# Improving the Trustworthiness of Service QoS Information in Service-based Systems

Stephen S. Yau[1], Jing Huang[1] and Yin Yin[1]

[1] Information Assurance Center, and
[1] School of Computing, Informatics, and Decision System Engineering
Arizona State University
Tempe, AZ 85287-8809, USA
{ yau, Jing.Huang.5, yin.yin}@asu.edu

**Abstract.** Service-oriented architecture facilitates rapid development and management of large-scale distributed service-based systems (SBS), where new workflows are composed of available services. Besides services' capabilities, the qualities of services (QoS) also need to be considered in service composition in order to have high-quality workflows. The QoS profiles of services provided by their service providers may be inaccurate. In this paper, an approach is presented to improving the trustworthiness of the QoS information of services to facilitate the development of high-quality workflows in SBS. Our approach is based on identifying the deviation between the QoS profiles of the services claimed by their service providers and the QoS profiles determined by monitors and service user feedbacks. An example is given to illustrate the approach.

**Keywords:** quality of service (QoS), service QoS, improving trustworthiness, high-quality workflow, service-based system.

## 1 Introduction

To facilitate the development and management of large-scale distributed applications, service-oriented architecture (SOA) [1] has been adopted for many large-scale distributed applications, including e-commerce, health care, transportation, homeland security and military [2]. Systems developed based on SOA are called *service-based systems (SBS)*. The basic components of SBS are individual services, each of which provides certain capabilities. The workflow for implementing an application is composed of services whose dependency and communication relations are specified in the workflow. With

SOA, the development of application systems is to develop workflows composed of available services, instead of from scratch. During workflow composition, SBS developers first need to identify the capabilities required by the application systems, search for available services providing such capabilities, and use them to compose the workflows.

In this development process, besides services' capabilities, the qualities of services (QoS), like throughput, completion time, accuracy and security protection, also need to be considered in service composition in order to have high-quality workflows. Service-level agreement (SLA) [3] and protection-level agreement (PLA) [4, 5] are provided by service providers to claim their services' QoS in service contracts. In this paper, we will present an approach to improving the trustworthiness of information on the service QoS to facilitate the development of high-quality workflows in SBS. In this approach, service providers provide the information on their services' QoS profiles along with the services they provide. SBS developers use the QoS profiles to select services among the available services providing the needed functionality. Service providers may update the QoS profile of a service when there are changes in the software or underlying hardware supporting the service. We consider in the paper that the QoS profile of each service claimed by its service provider will include both performance and security.

However, the QoS profiles provided by service providers may be questionable since service providers may exaggerate the QoS profiles of their services to attract the interests of developers in using their services. Hence, the developers need to have more trustworthy QoS profiles of the services to make good decisions on selecting the services. Our approach will generate *more trustworthy QoS profiles* (*MTQP*) for services in service-based systems by identifying the deviation of service providers' claimed QoS profiles from the measured QoS profiles through the use of service monitors and service user feedbacks.

In the rest of the paper, we will first discuss the current state of the art on trust evaluation for information systems. In Section 3, we will introduce our overall approach, and the details of certain major steps are elaborated in Sections 4 to 7. An example to illustrate our approach will be given in Section 8. In Section 9, we will discuss future work needed for improving our approach.

## 2 Current State of the Art

Trust is fuzzy and dynamic in nature, and is a subjective notion describing the degree of belief about a particular entity's behavior [6, 7]. DeFigueiredo, et al., [8] investigated the human's intuitive understanding of trust to better understand the concept of trust in the cyber domain. The trustworthiness of

information has been studied in social networking, where peers collaborate together to create and share information [9-12]. The trustworthiness of the information has been evaluated based on the information's quality, the credibility on the information quality, and the pertinence [13, 14].

