I.J. Information Technology and Computer Science, **2015, 08, 31-36**
Published Online July 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2015.08.05

A Dynamic Reputation-Based Approach for Web Services Discovery

Emad Elabd
Information Systems Department, Faculty of computers and information, Menoufia University, Egypt
E-mail: emadqap@gmail.com

Abstract—Web services discovery is a crucial process in the service-oriented architecture (SOA). This process depends on the functional and non-functional (Quality of services (QoS)) properties of the published services. Functional properties present the objectives of the service and the quality of the service indicates the performance of the service. Reputation of the services is one of the important QoS properties. Assigning robust and precise reputation values for the service affects positively in the output of the web services discovery process by returning the most relevant services for the consumer. In this paper, we propose an approach for updating reputation of the Web services based on the trust factors of the consumers and two pre-defined thresholds, reputation threshold, and agreement threshold. The proposed approach is implemented and tested on a set of services. The results show that the proposed reputation updating approach is powerful and gives robust and precise results in a dynamic way. As a consequence, the web service discovery process which is based on the proposed approach gives a robust and precise results in a dynamic way.

Index Terms— Quality of Service (QoS), Reputation, Trust Factor, Web Service Discovery.

I. INTRODUCTION

Web services technology is emerging as a main pillar of service-oriented architectures (SOA) [1]. This technology facilitates application integration by enabling programmatic access to applications through standard XML-based languages and protocols. An SOA approach solves many problems between the distributed enterprises information systems such as application integration, transaction management, and security policies [15]. Therefore, Web services in SOA can be shared and reused [16]. These facilities make the Web services indispensable for applications in the same enterprise or in different enterprises.

The services in the SOA are presented as self-contained software modules. These services are described using a standard Web service description language and provide a business functionality. Web services use a set of standards such as the WSDL, SOAP, and UDDI. The main objective of a service in an SOA is to represent a reusable unit of business-complete work. All the service providers publish the descriptions of their services in the registry (UDDI). The description of each service includes its functional and non-functional properties (QoS). Functional properties present the objectives of the service and the quality of service is a set of non-functional attributes such as response time, throughput, reliability, reputation and availability [14].

At the beginning of the emergence of the Web services, the consumer search for a service based on it functional properties only in the discovery process. After that, a set of problem is emerged during the invoking process because of the incompatibility between the quality of the service and the required one. Therefore, the discovery approaches starts to include the non-functional properties (QoS) beside the functional properties to select the most relevant set of services for the consumer using a set of different QoS levels [12].

In SOA, both service providers and consumers should be able to define QoS-related statements to enable QoS-aware service discovery. The returned service from the discovery process should be the most relevant to the consumer request. Therefore, the values of the QoS attributes effects on the discovery process. Some of the QoS-based services selection approaches assume that the QoS data coming from service providers and the consumers are effective and trustworthy. However, the values of QoS attributes which are provided by service providers may be unbelievable, since service providers sometimes may advertise higher QoS data than the factual level of the service in order to attract more users to use their services and so gain better benefits [11]. For example, the maximum response time of these services may be increased while the supplication rate remains under a certain threshold during runtime.

One of the important attributes of the QoS is the reputation of the service which is based on the feedback from the consumers. Service reputation is changed during the access of the service based on the different feedback of the consumers. The dishonest consumers can assign incorrect values for the service reputation which effects in the discovery process by returning incorrectly ranked list of relevant Web service for the consumer. Therefore, updating service reputation preciously to help the consumer to reference and choose the appropriate service becomes a problem to solve [3].

In this paper, a dynamic reputation updating approach is provided. This approach is the basis for a robust and precise Web service discovery approach. The main idea of this approach is to ensure that the value assigned by the consumer for the reputation of the service is correct by checking the trust level of the consumer, the agreement threshold (i.e., the set of consumer who assigned values near to this value) and the reputation threshold level (i.e., the allowed margin between this assigned reputation value and the current).

This paper is organized as follows. Section II outlines the related work to our proposed model. The proposed discovery and selection model is illustrated in Section III. Section IV presents the experiments that evaluate the effectiveness of our model and the effect of the new reputation algorithm. Section V presents the conclusion and the future work.

