A Trust Model for Services in Federated Platforms

Francisco Nieto

A Trust Model for Services in Federated Platforms

Francisco Javier Nieto

ATOS Research and Innovation, Atos Origin, Capuchinos de Basurto 6,
48013 Bilbao, Spain
Francisco.nieto@atosresearch.eu

Abstract. Web services are a powerful tool for executing functionalities using third party applications and they are widely used in business processes. For this reason, it is necessary to complement traditional security solutions by adding soft security mechanisms, which take care of trust, in order to determine whether a web service and its provider are performing as they should. An extensible model for a trust evaluation system is presented for this purpose. It determines some parameters important in enterprise and Future Internet environments and it defines how to calculate the perceived trust by applying a three round algorithm with fuzzy logic. It exploits services semantics and it takes into account last updates about the service, consistency rules based on the semantic relationships between aspects and other specific calculations for each parameter. Semantics are used as well for sharing information with other platforms and federations for improving interoperability in distributed environments.

Keywords: trust; model; security; web services

1 Introduction

Nowadays, web services are a powerful tool for accessing to functionalities offered by third parties, as a way to externalize the provision of certain operations which need to be integrated in complex systems.

Because of the importance of the functionalities to be used and the sensitive information exchanged in the process when several enterprises have to interoperate, those systems which perform an intensive use of external web services need to provide secure mechanisms which guarantee that the interactions are done as expected and that the information is exchanged in a secure way.

Analysis done about these kind of systems, like [1], reveal that there are hard security mechanisms (focused on encryption and access control) an soft security mechanisms (more focused on social aspects, such as trust and reputation).

There are not so many solutions for covering soft security mechanisms customized for SOA environments and which can be easily integrated with access control mechanisms to be applied in enterprise contexts. In addition, even if there are existing approaches related to reputation, it is hard to find solutions which cover more aspects related to the trust of web services and which are able to exploit services semantics for giving accurate results. The presented approach aims at exploiting the semantics

and, at the same time, covering a lot of aspects which are related to trustworthiness in services while improving the robustness of the model by maintaining consistency of the data used.

In this context, according to this approach, it can be said that the trust on the service is seen as the belief in the reliability, truth and capability of the service.

The paper is structured as follows: section 2 provides a vision of the related work in the area, while Section 3 describes the main motivation and ideas in which the approach is based. Section 4 will describe the areas and aspects in the conceptual model and Section 5 presents how trust is evaluated. Finally, Section 6 presents a set of conclusions and future work.

2 State of the Art

There are several publications [1][2][3] which analyze the reputation topic and related approaches, They provide classification schemes and information about how to evaluate reputation and trust, but they are quite focused only in users' ratings.

They also include analysis of online traditional models for reputation (like eBay and Amazon) and other based on these ones with concrete improvements, such as NICE [4] or Sporas and Histos [5].

There are many ways to calculate the reputation associated to an agent or to a service, based on the ratings received from users. Approaches like [6] are focused on applying probabilistic theory by means of concrete distribution functions (as they are applied to data of the same nature), while other like RATEWeb [7] use weighted averages, but also predict the reputation of service providers by using a Hidden Markov Model when ratings are not readily available.

Most of the solutions proposed are focused only on a single rating obtained from users while only a few of them, like RATEWeb, take into account SLAs fulfillment and QoS, but also expressed as users' ratings and not measuring them directly or using other inputs for calculating a meaningful initial trust value.

In addition, only a few solutions, like REGRET [8], use ontologies as a way to identify derived aspects to be used when calculating reputation (for example, in the case of a seller, aspects like deliver date, price and product quality). The approach presented in [9] also exploits semantics, but it is focused on trust for actions performed by entities and it is not oriented to services.

There are some approaches which present fuzzy models, such as Afras [10]. They are quite useful for defining rules according to the fuzzy values and they increase the usability of the model, as users will understand the measures obtained much better without any reference for comparing. But robustness is delegated in the way to calculate trust and concrete filters, instead of controlling the data involved in a meaningful way, amending it when inconsistencies arise.

Finally, no approaches were found with the aim at solving sharing information with other platforms in a seamless way, taking into account federations as well.