The measurement of trustworthiness of computer systems and networks has been considered at three levels: infrastructure, understanding, and policy [15]. The infrastructure level emphasizes the trustworthiness of the underlying system, i.e. hardware, software and the network. Remote attestation mechanism defined by Trusted Computing Group (TCG) [16] can provide users with more confidence in a remote cyber system environment. WS-Attestation [5] leverages TCG technologies in Web Services framework. At the understanding level, reputation systems are widely adopted to assist trust establishment [17-20]. While trust is a directional pairwise relationship, reputation is distilled about the information available from the individual's past behavior reported by those having interacted with it. For those agents who do not have direct interactions before, reputation is very important information to help them make trust decisions. To ensure the effectiveness of reputation evaluation mechanisms, dishonest feedback needs to be deleted. Statistical techniques are widely used to detect dishonest feedback, often with a presumed distribution of the collected ratings. Feedback may be screened out or given a weight based on their deviations from the majority opinion [20-22]. Such methods assume that a majority of the users are honest. Vu, et al. [23] used trust/distrust propagation and clustering techniques in assigning the weights to user reports. They made some assumptions on the properties of the service users and distribution of user ratings in the system. The efficiency of their algorithms is not clear. Jurca presented a novel application of reputation systems in SBS [24]. His mechanism targeted economic applications with attackers who aim at making money, but is fragile in hostile environments, where money is not a concern for the attackers. For critical systems, the attackers may try to sabotage the normal functioning of the system at any cost.

## 3 Our Approach

In this section, we will present the overview of our approach, including the participating entities and their relations of the SBS system and the mechanisms for generating *MTQP* of services in the SBS.

In our SBS with trust and reputation management, there are three types of entities: *service providers*, *service users* and the *administration*. The administration includes all administrative components, such as the *service directories*, *monitors* and *proxies*. Both service providers and service users can be benign or malicious. Therefore, in our approach, we will only trust the

administrative components which manage their reputation information, but do not fully trust service providers or users.

Service providers register their services with functionality meta data, claimed QoS and the access point of each of their services at service directories. Services should behave consistently with the registered information, regardless of the invoker, but their QoS may fluctuate around a stable value. Malicious service providers cheat on the QoS or even functional properties of their services either for economic benefits or for impeding system functionality, i.e. the experiences of service users. Some services are composed of other services in the SBS, and we call them *composite services*. Service $S_A$ is called a *partner service* of service $S_B$ if $S_A$ has invoked $S_B$ in any composite service or workflows. There are two types of service users: the end users and the partner services. Service monitors [25-30] are used to remotely monitor and report service QoS. Our approach does not require any specific proportion of services being monitored by a trusted monitor. Therefore, we have two categories of services in the SBS: those with a monitor, and those without. Fig. 1 shows an example of SBS with six services. Assume that services $S_3$, $S_4$ and $S_5$ are provided by the same service provider. The solid arrowheads in the figure represent a workflow in the SBS. The dashed arrowhead from service $S_1'$ to $S_2$ indicates that $S_1'$ is an alternate of service $S_1$ because they provide the same functionality. The services placed in a green rectangle are currently monitored by the same service monitor.

Service monitors periodically report the observed QoS of the services they are monitoring. Service users report the observed QoS of the services when they have used them. Since the QoS observations on delay, throughput, etc. need to be made automatically, we assume that the administration makes the observing and reporting code available for the users to submit feedback automatically. However, malicious users may tamper the reporting code and report dishonest QoS feedback for the services. Service proxies, which will be explained in Section 4, may capture some of the dishonest user feedback. Dishonest feedback and feedback from users who have submitted certain amount of dishonest feedback in the past will be deleted.

Using the information collected from monitors and users, we calculate two types of trust values for each service: *Mutual Trust Value (MTV) and Individual Trust Value (ITV)*. They reflect the accumulated reputation of the service provider in providing accurate QoS information for the service. MTV represents the accuracy of a service's QoS profile from the point of view of one of its partner services. It is based only on the partner service's experience and only represents the partner service's opinion. ITV represents the accuracy of an individual service's QoS profile considering its interactions with all partner services. It is either generated with the information from the monitor or aggregated from all its partner services' feedbacks. These two types of trust values are generated from two types of information sources and reflect the

trustworthiness of the QoS profile from a local view or a holistic view. In our approach, the *ITV* and *MTV* interact and affect each other. We evaluate a service's *MTV* from each of its partner services through feedbacks from that partner service. If the service is monitored by a monitor, we compute its *ITV* using the monitor's reports; otherwise, we compute its *ITV* by aggregating all the *MTV*s from all partner services. The trust values are applied on the latest claimed QoS profile of the service to generate *MTQP* for a service.