II. Related Work

There are many approach that are presented for the Web services discovery process [14, 4, 7, 5, 8, 13, 10, 3, 9]. Some of these works are based on the functional properties and the others are based on both the functional and nonfunctional properties (QoS).

Zhang et al. [13] present a tool for WS-QoS measurement. This tool calculates the reputation based on the similarity values between the value offered by the service and the measured quality data value. This algorithm is not updated for the trust and reputations, making trustworthiness information reflect the latest changes in service

Sathya et al. [10] evaluate the various techniques that are used in the quality of service based service discovery approach. They defined a set of criteria for QoS discovery approach. They organized the approaches into three main categories: Functional based service discovery approach, non-Functional based service discovery approach, and user-based service discovery approach.

Wang et al. [9] propose an approach for measuring reputation precisely. They give a solution for the malicious rating of service users. Their approach consists of two phases. In the first phase, they detect malicious feedback ratings using “Cumulative Sum Method”. The second phase, they use the Pearson Correlation Coefficient to detect and reduce the effect of different user feedback.

Nianhua et al. [8] present a reputation evaluation algorithm based on the similarity theory for the new added Web service. They use trust and similarities as weights for computing reputations from different recommenders.

Zhao et al. [6] designed a reputation-aware model for service discovery based on an adaptive adjusting reputation evaluation method of Web services. This evaluation is based on eliminating the collusive behaviors of consumers step by step. They compute the reputation score based on a subjective judgment of service users but not an objective measurement

Josang et al. [5] use Bayesian reputation systems with a trust model to evaluating the quality of service in a single framework. Nepal et al. [2] present a fuzzy trust evaluation approach for Web services. They present a trust-based reputation management framework for Web service selection.

Ran [1] extends the traditional SOA model, which consists of three roles: service provider, service consumer and UDDI registry, with a new role called a Certifier. The QoS of the advertised Web service is verified by the certifier before its registration. The consumer can also verify the QoS claims to guarantee satisfactory transactions. Although this model incorporates QoS into the UDDI, it does not integrate consumer feedback into the discovery process.

Al-Sharawneh et al. [7] presented a Web service selection model based on the assessment and prediction of the reputation values in the service oriented architecture. They aggregate the feedback of different aspects of the ratings by what they called reputation key metrics. In addition, they proposed a Feedback Forecasting Model (FFM) to predict the reputation of a web service in dynamically and integrate it into aspect-based reputation management framework. The semantic concept reputation aspect and how to compute it efficiently is introduced in this work too.

Xu et al. [3] extended the UDDI to store the QoS information and presented a web service selection model. In addition, they proposed a reputation management system to build and maintain service reputations and a discovery agent to facilitate service selection.

Serhani et al. [4] present an architecture to support QoS management and enforcement for both consumers and providers of Web service. In their system, the providers are allowed to introduce QoS-centered annotations in the service description. In addition, they developed a validation process to test the service interface and the QoS for the providers before publishing its services. The consumers are allowed to introduce theirs QoS requirements in their requests to selected Web service.

Al-Shargabi et al. [17] present a Web service composition technique based on the user preferences such as price and availability. They use a Web service selection agent forced by users’ preferences. The effect of the reputation in the selection approach did not be included.

To the best of our knowledge, building dynamic reputation-based Web service discovery model based on the consumers trust factors is not considered in the previous works.

III. The Proposed Approach

The traditional service-oriented architecture for Web services discovery has three roles: service provider, service consumer and Web service registry. In the proposed model shown in Figure 1, the Web service registry includes the Web services description, QoS, and a reputation-based discovery unit. In addition, the service provider and the service consumer have their own quality matrices. The reputation-based discovery unit acts as a third party between the service consumer and the service provider that select the Web services that satisfies the consumer functional, QoS and Reputation requirements. The reputation unit collects and processes service reputation from consumer, and update the service reputation scores.