3 Principles and Objectives

The work in [1] defines a set of requirements which should be covered by a trust model. The approach presented in this paper aims at fulfilling these requirements in the best way possible, providing a robust solution against attacks which calculates accurate values for long-term performance and which weights aspects in a fair way guaranteeing smoothness.

Moreover, the presented trust model goes beyond other existing approaches by looking for completeness, focusing not only on users' rating and its associated reputation, but also on general aspects and capabilities of the service, as a mean to evaluate service potential as well.

Since it may be necessary to customize the model, the presented approach offers the possibility to modify it, adding or modifying aspects and parameters. Besides, in a wide environment where enterprises are heterogeneous and are grouped in different ways, it is necessary to provide a mechanism which make possible to share information as a way to facilitate interoperability in scenarios where it is necessary to involve external platforms and services for executing business processes.

Finally, it is necessary to provide a mechanism which guarantees robustness of the model, not only based on the way to evaluate the trust, but based on controlling the most important asset of the model: the information on which everything is based.

For doing so, this solution aims at exploiting the full potential of the semantics associated to web services, as a mean to share information, evaluate the trust and avoid malicious attacks.

4 The Conceptual Model

The first step to create the model was to determine which concepts it should contain. For doing so, the SOA context and other trust models were analyzed, in order to know which aspects were meaningful for web services trust. In addition, an innovation game was carried out with some Atos Origin consultants. This game required adding notes to a picture of a big tree representing aspects which consultants take into account when thinking about the trustworthiness of software, indicating the importance of each aspect. At the end, some of these concepts were adapted to the SOA context and others were included in the model as defined originally.

The model was completed by adding some concepts related to trust, reputation and other SOA related concepts, such as Quality of Service (QoS) parameters.

The result is a model oriented to determine the trustworthiness of a web service which is divided in four main areas, as presented in Figure 1. The result obtained from applying the model is a value representing how much we can trust on a service, as a non-functional property, which can be used automatically together with other properties for ranking candidates when performing a service discovery, for instance.

The main aspects contained in each area are described in the following sections. Although some of them have to be obtained from the service provider and introduced manually, most of them can be obtained automatically by accessing to the service

description and existing Service Level Agreements (SLAs), by requesting information to other platforms and federations or by monitoring the interactions with the service.

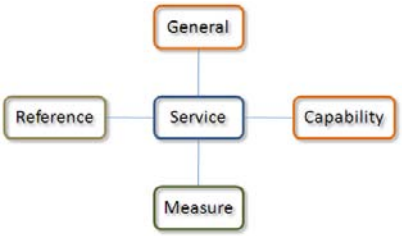

Fig. 1. High level view of the Conceptual Model.

4.1 General Area

The General area is about a set of aspects which do not represent the behavior of the service itself, but those general characteristics of the service related to how it has been developed, its sustainability and how it is maintained, as shown in Figure 2.

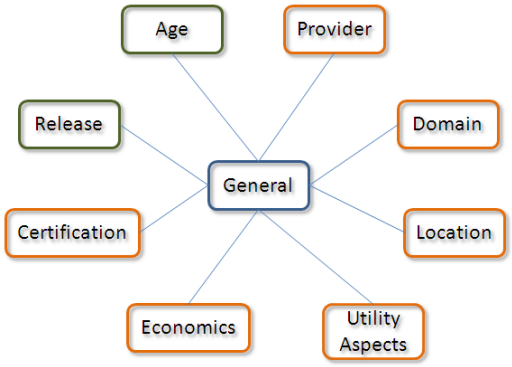

Fig. 2. General area of the Conceptual Model.

These aspects are quite static, as it is not usual that they change, but some of them are important for determining whether the web service can guarantee a good operation, such as the aspect 'Provider' which represents the brand developing the service and evaluates as well the support to users when there is any problem.

Another aspect which is important is 'Certification' which evaluates whether there are any authority certifying that the development of the service is secure (such as the Common Criteria certification [11]) and that standardized processes are followed.

'Location' evaluates if the service is located in countries with specific regulation for protecting data and if the country provides good network infrastructures for accessing the service. The 'Release' aspect represents the effectiveness of the last version of the service and with which frequency it is updated. It gives an idea of the maintenance and improvement of the service, although it depends on other aspects for

determining whether this is a positive or negative indication (e.g. too many bugs on previous releases).