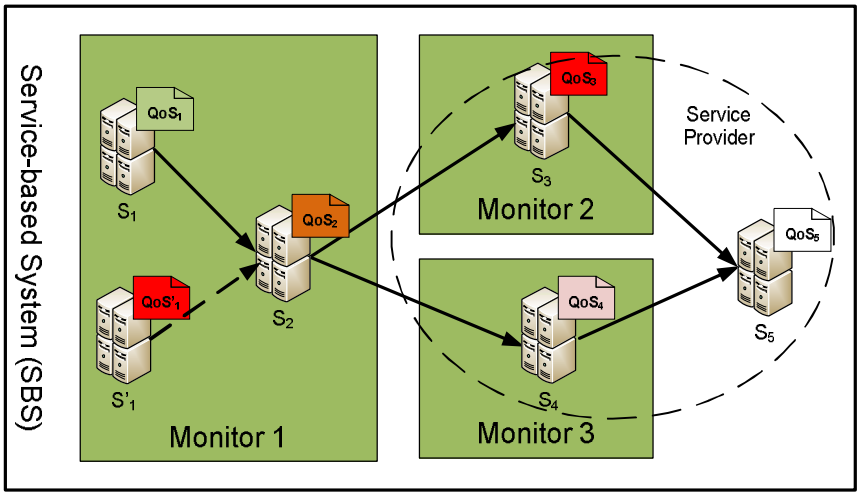


**Fig. 1.** An example SBS with service monitors.

Our approach can be summarized in the following four steps. The operations in Step 1)-3) are performed whenever there is new input information. Step 4) is performed when *MTQP* of a service is needed for service selection. Step 1) receives input from the service monitors and service users; the other steps use the output generated by the previous step.

1) Collect QoS feedback from service monitors and service users.
2) Filter dishonest user feedback for service proxies.
3) Calculate two types of trust values for each service: *MTV* and *ITV*.
4) When *MTQP* of a service is needed for service selection, it is computed using the latest claimed QoS profile of the service and either its *ITV* or one of its *MTV*.

Step 2) will be explained in Section 4. Sections 5 and 6 will give the details of Step 3) and Section 7 will elaborate Step 4).

## 4. Detection of Dishonest Feedback Using Service Proxies

Service proxies are special types of services hosted by the administration. They are registered in the service directory as regular third party services and are not recognizable to service users. However, a service proxy does not hold particular functionalities itself, but acts as a proxy between the user and another service. It provides the same interfaces as the third service, and when invoked, it will invoke that service to fulfill the functionality instead. However, the service users do not know how the service proxy provides the requested functionality, and will submit QoS feedback for the service proxy.

The purpose of using service proxies is to detect dishonest feedback without much overhead. When a user invokes a service proxy, it provides the opportunity to collect both the QoS of a service observed by the administration and the feedback submitted by the user for the service proxy. The difference between them should not exceed the small communication overhead between the user and the service proxy, plus some possible random noise. A large difference between them indicates dishonest feedback from the service user. The specific threshold is determined based on the QoS aspect and application scenario.

To allow dynamic binding in SBS, services with the same or similar functionality usually have similar interface. The administration can host one service proxy for each group of such services, and periodically change the service it invokes among the group of services without much development and configuration overhead. Since service proxies do not perform functionalities themselves, developing and hosting service proxies do not consume much effort and system resources. Therefore, it does not increase the system complexity much although the number of service proxies needed may grow with the number of types of services in the SBS.

It is noted that using service proxies to detect dishonest users is not based on any assumptions on the distribution of collected feedback or the percentage of dishonest users among all users.

## 5. Generation of Mutual Trust Value *(MTV)*

Because services' qualities are manifold, we define the *QoS profile of a service* as a vector with $n$ elements, each of which represents the QoS of the service on one of the n QoS aspects, such as accuracy, delay, and security. For a service $S$, let the QoS profile of the given service $S$ provided or claimed by its service provider be denoted by

$$cQoS = <cQ_1, cQ_2, ..., cQ_n>, \quad (1)$$

where $cQ_i$, is the *ith* QoS aspect provided by the service provider, $i = 1, 2, ..., n,$. Let the QoS profile based on the feedback of the *jth* partner service of $S$ be denoted by

$$fQoS_j = <fQ_{1j}, fQ_{2j}, ..., fQ_{nj}>, \quad (2)$$

where $fQ_i$, is the *ith* QoS based on the feedback of the partner service, $i = 1, 2, ..., n,$.