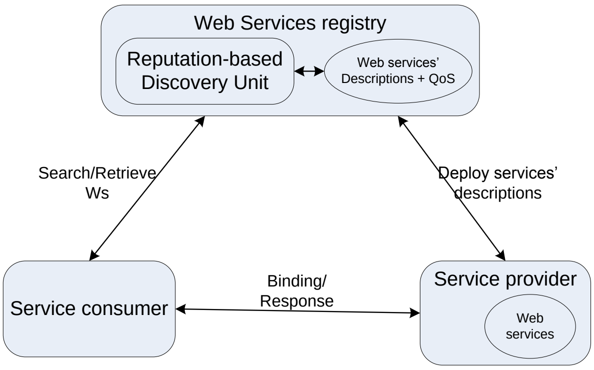

Fig.1. SOA and Web Services Discovery Model based on Reputation

As shown in Figure 1, the operation of the proposed model is as follows: The service provider publishes the service description and the QoS information in the Web service registry. The consumer requests a service that matches its QoS matrices using the discovery unit. The discovery unit receives the request from consumer and returns the services that match his/her requirements. Functional QoS and reputation requirements are specified in the discovery request. The consumer receives result as a list of Web services from the discovery unit. The consumer requests the services from the provider after receiving the result of searching from the discovery unit. After the consumer gets the service, s/he can assign a reputation score for the service. Then the reputation module update the reputation value for the service based on the value assigned by the consumer, the trust factor of the consumer, and the previous reputation value of the service and two pre-defined thresholds using the reputation algorithm (Algorithm 1).

A. Reputation module

The service reputation module is responsible for collecting data from the service consumer, processing data, updating the reputation scores for related service provider. The QoS reputation score in the proposed approach is calculated based on the reputation assigned by the service consumers, their trust factors, Trust Factor Threshold (TFT), Agreement Threshold (AT), Reputation Threshold (RT), and the current reputation value of the service.

We have three different thresholds used to make the algorithm adaptive. The first is the trust factor threshold which restrict the set of accepted consumer reputation values based on their trust factors. The discovery unit will not accept the reputation values assigned by users with trust factors less than the trust factor threshold. For example, if the trust factor threshold is 0.8 then only users' reputation values of the consumers with a trust factor greater than or equal 0.8 will be considered. The second threshold is the reputation threshold which is used to set the acceptable margin of difference between the assigned reputation value, by consumer, and the current service reputation value. The third is the agreement threshold that shows the number of the consumers who are considered for calculating the reputation threshold. For example, if the reputation threshold is 10% and the agreement threshold is 3, this means that if there are three consumers with trust factor values greater than or equal to the trust factor threshold and their reputation values satisfy the reputation threshold between any two values of them, then these values are considered in the calculation of the reputation.

Algorithm 1: Reputation algorithm.

- I/P
  - Current reputation value of the service (it is null in the beginning).
  - The trust factor threshold (TFT), reputation threshold (RT), and the agreement threshold (AT).
  - Consumer assigned reputation value.
  - Consumer trust factor (can be null, in case that the consumer does not access any services from this registry before)
- O/P
  - Updated reputation value for the service.
  - Updated trust factor value for the consumer.

For each new reputation value a_j assigned by consumer j

- *If Tfc_j>=TFT or Null, where Tfc_j is the current trust factor of consumer j, and TFT is the trust factor threshold, then*
  - *If $R_c(s_i)$=Null, where $R_c(s_i)$ is the current service reputation.*
    - $R_u(s_i) = a_j$
  - *Else*
    - *If $(|a_j - R_c(s_i)| <= RT$*
      - *$R_u(s_i)=(R_c(s_i)*c+ a_j)/(c+1)$ where c is the number of previously trusted users(Their assigned reputation values are considered in the calculation of the current reputation of the service)*
    - Else
      - *Add a_j to the list L= $\{a_j \mid 0 < j < l\}$ Where l is the number of consumers.*
      - *If there exist an ordered $L_n \subseteq L$ where $L_n = \{a_r \mid 0 < r <= n, a_n - a_{1<=RT}\}$ where n is the agreement threshold and RT is the reputation threshold then*
        - *$R_u(s_i) = ((\sum_{r=1}^{n} a_r + R_c(s_i)*c)/(c+n)$*
        - *Remove L_n elements from the list*
  - *If Tfc_j =Null*
    - $Tf_j = 1 - \left|\frac{R_u(s_i)-a_j}{100}\right|/2$, *where Tf_j is the new trust factor for the consumer j.*
  - *Else*
    - $Tf_j = ((1 - \left|\frac{R_u(s_i)-a_j}{100}\right|/2) + Tfc_j)/2$

