Web Service Selection Based on Trust

Ioan Alfred LETIA, Octavian POP
TechnicalUniversity of Cluj-Napoca
str.Constantin Daicoviciu nr.15, RO-400020 Cluj-Napoca

***Abstract*—Service discovery and selection is an important task in semantic web services. Besides the problem of selecting web services according to client requests, we are intending to implement a mechanism to select web services based on the reviews and feedback users provide. Another aspect we are dealing with is related to false ratings, clients can rate web services incorrectly, related to service behavior, influencing future service selection processes. Rating service providers can help providing a more accurate service selection, al least when a web service has a reduced number of ratings and false ratings can influence the selection result.**

***Index Terms*—Reputation, Semantic Web Services, Service Selection, Trust, Web Services.**

I. INTRODUCTION

Current approaches on web service selection follow some general ideas, differences occur in the algorithm used. The existing directions in web service trust representation are organizes as follows [1]:

- Centralized architectures where the information regarding trust is kept in a central location;
- Distributed architectures – in this situations the members of the system have to cooperate to manage the reputation;

Solutions for trust/reputation management are constructed mainly on centralized architectures; service discovery mechanisms are enhanced with capabilities to retrieve and update information related to service attributes. A general solution adopted consists in selecting from service repository all services that met client functional requirement; from this list select the top M services ranked with the highest reputation. Reputation is computed based on web service rated attributes and user preferences.

Within this article we are intending to propose architecture for service discovery based on customers review and feedback. Service discovery based on trust is a subject widely studied ([2], [3], [4], [5], [6], [7], [8], [9]) in the field of semantic web services, but in other approaches [2] client reviews are taken for granted, or it is considered that lying is uninterested due to economic incentives [10]. In some cases these approaches are not feasible, so customer reviews have to be validated against an agreement between service provider and service consumer to avoid false positive or false negative ratings. There are cases when a service provider wants to reduce the reputation of another service provider giving a bad feedback even if the service quality is better. The agreement between the two parties is established using a service a Service Agreement Manager and it is supervised by a broker that will intercept calls from a customers. We develop a broker based approach for web service selection and recommendation. The broker will act as the service provider, intercepting service requests from client and responding for these requests.

The feedback is divided into two categories:

- Related to broker supervised attributes
- Client subjective feedback

Also, we consider that keeping a rating for the service provider, not only for published services, will help to provide a more accurate service selection.

The remaining of this article is organized as follows. Section 2 outlines the related research conducted in the field of Web service selection, web service trust and reputation. Our proposed architecture is presented in section 3, with the main components involved. Section 4 presents our solution for web service trust estimation, including service provider trust. Section 5 presents an illustrative example. We conclude with section 6 with a summary of our work and possible future research in this direction.

II. RELATED WORK

In this section we summarize some of the existing directions in web service selection process, service trust issues, providing an overview of some of the work as a context of our research.

A. System architecture

Broker based architectures was previously adopted for agent based systems, such as the one proposed in [11]. For each web service there is a broker, as an independent component from client and service provider side. Using a broker based architecture we can avoid situations where reviews are provided by clients, which being malicious or friendly, can supply false positive or false negative reviews. Other approaches extend the existing architecture based on the Universal Description, Discovery and Integration (UDDI) ([12-13]), in [2] is proposed a Web service discovery model that extends UDDI to include QoS information, a reputation management system and a discovery agent to perform service discovery.

1. Web Service trust

Trust has an important role in Semantic Web. The Web is an open environment, “anyone can say anything about anything”, that’s why we have to consider the trust on the service providers and service instances. According to the architecture adopted, different approaches that cope with unfair services and unfair clients have been proposed.

Existing solutions, according to [10], rely on the following techniques:

- A trusted monitor intercepts the message exchange between the client and the provider and outputs an estimate of the delivered QoS;

- The monitoring code runs on the provider side, as part of the service middleware. The monitoring layer intercepts messages addressed to or originating from the provider, and estimates the delivered QoS;
- A trusted party periodically probes the service and outputs performance metrics.