The mutual trust value (*MTV)* of $S$ from the *j*th partner service $S_j$ is denoted by

$$MTV_j = <mtv_{1j}, mtv_{2j}, ..., mtv_{nj}>, \quad (3)$$

where $mtv_{ij}$ is given by

$$mtv_{ij} = F_{t1}(fQ_{ij} - cQ_i), \quad (4)$$

and the function $F_{t1}$ satisfies the following three properties.

- **The value of $mtv_{ij}$ is in the range (-1, 1)**. The QoS $fQ_i$ based on the feedback of partner service $S_j$ may be better or worse than $cQ_i$. If $fQ_i$ is as good as $cQ_i$, $mtv_{ij}$ is equal to 0. A negative $mtv_{ij}$ means that the value of $fQ_i$ is smaller than that of $cQ_i$, and a positive $mtv_{ij}$ means larger.
- **For a constant $cQ_i$, the value of $mtv_{ij}$ increases as $fQ_i$ increases**. A larger QoS in the feedback will lead to a larger $mtv_{ij}$.
- **For a constant $cQ_i$, the value of $mtv_{ij}$ increases faster as $|fQ_i - cQ_i|$ increases, until asymptotically approaching the upper or lower bound**. When $fQ_i$ is closer to $cQ_i$, the value of $mtv_{ij}$ is insensitive to the difference $|fQ_i - cQ_i|$. This property is to model the normal variation of quality. First, it is difficult for service providers to accurately specify their services' QoS, which is usually an estimation based on historical data. Second, services' QoS is not stable and will vary in a small scale. On the other hand, the value of $mtv_{ij}$ will increase faster as the award or penalty for better or worse $fQ_i$.

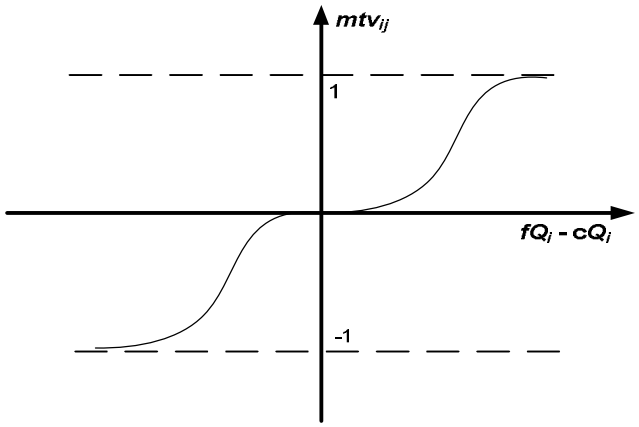


**Fig. 1**. The variation of $mtv_{ij}$ based on $fQ_i$ - c$Q_i$

Based on the above three properties, $mtv_{ij}$ can be expressed in Fig. 2 and (5).

$$F_{t1}(\mathrm{x}) = \frac{2}{1+\exp(-\mathrm{x}/\mu)} - 1 \tag{5}$$

where $\mu$ is a parameter which depends upon the range of a certain QoS aspect. This parameter can be used to tune the growing speed of the curve.

## 6. Generation of Individual Trust Value *(ITV)*

### 6.1 Generation of *ITV* with Monitors

For a service $S$ with a monitor, the QoS profile reported by its monitor is denoted by

$$mQoS = <\mathrm{m}Q_1, mQ_2, \ldots, mQ_n>, \tag{6}$$

where m$Q_i$, is the *ith QoS* reported by the monitor, $i = 1, 2, \ldots, n$,. The *ITV* of $S$ is denoted by

$$ITV = <itv_1, itv_2, \ldots, itv_n>, \tag{7}$$

where $itv_i$ is given by

$$itv_i = F_{t1}(mQ_i - cQ_i), \tag{8}$$

where the function $F_{t1}$ was discussed in Section 5.

### 6.2 Generation of *ITV* without Monitors

When there is no monitor monitoring the service *S*, we evaluate the service's *ITV* using all its *MTVs* between the service and its partners.