After the finishing of the access process of the service by the consumer, the consumer assigns a reputation value for the service and the reputation algorithm (Algorithm1) updates the reputation value of the accessed service based on this value. In addition, the algorithm takes into account the trust factor of the user, the trust factor threshold, the reputation threshold, and the agreement threshold which are assigned by the system. The trust factor of the user is assigned and updated by the system based on the behavior of the user by comparing his assigned reputation values and other users' reputation values. The algorithm provides an approach for updating the trust factor of the users too.

The algorithm starts by checking the trust factor of the consumer and based on this value and the trust factor threshold, the consumer reputation value can be considered or not. If the value of the trust factor is equal or greater than the value of the trust factor threshold and the reputation threshold is satisfied then the reputation value which is assigned by this consumer is considered and included in the calculation of the updated reputation for the service. Otherwise, if the value of the consumer trust factor is equal or greater than the value of the trust factor threshold but the reputation threshold is not satisfied then the consumer reputation is stored in a list *L*. After that, this list is checked to find a list $L_n \subseteq L$ of reputation values with cardinality equal to the agreement threshold *n* where the difference between the smallest and the largest value is less than or equal the reputation threshold. If this is the first time to assign a reputation value for the service then the value is assigned directly. Otherwise, the average of all the previous and current reputation is calculated to obtain the updated reputation value. If the value of the trust factor of the consumer does not satisfy the trust factor reputation, then this value is not considered in the calculation of the updated reputation. The trust factor of the user is updated based on the assigned reputation and the new reputation value that assigned to the service.

B. Discovery Approach

The discovery approach includes the functional and nonfunctional properties. The functional properties include the attributes that the service must have to perform the service and the nonfunctional properties present the QoS including the service reputation attribute.

Figure 2 shows a flowchart of the discovery approach. The input is the consumer functional and QoS requirements. The discovery unit receives discovery requests; it checks the functional and QoS matching in the service registry and returns a list of services that meet the consumer requirements.

The consumer then chooses a service from the returned list and invoke it. After the execution of the selected service, the consumer assigns a reputation value for the service. This reputation value is entered in the reputation module to update the reputation of the service and the trust factor of the consumer based on Algorithm 1. In addition, the trust factor of the consumer is updated based on the relation between her/his assigned reputation value and the updated service's reputation value.

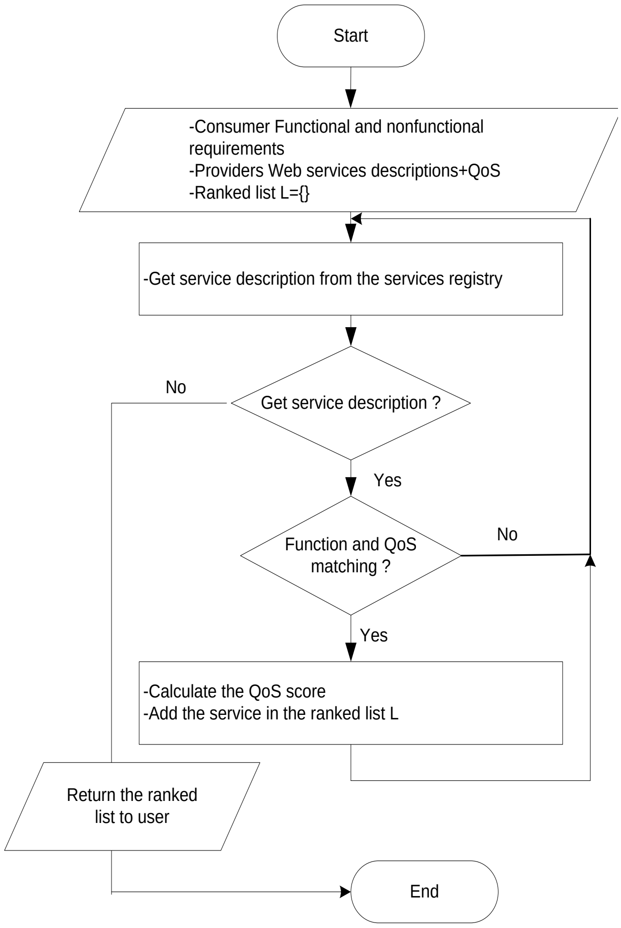

Fig.2. Web Services Selection Based on Reputation flowchart.