Other avoids this issue in different ways, for example in [10] lying is considered uninteresting, relying on economic incentives. In most examples the above case is not valid. The trust is computed based on the reviews provided by clients or some advisor agents. Advisor agents are used especially in agent based environments. In [14] are presented some methods, with their advantages and disadvantages, to handle unfair ratings, we briefly present them below:

- *Cluster Filtering* – copes with unfair ratings, it takes into account preference similarity between buyer and advisor agent. One problem about this approach is that it does not handle unfair low ratings
- *The beta reputation system (BRS)* – estimates reputation of seller agent using a probabilistic method, based on the beta probability density function. The seller reputation is estimated by propagating ratings provided by multiple advisor agents
- BRS was extended to *Iterative filtering*. This method filter out ratings that are not in the majority amongst other ones. This method id effective when the majority of ratings are fair.

The innovation we brought consists in combining two different kinds of review:

- The one provided directly by client, as a subjective perspective of the web service – that cannot be evaluated as incorrect;
- Review provided by the broker, as a trusted third party, so the effect of lying clients is reduced.

Transferring trust from web service to service provider will provide the necessary information to evaluate the provider trust, which can be used to estimate the trust on other services published by this provider.

Gil et al. [15-16] propose to make a distinction between entity trust and content trust. Entity trust is concerned with the evaluation of trust users can have on the information provider, content trust reflecting the trust users have on the actual information. We are intending to transfer trust between these two layers, being able to evaluate the expected trust on a service provider using the feedback users provide for his related web services.

2. Review management

Service selection consists in recommending an ordered list of services based on the reviews that exists related to each web service. Different solutions work with homogeneous list of attributes related to a web service, this means that all services have to have the same list of attributes. Our solution works with web services that are not homogeneous.

The reputations systems adopt two general solution, as depicted from [2], [11]:

- reviews are time dependent – latest reviews are more important that the older ones;
- reviews are not time dependent, all having the same importance;

Review time dependence is implemented in different ways, for example [2] suggests either the use of a damping function to model the reduction of reputation over time, or applying an aging factor for the reputation score to each of the ratings for a service, thus newer ratings are more significant than the older ones.

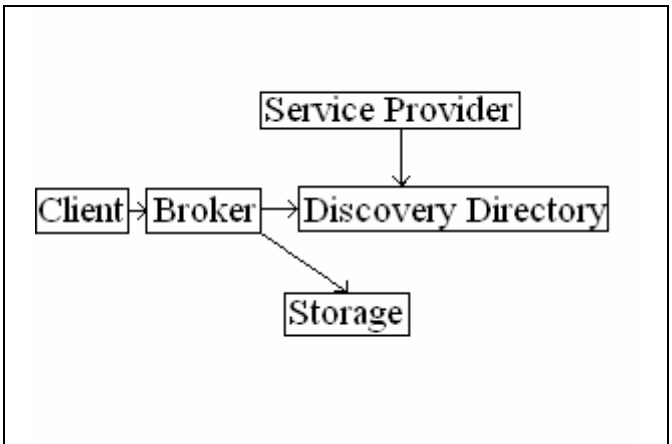
We combine both directions; trust function is computed as a product between an aggregation function and a weight function. If reviews are not time independent, the weight function is constant (G(X)=1), if reviews are time dependent we use a Gaussian distribution, adjusting the deviation as needed to take into account older ratings.

III. ARCHITECTURE OVERVIEW

Our approach is based on the existence of a broker between service client and web service provider. This approach has the advantage that the broker is considered an independent component that will rate truthfully web service usages. On the other hand it will be difficult that for each web service or for a group of web services to develop and to maintain a broker, taking into consideration the increasing number of web services.

Client requests are passed to a Broker that will perform the service query, searching for services from discovery directory according to client specifications. Top results are returned to client, which will interact with service provider to establish an agreement (Service Agreement) regarding the properties and their values that service should respect.

This agreement is then passed to a Service Agreement Manager that will provide a feedback according to service behavior.



.

Figure 1. System Architecture.

The broker will provide the feedback for the measurable parameters, to avoid false rating. Being a third party component, we will assume that is fair so we exclude the possibility of incorrect rating, even though the mechanism

will be more complicated.

Trust being considered as a subjective expectation, like in [17], the algorithm we implement for estimating trust gathers reviews from users, as a subjective feedback and data provided by the broker, which we consider objectively collected data.

For the broker to be able to rate services based on his measurements needs a set of rules and parameters, which will be fetched from a storage system or will be build in the proxy layer that is in front of the actual web service.