Assume that service *S* has interacted with *m* ($m \geq 1$) partner services and each of them may have a different *MTV* about the trustworthiness of *S*'s claimed QoS profile based on their own experiences. The *ITV* of *S* is denoted by

$$ITV = <itv_1, itv_2, \ldots, itv_n>, \tag{9}$$

where $itv_i$ is given by

$$itv_i = F_{ag}(mtv_{i1}, mtv_{i2}, \ldots, mtv_{im}), \tag{10}$$

where $F_{ag}$ depends on $mtv_{ij}$'s distribution. For example, if $mtv_{ij}$ has the uniform distribution, the simple average function is a good candidate for $F_{mtv}$. If $mtv_{ij}$ has normal distribution, $F_{mtv}$ should give higher weights for $mtv_{ij}$ close to the average of the means of all $mtv_{ij}$. The time duration between the last interaction between the service and its partner service that produces $mtv_{ij}$ can also be used as its weighting factor.

## 7. Generation of MTQP of Services

At the time of service selection, a service's *MTQP* can be derived from its latest claimed QoS profile and either its *ITV* or its *MTV*. We consider the following two types of applications. In the first application type, the QoS of a service is independently of the partner services it interacts with, and we use the service's *ITV* and its latest QoS profile claimed by its provider to generate its *MTQP*.

The *MTQP* of *S* can be computed as follows:

$$\begin{aligned} tQoS &= <tQ_1, tQ_2, \ldots, tQ_n> \\ &= <F_{t2}(cQ_1, itv_1), F_{t2}(cQ_2, itv_2) \ldots, F_{t2}(cQ_n, itv_n)> \end{aligned} \tag{11}$$

where $tQ_{i,}$ is the *ith* QoS generated by our approach, and the function $F_{t2}$ inverses the result of function $F_{t1}$ in (5), and is given by

$$F_{t2}(x, y) = x - \mu \cdot \log_e(\frac{1-y}{1+y}). \quad (12)$$

In the second application type, the QoS of a service is particularly sensitive to its partner service. In this case, we use the service's *MTV* to estimate its QoS profile instead of *ITV* because the *MTQP* of a service cannot be determined before choosing its partner service. If *S* is expected to be invoked by service $S_j$, we will use the *MTV* of *S* from $S_j$ and the latest claimed QoS profile of *S* to generate its *MTQP*.

The *MTQP* of *S* can be computed as follows:

$$tQoS = <tQ_1, tQ_2, \ldots, tQ_n>$$
$$= <F_{t2}(Q_1, mtv_{1j}), F_{t2}(Q_2, mtv_{2j}) \ldots, F_{t2}(Q_n, mtv_{nj})>. \quad (13)$$

## 8. An Example

Consider the example *SBS* given in Fig. 1, which has 6 services and 5 are monitored by monitors. In this example, we will consider only one QoS aspect, the completion time, because other QoS aspects can be handled similarly. We will generate the *MTQP*s of the six services, for the application type where services' QoS is independent of their partner services for composing workflows. Hence, we compute the *ITVs* for all the services.

The completion time of these services claimed by their service providers and reported by their monitors and partner services is given in Table 1.

**Table 1.** Computation results of the example.

| Service | $S_1$ | $S_1$' | $S_2$ | $S_3$ | $S_4$ | $S_5$ |
|---|---|---|---|---|---|---|
| Claimed completion time | 0.25 | 0.40 | 0.15 | 0.20 | 0.35 | 0.25 |
| Monitored completion time | 0.40 | 0.35 | 0.20 | 0.30 | 0.30 | |
| User feedback | | | | | | 0.15, 0.20 |
| ITV | 0.64 | -0.25 | 0.25 | 0.46 | -0.25 | -0.36 |
| More trustworthy completion time | 0.40 | 0.35 | 0.20 | 0.30 | 0.30 | 0.18 |

• **Generating *ITVs* for services monitored by service monitors**. First, we set the parameter $\mu$ = 0.10, assuming that a difference of 1 second or more

between the claimed and actual completion time is considered unacceptable. Using (5), we obtain the *ITV* of $S_1$

$$ITV_1 = F_{t1}(0.4\text{-}0.25) = \frac{2}{1+\exp\left(-\frac{(0.4-0.25)}{0.1}\right)} - 1 = 0.64 \quad (14)$$

Similarly, we find the *ITVs* of $S_1$ to $S_4$, which are also included in Table 1. Note that the negative trust values indicate worse QoS for the completion time because smaller completion time is more desirable.

• **Generating *ITVs* for services not monitored by service monitors**. For service $S_5$, assume that $S_3$ and $S_4$ have interacted with $S_5$ and provided their feedback on its completion time as 0.15 and 0.20, respectively. Using (5) we obtain the *MTVs* of $S_3$ to $S_5$ and $S_4$ to $S_5$ -0.46 and -0.25, respectively. We choose the average *MTV* as the service's *ITV*, which is -0.36.