The calculation of QoS scores of services is performed by the equation below:

$$QoSScore = \frac{\sum_{i=1}^{n} QoS_i}{n} \quad (1)$$

Where *QoSScore* is the QoS score of a service and QoS_i is the normalized value of the quality of services attribute *i* of the service in which QoS_i is equal to one in case that this attribute is compatible with the corresponding attribute of the user and equal zero if it is incompatible, *n* is the total number of attribute of the service

IV. Experiment Results

The effectiveness of the proposed adaptive reputation approach is checked by implementing the discovery software including the proposed reputation algorithm. The effect of the trust factor threshold (TFT), agreement threshold (AT), and the reputation threshold (RT) is tested. In the case of the adaptive reputation approach, there are many cases based on the values of the different thresholds are tested. Table 1 shows the twenty consumers, the trust factor for each consumer, and her/his assigned reputation for the tested service.

Table 1. The consumers with their trust factor and their assigned reputation for the tested service.

Consumer	Trust Factor	Assigned Consumer Reputation
C1	0.8	85
C2	0.85	86
C3	0.8	85
C4	0.5	50
C5	0.84	69
C6	0.87	85
C7	0.8	86
C8	0.4	20
C9	0.8	85
C10	0.82	85
C11	0.8	84
C12	0.7	77
C13	0.82	70
C14	0.88	84
C15	0.77	82
C16	0.82	84
C17	0.5	60
C18	0.82	84
C19	0.88	85
C20	0.8	71

Case 1: In this case, the effect of the trust factor threshold is tested. Twenty different service requests in Table 1 are used. The agreement threshold is not considered and the reputation threshold is fixed at 10%. The service reputations updates are checked based on different values of the trust factor threshold. Figure 3 shows the reputation values assigned by the twenty consumers with TFT = None, 0.5, 0.7, 0.8, and 0.85. The results in the figure show that as the trust factor threshold is increased the updated reputation value is more precise and express the actual reputation of the service. When the TFT is not considered or low, the dishonest consumers affect negatively on the reputation value of the service.

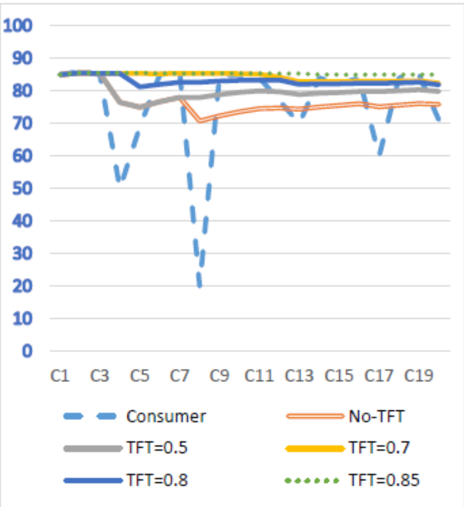

Fig.3. The reputation updated values in case of TFT= None, 0.5, 0.7, 0.8, and 0.85.

Case 2: In this case, the effect of the agreement threshold is tested where TFT is fixed to 0.7 and 0.8 and RT is fixed to 10. Figure 4 and 5 shows the updated reputation values with different values of the agreement threshold AT=0, 2, 3, and 4 where TFT=0.7 and 0.8 respectively. The results shows that as the value of the agreement threshold is increased, the results is more precise but the number of the included users is decreased. Therefore, if the service is accessed frequently, then this value can be assigned high. In case that the service is not accessed frequently, it is best to assign a small value for this threshold.