Let's take the example of response time web service attribute. For the broker to be able to rated this service feature we need two parameters representing the minimum accepted response time $\min_{acc}$, a maximum accepted response time $\max_{acc}$ and a rule, in this case a function that will evaluate the service behavior. If we select the following parameters: $\min_{acc} = 0$, $\max_{acc} = 2.1$ and the rating function to be $f(x) = e^{-x}$. A web service will be rated with values as presented in the following figure, Fig. 1:

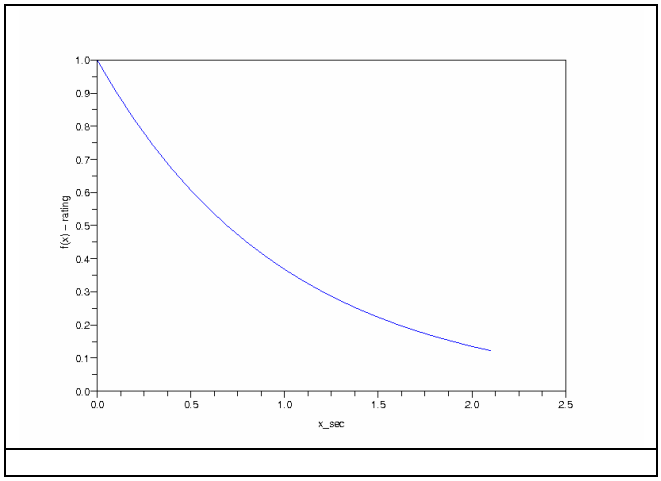


Figure 2. Sample rating function.

IV. TRUST EVALUATION

We developed our theory based on associations between web services and their features and between web service providers and web services.

Associations are defined as a topple of the form (*ar, ae*), where *ae* is an entity which can be everything that can be trusted, *ar* is an association relation associated to the corresponding entity. We define, in terms of associations, a web service as a subset of A, A is the set of all associations. A service provider is also defined as a subset of A, where *er* is a web service related and *ar* is the relation between service provider and a web service (e.q. published, referenced).

Using some translation principles we can transfer trust from web services to web service provider, for example based on related resources principle if two resources are related then trust can be transferred from one resource to the other, from web service to service provider.

1. Service provider trust

Web service provider trust is estimated based on trust users have on his related web services depicted from his associations. Before presenting the algorithm for service provider trust we provide some preliminary details and assumptions.

Each user has a specified trust on all associations that will be involved in his queries, formally this can be expressed through the following function: $t : U, A \to [0,1]$, where U is the set of all users, A is the set of all associations. Provider and web service trust is estimated as a real number between 0 and 1, 0 representing no trust and 1 for full trust.

For each web service we can estimate the trust users can expect based on previous feedback, this aspect is covered in more detail in the following paragraph.

The algorithm is presented below:

a. Provide a list L of association relations and an optional list F of features
b. Select the related web services *ws* based on the provided list of relations:

$$WS = \{ws \mid (ar, ws) \in A \wedge ar \in L\} \quad (1)$$

c. For all *ws* from WS, estimate web service trust based on *F*, if this list of features is missing, consider all *ws* attributes.
$\forall ws \in WS$ compute $t(ws)$ - web service trust. Using these values build the web service trust estimate:

$$WSTE = \{t(ar) \cdot t(ws) \mid (ar, ws) \in A\} \quad (2)$$

t represents the trust on the association rule, which is client specific.

Service provider trust can be evaluated as the min/max/average value over web service trust estimates.

2. Web Service trust

Within this paragraph we present the method used to evaluate service trust, our assumption is that all associations related to a particular web service are known, as a subset of A.

When a client is searching for a web service it have to provide a list of service attributes that have to be considered when the broker is searching for web services. Web service attributes correspond to the first term from associations. For each element from list, also, he can specify a weight for the corresponding attribute, for example if $L = \{a_1, a_2, a_3\}$ is the list of service attributes that have to be considered and $P = \{1, 1, 0.4\}$ are the weights of attributes in trust evaluation, these weight have as default values set to 1.

The broker will select from repository ratings corresponding to the specified attributes, if an attribute is missing, it will be ignored when trust function is evaluated setting the corresponding weight, from P to 0.

For each web service, the repository contains the feedback users provide. This is formally considered as a set of attributes, or service features, A, where $\|A\| = n$, for each feature a_i there are n_{a_i} values.

In this paragraph we aggregate feedback for attributes a_i

to evaluate the service trust.