• **Generating *MTQP*s**. The more trustworthy completion time of $S_1$ is computed using (13) and is given below:

$$F_{t2}(0.25, 0.64) = 0.25 - 0.1 \times \log_e\left(\frac{2}{0.64+1} - 1\right) = 0.4. \quad (15)$$

Note that the more trustworthy completion time of all the services, except $S_5$, is the same as the monitored completion time. However, if their service providers have updated the QoS profiles to claim a different completion time, the more trustworthy completion time will be different from the monitored value, but maintains the same *ITV* with regard to the new QoS profile, until new data is collected by the monitors and integrated in the *ITVs*.

Assume that $S_2$ and $S_5$ provide the same functionality. $S_2$ has shorter completion time based on the QoS profiles provided by their service providers, but the result of our approach indicates that $S_5$ is likely to have a shorter completion time, and hence $S_5$ is more desirable.

## 9. Discussions

In this paper, we have presented an approach to improving the trustworthiness of service QoS profiles in SBS to facilitate the development of high-quality workflows. In our approach, we assume that some services in the SBS are monitored by service monitors, but do not require that all the services are monitored. We will investigate how the proportion of monitored services and different ways of choosing services to be monitored affect the effectiveness of our approach.

Since service proxies cannot detect all the dishonest feedback submitted by service users, we will study the detection of dishonest feedback through inconsistencies between the QoS feedback of a composite service and that of its component services.

In the future, we will also evaluate the effectiveness of the more trustworthy QoS profiles and the techniques for detecting dishonest feedback through simulation and experiments.

## Acknowledgment

This work reported here is sponsored by National Science Foundation under Grant No. CCF-0725340. The authors would like to thank Dazhi Huang of Arizona State University for many valuable discussions of this research.

## References

1. IEEE Service-Oriented Architecture Standards, http://www.soa-standards.org/
2. Bartoletti, M., Degano, P., Ferrari, G. L.: Enforcing Secure Service Composition. In: Proc. of 18th IEEE Computer Security Foundations Workshop (CSFW), pp. 211-223 (2005)
3. Rajan, H., Hosamani, M.: Tisa: Towards Trustworthy Services in a Service-Oriented Architecture. In: IEEE Trans. on Services Computing, vol. 1(4), pp. 201-213 (2008)
4. Alam, M., Zhang, X., Nauman, M., Ali, T.: Behavioral Attestation for Web Services (BA4WS). In: Proc. of 2008 ACM Workshop on Secure Web Services, pp. 21-28 (2008)
5. Yoshihama, S., Ebringer, T., Nakamura, M., Munetoh, S., Maruyama, H.: WS-Attestation: Efficient and Fine-Grained Remote Attestation on Web Services. In: Proc. of 2005 IEEE Int'l Conf. on Web Services, pp. 750-757 (2005)
6. Chang, E. J., Hussain, F. K., Dillon, T. S.: Fuzzy Nature of Trust and Dynamic Trust Modeling in Service Oriented Environment. In: Proc. of 2005 Workshop on Secure Web Services, pp. 75-83 (2005)
7. Cook, K. S. (editor): Trust in Society. Russell Sage Foundation Series on Trust, vol. 2, New York (2003)
8. DeFigueiredo, D. B., Barr, E. T., Wu, S. F.: Trust Is in the Eye of the Beholder. In: Proc. of Int'l Conf. on Computational Science and Engineering, pp. 100-108 (2009)

