simulation results using $WSTA_{UO}$ and $WSTA_{IUP}$. $WSTA_{UO}$ maintains the accuracy of about 0.7 while $WSTA_{IUP}$ only 0.3. $WSTA_{IUP}$ suffers from low accuracy because the ignorance of the difference among user preferences. Fig. 5 shows the comparison between the simulation results using $WSTA_{UO}$ and $WSTA_{IC}$, $WSTA_{IC}$ gains poor accuracy about 0.2 for ignoring the difference among user contexts.

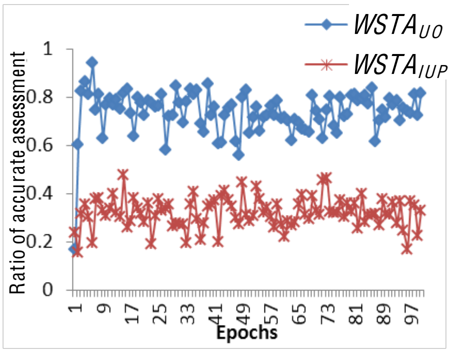

Fig. 4. Simulation results of $WSTA_{UO}$ and $WSTA_{IUP}$.

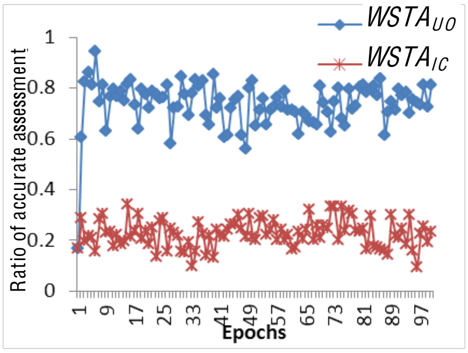

Fig. 5. Simulation results of $WSTA_{UO}$ and $WSTA_{IC}$.

Fig. 6 shows the comparison between the simulation results using $WSTA_{UO}$ and $WSTA_{IT}$, the accuracy of $WSTA_{IT}$ is same as $WSTA_{UO}$ initially. However, as simulation progress, some services change their quality properties values randomly. Without taking the timeliness of feedbacks into consideration, $WSTA_{IT}$ is misled by outdated feedbacks and its accuracy decreases sharply with time. Fig. 7 shows the comparison between simulation result using $WSTA_{UO}$ and $WSTA_{OR}$. By collecting overall ratings and treating them equally, $WSTA_{OR}$ gets poor accuracy about 0.2.

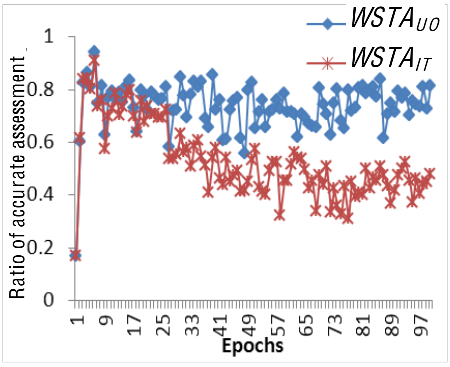

Fig. 6. Simulation results of $WSTA_{UO}$ and $WSTA_{IT}$.

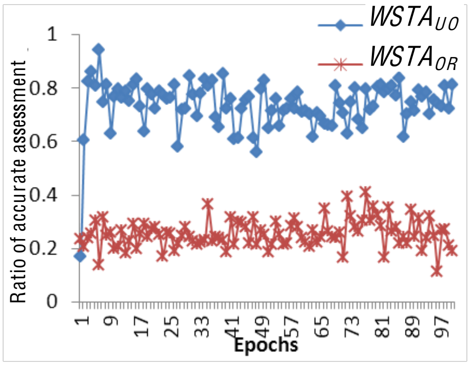

Fig. 7. Simulation results of $WSTA_{UO}$ and $WSTA_{OR}$.

6 Conclusion

We have presented a user oriented approach to assessing Web service trustworthiness. Different from other approaches, this approach runs the assessment from the perception of users, taking account of the context sensitivity and dynamic feature of service quality properties as well as the difference among user preferences.

A prototype of our approach is implemented on the basis of a Web service repository and search engine, namely ServiceXchange. Users can discover services that meet functional requirements with ServiceXchange, and then our proposed approach can be used to assess the trustworthiness of the Web services.

The simulation results show that our approach has a significant accuracy advantage over other approaches that treat feedbacks equally and ignore the difference among user preferences.

For future work, we plan to release ServiceXchange combining with our trustworthiness assessment approach to public and collect real-world feedbacks to evaluate our approach. Moreover, we will design an algorithm to identity malicious users that report fake feedbacks.

References

1. H. Wang, G. Yin.: Trustworthy software evolution in the Internet era (in Chinese). Communications of the CCF, Volume 6, No.2, (2010)
2. H. Mei, D. Cao.: Trustworthiness of software: Challenges from Internet (in Chinese). Communications of the CCF, Volume 6, No.2, (2010)
3. L. Shao, J. Zhang, T. Xie, L. Zhang, B. Xie, and H. Mei.: User-Perceived Service Availability: A Metric and an Estimation Approach. In: Proceedings of IEEE International Conference on Web Services, pp.647-654. IEEE Computer Society. Los Angeles (2009)
4. J. Zhang, D. Xu.: A Mobile Agent-Supported Web Services Testing Platform. In: Proceedings of the 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing - Volume 02, pp. 637-644. IEEE Computer Society, Washington (2008)
5. X. Bai, W. Dong, W.-T. Tsai, and Y. Chen.: WSDL-Based Automatic Test Case Generation for Web Services Testing. In: Proceedings of IEEE International Workshop on Service-Oriented System Engineering, pp. 215–220. IEEE Computer Society, Los Alamitos (2005)
6. S. Nepal, Z. Malik, and A. Bouguettaya.: Reputation Propagation in Composite Services. In: Proceedings of the 2009 IEEE International Conference on Web Services, pp. 295-302. IEEE Computer Society, Washington (2009)
7. Z. Malik, A. Bouguettaya.: Reputation Bootstrapping for Trust Establishment among Web Services. IEEE Internet Computing 13(1), 40–47 (2009)
8. Z. Malik, A. Bouguettaya.: Evaluating Rater Credibility for Reputation Assessment of Web Services. In: Proceedings of the 8th International Conference on Web Information Systems Engineering, pp. 38-49. Springer, Nancy-France (2007)
9. ServiceXchange, http:// www.servicexchange.cn
10. A. Jøsang and R. Ismail.: The Beta Reputation System. In: Proceedings of the 15th Bled Electronic Commerce Conference, Bled, Slovenia (2002)
11. M. Gupta, P. Judge, and M. Ammar.: A reputation system for peer-to-peer networks. In: ACM 13th International Workshop on Network and Operating Systems Support for Digital Audio and Video, Monterey, CA (2003)
12. Wang Y. Vassileva J.: Trust and Reputation Model in Peer-to-Peer Networks. In: Proceedings of IEEE Conference on P2P Computing, Linkoeping, Sweden (September 2003)