Trust evaluation consists in estimating the trust users can expect based on the ratings form previous web service usages. Service manager maintains a list of features/attributes that clients expressed an opinion on. For each attribute the values are between 0 and 1, 0 meaning no trust and 1 for full trust.

The set A_i of features is composed by aggregating two types of attributes:

- Objective attributes (e.g. response time) – attributes that can be measured and evaluated exactly.
- Subjective attributes – that express the user opinion from a specific point of view regarding the web service (ex. usability).

Service trust is computes as a combination of the two components:

$$f(x) = a \cdot s(\bar{x} \times \bar{y}) + (1-a)o(\bar{x} \times (1-\bar{y})), \quad (3)$$

Where:

- $\bar{x}$ - is the set of rated attributes from a particular user;
- $s(\bar{x})$ - is the estimate of the trust based on subjective features
- $o(\bar{x})$ - is the estimate of the trust based on objective features

$$y = \begin{bmatrix} 1 \\ 1 \\ . \\ 0 \\ 0 \end{bmatrix}$$

$$y_i = \begin{cases} 1, \text{if } a_i \text{ objective attribute} \\ 0, \text{if } a_i \text{ subjective attribute} \end{cases}$$

- a is the weight objective attributes have on the overall trust;
- $o(\bar{x}) = (\bar{x} \times P^T) \cdot \frac{1}{\|P\|}$, where P is a vector that express the weight of each attribute. Web service clients can express their preferred attributes and the degree of preference.

Using the above defined function, f, we provide some alternative solutions for evaluating overall web service trust:

- We can consider that the feedback users provide is not time dependent; in this case older ratings have the same weight as the new ones; web service trust is computed as the mean value of all ratings:

$$t(\bar{x}_i) = \frac{1}{n}\sum_{i=1}^{n} f(\bar{x}_i) \quad (4)$$

- Otherwise, we can consider that newer ratings are more accurate than older ones, so the trust function is computed as a weighted sum. The weights used are values from a Gaussian distribution, with $m = 0$. The number of ratings taken into consideration depends on m.

$$t(\bar{x}_i) = \frac{\sum_{i=1}^{n} f(\bar{x}_i) \cdot G(time(\bar{x}_i))}{\sum_{i=1}^{n} G(time(\bar{x}_i))} \quad (5)$$

$time(x)$ represents the time moment when rating x was provided, G is the Gaussian distribution function, as in the Fig. 3, for different values of deviation s :

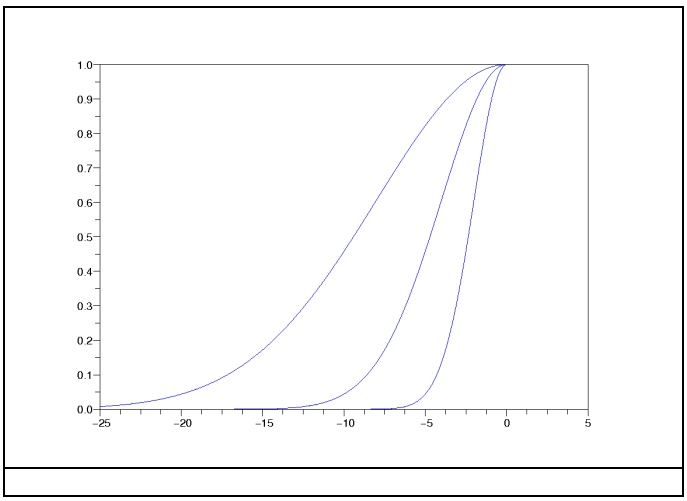


Figure 3. Different weight functions generated modifying s

For each feedback s, the trust function $f(s)$ is multiplied with the Gaussian computed in $time_s$, time when the feedback was supplied. Current time corresponds to $x = 0$, if we want to consider older values we have to increase s .

At a closer look, the first alternative is a particular case for the second one, the trust function becomes:

$$t(s) = \frac{\sum_{i=1}^{n} f(s) \cdot r(time_s)}{\sum_{i=1}^{n} r(time_s)} \quad (4)$$

$r(time_s)$ is a constant function, as in the following image, or a semi-Gaussian, presented in Fig 3.