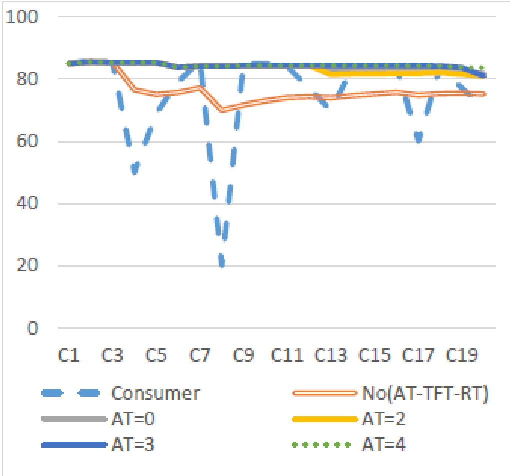

Fig. 4. The reputation updated values in case of AT= 0, 2, 3, and 4 with **TFT=0.7 and RT=10**.

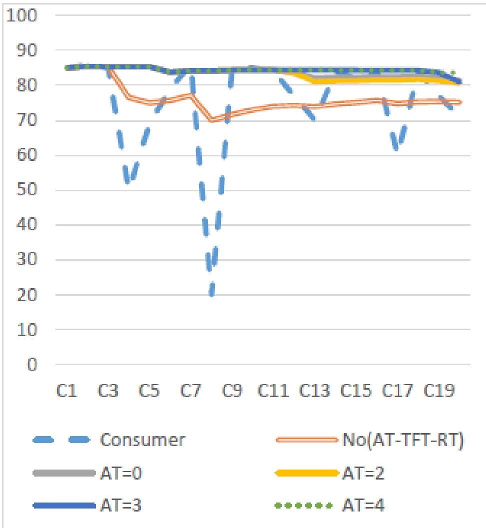

Fig. 5. The reputation updated values in case of AT= 0, 2, 3, and 4 with **TFT=0.8 and RT=10**.

Case 3: In this case, the effect of the reputation threshold is tested where TFT is fixed to 0.8 and the agreement threshold is AT= 0, 2, 3, and 4, but RT is changed to the value 5 instead of 10 in case 2 figure 5. Figure 6 shows the resulted reputation values in case of AT= 0, 2, 3, and 4 with TFT=0.8 and RT=5. The comparing between results in Figure 5 where RT=10 and Figure 6 where RT=5 shows that decreasing the reputation threshold increase the precision of the updated reputation.

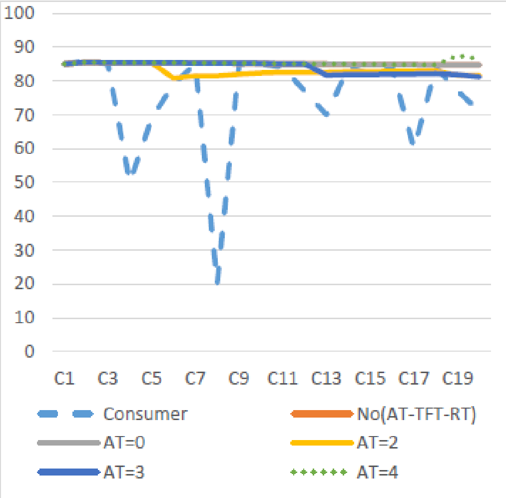

Fig. 6. The reputation updated values in case of AT= 0, 2, 3, and 4 with **TFT=0.8 and RT=5**.

V. Conclusion

The more precise calculation of the QoS attributes provide more relevant service for the consumer requests. In this paper, an adaptive reputation update approach is introduced to provide precise reputations for the services. The approach is based on three thresholds AT, TFT, and RT. The results shows the effect of each threshold on the precise of the reputation values. As a result, the choice of the value of these thresholds effects on the discovery process. Therefore, the returned list of nominated Web services are the most relevant to the consumer request if the reputation attribute is updated precisely. For future work, we will apply our approach on a bigger standard dataset. In addition, we will study the behavior of the users of all services and try to get a trust factor based on the profile data of the consumer.

References

[1] Mike P. Papazoglou and Willem-Jan van den Heuvel, Service oriented architectures: approaches, technologies and research issues, *The International Journal on Very Large Data Bases, 16(3)*:389–415, 2007.