9. Thuraisingham, B. M.: Trust Management in a Distributed Environment. In: Proc. of Annual IEEE Computer Software and Applications Conf., pp. 561-562 (2005)
10. Thuraisingham, B. M.: Assured Information Sharing between Trustworthy, Semi-trustworthy and Untrustworthy Coalition Partners. In: Proc. of the 4th Int'l Symp. on Info., Computer, and Comm. Security, Keynote (2009)
11. Hamlen, K. W., Thuraisingham, B. M.: Secure Peer-to-peer Networks for Trusted Collaboration. In: Proc. of Int'l Conf. on Collaborative Computing: Networking, Applications and Worksharing, pp. 58-63 (2007)
12. Layfield, R., Kantarcioglu, M., Thuraisingham, B. M.: Incentive and Trust Issues in Assured Information Sharing. In: Proc. of Int'l Conf. on Collaborative Computing: Networking, Applications and Worksharing, pp. 113-125 (2008)
13. Moturu, S. T., Yang, J., Liu, H.: Quantifying Utility and Trustworthiness for Advice Shared on Online Social Media. In: Proc. of Symp. on Social Intelligence and Networking, IEEE Int'l Conf. on Social Computing (2009)
14. Moturu, S. T., Liu, H.: Evaluating the Trustworthiness of Wikipedia Articles through Quality and Credibility. In: Proc. of 5th Int'l Symp. on Wikis and Open Collaboration, (2009)
15. Yang, S. J. H., Hsieh, J. S. F., Lan, B. C. W., Chung, J-Y.: Composition and Evaluation of Trustworthy Web Services. Int'l J. of Web and Grid Services, vol. 2(1), pp. 5-24 (2006)
16. Trusted Computing Group Specification Architecture Overview, Revision 1.2, https://www.trustedcomputinggroup.org/downloads/TCG_1_0_Architecture_Overview.pdf
17. Emekci, F., Sahin, O. D., Agrawal, D., Abbadi, A. E.: A Peer-to-Peer Framework for Web Service Discovery with Ranking. In: Proc. of IEEE Int'l Conf. on Web Services, pp. 192-199 (2004)
18. Kalepu, S., Krishnaswamy, S., Loke, S. W.: Reputation = F(User Ranking, Compliance, Verity). In: Proc. of IEEE Int'l Conf. on Web Services, pp. 200-207 (2004)
19. Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J., Sheng, Q. Z.: Web Engineering: Quality Driven Web Service Composition. In: Proc. of Int'l World Wide Web Conf., pp. 411-421 (2003)
20. Malik, Z., Bouguettaya, A.: RATEWeb: Reputation Assessment for Trust Establishment among Web services. The VLDB J., vol.18(4), pp. 885-911 (2009)
21. Dellarocas, C.: Immunizing Online Reputation Reporting Systems against Unfair Ratings and Discriminatory Behavior. In: Proc. of 2nd ACM Conf. on Electronic Commerce, pp. 150-157 (2000)

22. Whitby, A., Josang, A., Indulska, J.: Filtering Out Unfair Ratings in Bayesian Reputation Systems. The Icfain J. of Management Research, Vol. 4(2), pp. 48-64 (2005)
23. Vu, L.-H., Hauswirth, M., Aberer, K.: QoS-based Service Selection and Ranking with Trust and Reputation Management. In: Proc. of OTM'05, R. Meersman and Z. Tari (Eds.), LNCS 3760, pp. 466-483 (2005)
24. Jurca, R.: Truthful reputation mechanisms for online systems. PhD dissertation, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland (2007)
25. Yau, S. S., Ye, N., Sarjoughian, H., Huang, D.: Developing Service-Based Software Systems with QoS Monitoring and Adaptation. In: Proc. of 12th Int'l Workshop on Future Trends of Distributed Computing Systems, pp. 74-80 (2008)
26. Yau, S. S., Yin, Y., An, H. G.: An Adaptive Model for Tradeoff between Service Performance and Security in Service-Based Environments. In: Proc. of Int'l Conf. on Web Services, pp. 287-294 (2009)
27. Yau, S. S., Ye, N., Sarjoughian, H., Huang, D., Roontiva, A., Baydogan, M., Muqsith, M.: Towards Development of Adaptive Service-Based Software Systems. In: IEEE Trans. on Services Computing, vol. 99, pp. 247-260 (2009)
28. Raimondi, F., Skene, J., Emmerich, W.: Efficient Online Monitoring of Web-Service SLAs. In: Proc. of the 16th ACM SIGSOFT Int'l Symp. on Foundations of Software Engineering (2008)
29. Ezenwoye, O., Sadjadi, S. M.: RobustBPEL2: Transparent Autonomization in Business Processes through Dynamic Proxies. In: Proc. of the 8th Int'l Symp. on Autonomous Decentralized Systems (2007)
30. Moser, O., Rosenberg, F., Dustdar, S.: Non-Intrusive Monitoring and Service Sdaptation for WS-BPEL. In: Proc. of the 17th Int'l Conf. on World Wide Web (2008)