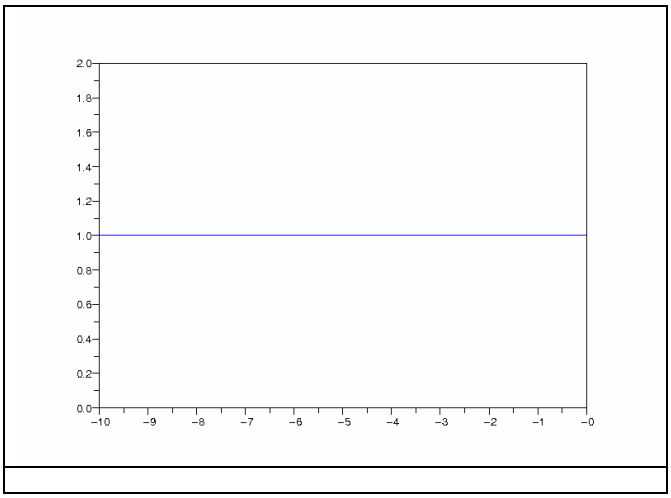


Figure 4. Constant weight function.

3. Integrate service provider trust

There are situations when clients select web service based on service provider reputation.

If a published web service is not rated by many clients, their reviews cannot be considered relevant, in this situation taking into consideration the service provider reputation can help in estimating the web service trust.

Service provider trust can be used either as a service feature or if the web service requires more client feedback for a more accurate trust estimate then service provider can substitute web service trust. In both cases we presume that the service provider has more than one service published.

V. ILLUSTRATIVE EXAMPLE

Let A = {responseTime, availability, | usability, aesthetic}, and the client is interested in a service easy to handle (high usability) and with high availability- the service need to be accessed with no restriction (support service), then p would be:

$$p = [0 \quad 1 \quad 1 \quad 0]$$

And the collected feedback is presented in the following table:

TABLE I. CLIENT FEEDBACK FOR A WEB SERVICE

A	1	2	3	4	5	6	7	8
response Time	0.5	0.6	0.8	0.4	0.6	0.5	0.8	0.9
availability	0.7	0.8	0.8	0.6	0.7	0.7	0.5	0.4
usability	0.6	0.6	0.5	0.4	0.5	1	1	1
aesthetics	0.5	0.6	0.5	0.5	0.5	1	1	1

$n = 8$ - number of users ratings

$$f(s) = \frac{1}{n}\sum_{i=1}^{n} f(\bar{x}_i) = \frac{1}{n}\sum_{i=1}^{n}(a \cdot o(\bar{x}_i) + b \cdot s(\bar{x}_i));$$

Let's consider that the last 3 ratings are false positive, attributes rated by client are all 1, full trust.

First we calculate the trust function without the last 3 ratings and then will consider all ratings.

The following table shows trust estimation based on subjective components based on the feedback from Table II**,** using only the first five ratings, which we consider to be correct. The usability and aesthetics attributes are false positively rated in the last three ratings sets.

TABLE I I. SUBJECTIVE TRUST ESTIMATES

i	1	2	3	4	5
s_i	$\frac{1}{\sqrt{2}}0.7$	$\frac{1}{\sqrt{2}}0.8$	$\frac{1}{\sqrt{2}}0.8$	$\frac{1}{\sqrt{2}}0.6$	$\frac{1}{\sqrt{2}}0.7$

The following table shows trust estimation based on objective components based on the feedback from Table III

TABLE I I I. OBJECTIVE TRUST ESTIMATES

i	1	2	3	4	5
o_i	$\frac{1}{\sqrt{2}}0.6$	$\frac{1}{\sqrt{2}}0.6$	$\frac{1}{\sqrt{2}}0.5$	$\frac{1}{\sqrt{2}}0.4$	$\frac{1}{\sqrt{2}}0.5$

If the objective attributes have the same weight as the subjective ones, then $a = b = 0.5$, then

$$f(s) = \frac{1}{5}\frac{1}{2}\frac{1}{\sqrt{2}}6.2 = 0.43$$

In the second case, with al the ratings taken into consideration, the trust value would be:

$$f(s) = \frac{1}{8}\frac{1}{2}\frac{1}{\sqrt{2}}10.5 = 0.46$$

The difference between the two cases is 0.03, if 37% of ratings are malicious.

This value represents the service trust computed based on the provided feedback, without taking into consideration the service provider trust.