[2] S. Nepal, W. Sherchan, J. Hunklinger, and A. Bouguettaya. A fuzzy trust management framework for Service Web, *Proc. IEEE International Conference on Web Services (ICWS)*, Florida, USA . 321–328, July 2010.

[3] Ziqiang Xu, Patrick Martin, Wendy Powley and Farhana Zulkernine. Reputation Enhanced QoS-based Web services Discovery, *Proc. IEEE International Conference on Web Services (ICWS).* UTAH, USA, vol., no., pp.249,256, 9-13 July 2007.

[4] Mohamed A. Serhani, Rachida Dssouli, Houari Sahraoui, Abdelghani Benharref, and M. E. Badidi. QoS Integration in Value Added Web Services. *In Proceeding of the Second International Conference On Innovations and Informal Technology* (IIT'05), September 2005.

[5] A. Josang, T. Bhuiyan, Y. Xu, and C. Cox. Combining trust and reputation management for Web-based services, *in Trust, Privacy and Security in Digital Business, vol. 5185 of Lecture Notes in Computer Science*, pp. 90–99, 2008.

[6] S. Zhao, G. Wu, Chen, and H. Chen. Reputation-aware service selection based on QOS similarity, *Journal of Networks*, vol. 6, no. 7, pp. 950–957, 2011.

[7] J. Al-Sharawneh, M. Williams, and D. Goldbaum. Web Service Reputation Prediction based on Customer Feedback Forecasting Model. *In Proceedings of the Fourteenth IEEE International EDOC-Conference*, 2010, pp. 33-40.

[8] Y. Nianhua, C. Xin, and Y. Huiqun. A reputation evaluation technique for web services, *International Journal of Security and Its Applications*, vol. 6, no. 2, pp. 329–334, 2012.

[9] Wang, S., Zheng, Z., Sun, Q., Zou, H., Yang, F.: Evaluating Feedback Ratings for Measuring Reputation of Web Services. *In Proceeding of the IEEE International* Conference *on Services Computing SCC'11*, pp. 192-199 (2011).

[10] Sathya, M., Swarnamugi, M., Dhavachelvan, P., Sureshkumar, G. Evaluation of qos based web- service selection techniques for service composition, *International Journal of Software Engineering (IJSE) IJSE* 1, 73–90 (2010).

[11] X. Fu, P. Zou, Y. Jiang, and Z. Shang. QoS consistency as basis of reputation measurement of web service, *in Proceedings of the 1st International Symposium on Data, Privacy, and E-Commerce (ISDPE '07*), pp. 391–396, IEEE, November 2007.

[12] K. Yue, X.-L.Wang, and A.-Y. Zhou. Underlying techniques for web services: a survey, *Journal of Software*, vol. 15, no. 3, pp. 428–442, 2004.

[13] Haiteng Zhang, Zhiqing Shao, Hong Zheng, and Jie Zhai. Web Service Reputation Evaluation Based on QoS Measurement, *The Scientific World Journal*, vol. 2014, Article ID 373902, 7 pages, 2014.

[14] S. Ran. A model for web services discovery with QoS, *in: ACM SIGecom Exchange, vol. 1, ACM Press*, New York, NY, USA, 2003, pp. 1–10.

[15] Gustavo Alonso, Fabio Casati, Harumi A. Kuno, and Vijay Machiraju. Web Services-Concepts, Architectures and Applications, *Data-Centric Systems and Applications.* Springer, 2004.

[16] Heather Kreger. Fulfilling the web services promise, *Communications of the ACM*, 46(6):29–ff, 2003.

[17] B Al-Shargabi, O Sabri, A Sabri: Toward a User Preferences Agent Based Web Service Composition. International Journal of Information Technology & Computer Science (IJITCS), volume 14(2), pp. 1-5, 2014.

Author's Profiles

Emad ELABD: PhD, Assistant professor, Department of Information Systems, Menoufia University, Egypt. He got his Ph.D. in the field of Web services compliance over high level specifications at LIRIS, University Lyon1, France, July 2011. He received bachelor ́s degrees in Electronic Engineering from Menoufia University, Egypt where he did his master ́s studies in computer science also. His research interests include Web services modeling and analysis with access control and time aspects, Web services (specification, composition), Semantic Web, and Information retrieval.