VI. CONCLUSION

A In this paper we provide a solution to web service selection problem when clients can be malicious, reducing the impact of false ratings. Our method for service provider trust estimation can help to provide a more accurate service selection, if also can be used in other situations; we intend to use it in service composition approaches. An innovative aspect we introduced in service selection consists in estimating service provider trust, which can be used to increase the accuracy in predicting service quality. Also, our approach is reducing the effect of incorrect ratings by combining client feedback with trusted measurements and also giving the user the possibility to specify his preferred service features that would be considered and a weight specifying the importance of each attribute.

REFERENCES

[1] J. Cardoso. Approaches to developing semantic web services," International Journal of Computer Science, vol. 1, no. 1, 2006.

[2] Z. Xu, W. P. Patrick Martin, and F. Zulkernine, Reputation-enhanced QoS-based web services discovery," International Conference on Web Services, vol. 1, pp. 249-256, 2007,ICSW Web Service Indexing and Discovery,Salt Lake City, Utah, USA.

[3] E. M. Maximilien and M. P. Singh, Conceptual model of web service reputation, ACM SIGMOD, vol. 31, pp. 36{41, 2002.

[4] C. Makris, Y. Panagis, E. Sakkopoulos, and A. Tsakalidis, Efficient and adaptive discovery techniques of web services handling large data sets,The Journal of Systems and Software, Elsevier, vol. 79, pp. 480-495, 2006.

[5] K. Karta, An investigation on personalized collaborative filtering for web service selection,Partial fulfilment of the requirements for the

Honours Programme of the School of Computer Science and Software Engineering, The University of Western Australia, 2005.

[6] L.-H. Vu, M. Hauswirth, and K. Aberer. QoS-based service selection and ranking with trust and reputation management. OTM Conferences (1), ser. Lecture Note in Computer Science, R. Meersman, Z. Tari M.-S. Hacid, J. Mylopoulos, B. Pernici, Özalp Babaoglu and Hans-Arno Jacobsen and Joseph P. Loyall and Michael Kifer and Stefano Spaccapietra, E vol. 3760. Springer, 2005, pp. 466-483. Available: http://dblp.utrier.de/db/conf/otm/otm2005-1c.html#VuHA05

[7] E. M. Maximilien and M. P. Singh, Toward autonomic web services trust and selection, in ICSOC '04: Proceedings of the 2nd international conference on Service oriented computing. New York, NY, USA: ACM, 2004, pp. 212-221

[8] Y. Liu, A. H. Ngu, and L. Zeng, QoS Computation and Policing in DynamicWeb Service Selection, 2004, Proceedings of the 13th International World Wide Web Conference.

[9] L.-H. Vu, M. Hauswirth, and K. Aberer, Towards P2P-based Semantic Web Service Discovery with QoS Support, in Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2006, vol. 3812, pp. 18-31.

[10] R. Jurca, B. Faltings, and W. Binder, Reliable QoS monitoring based on client feedback, in WWW '07: Proceedings of the 16th International Conference on World Wide Web. New York, NY, USA: ACM Press, 2007, pp. 1003-1012. Available: http://portal.acm.org/citation.cfm?id=1242572.1242708

[11] E. M. Maximilien and M. P. Singh, Multi-agent system for dynamic web services selection, AAMAS, 2005.

[12] M. Vallee, F. Ramparany, and L. Vercouter, A multi-agent system for dynamic service composition in ambient intelligence environments, The 3rd International Conference on Pervasive Computing (PERVASIVE 2005), 2005.

[13] Y. Makripoulias, C. Makris, Y. Panagis, E. Sakkopoulos, P. Adamopoulou, M. Lytras, and A. Tsakalidis, Towards ubiquitous computing with quality of web service support.

[14] J. Zhang and R. Cohen, Trusting advice from other buyers in e-marketplaces: the problem of unfair ratings, in the 8th International Conference on Electronic commerce, vol. 156. ACM, 2006, pp. 225-234.

[15] Y. Gil and D. Artz, Towards content trust of web resources, Web Semantic., vol. 5, no. 4, pp. 227-239, 2007.

[16] M. Keidl and A. Kemper, Towards context aware adaptable web services," in WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters. New York, NY, USA: ACM, 2004, pp. 55-65.

[17] D. Artz and Y. Gil, A survey of trust in computer science and the semantic web, Web Semantic., vol. 5, no. 2, pp. 58-71, 2007.