The concept 'Utility Aspects' is interesting in the context of enterprise interoperability, as it evaluates the service with respect to its capacity to be used as a utility (for enterprise interoperability or enterprise collaboration) and to its usage in Virtual Organizations which carry out complex business processes.

While the 'Age' aspect gives an idea of how long the service has been working, 'Domain' determines the applicability of the service in different business domains, or its specialization in concrete domains only.

Finally, 'Economics' provide information about the revenue model and available payment methods, which may represent the sustainability of the service.

Most of this information will be introduced in the system by the system administrator (at least, for the first time). Even if for some aspects we need to rely on the information given by the service provider, most of that information can be validated, such as certifications, which can be validated with the corresponding authority, in order to guarantee the reliability on the model.

4.2 Capability Area

This area gathers information about WHAT the service can do and HOW it can do it. It represents what the service provider claims about the service. This information is expected not to vary too much during the service lifetime unless the service design and/or implementation changes. It is clearly divided in two main aspects: functionalities offered and non functional properties (NFP), as shown in Figure 3.

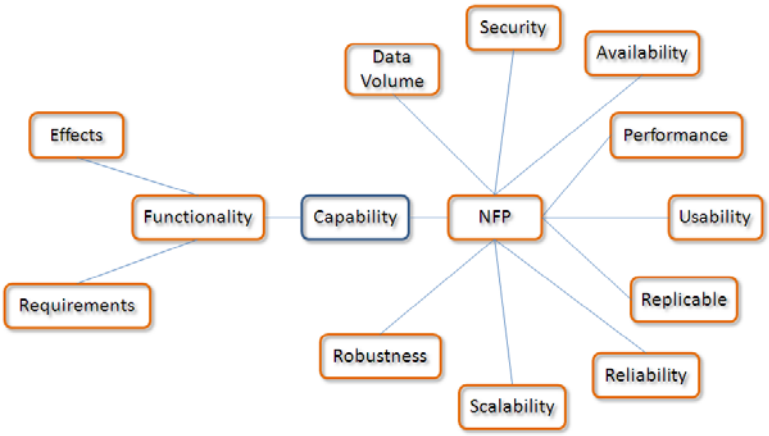

Fig. 3. Capability area of the Conceptual Model.

The 'Functionality' aspect is divided in two aspects. While 'Requirements' represents the dependencies of the service (in the form of pre-conditions and assumptions) and the final requirements (the post-conditions of the service), 'Effects' represents the functionalities by what they do in the context where they are executed. This information is used for evaluating whether the web service may have too many

requirements and whether it offers a lot of functionalities to users. As semantics of the web service are exploited in these aspects, it can be compared with other services which are similar or which belong to the same domain.

In the other hand, 'NFP' represents those aspects which are related to the claimed QoS. Some of these aspects can be compared later to the real QoS measures obtained from interactions with the service.

'Data Volume' refers to the amount of information exchanged between the service and the service consumer, for determining whether too much information is used or not, in comparison with other similar services. The aspect 'Availability' is focused in the percentage of time the service is available in comparison to other similar services.

The aspect 'Robustness' evaluates the percentage of failures expected from the service and its capability to react when there are problems (i.e. checking compliance with WS-Transaction [12]). Instead, 'Reliability' is more oriented to the trustworthiness in the delivery and reception of messages thanks to specifications such as WS-Reliability [13].

While 'Performance' evaluates if the service claims to offer good performance with respect to other services, 'Scalability' evaluates if the service maintain performance when requests and data volume increase, comparing with other services. 'Replicable' evaluates whether the service can be copied to several servers or if it can be executed in a distributed way, having a potential impact in performance, scalability, robustness and availability.

'Usability' evaluates whether it is simple or not to access to the web service. It is determined in terms of the signature complexity and the existence of semantic description for the operations, enabling automatic discovery and invocations.

Finally, 'Security' evaluates the kind of mechanisms offered by the service for guaranteeing a secure interaction. This covers aspects such as the existence of digital certificates and trust agreements, and the strength of encryption algorithms applied.

The service description (syntactical and semantic) and SLAs are the best sources of information for gathering data about the aspects presented here.

4.3 Measure Area

This area is the most dynamic one, as it evaluates the information which is continuously gathered about interactions with the service. Each time the service is invoked, it is monitored and analyzed according to the aspects presented in Figure 4.

Most of the aspects are directly related with those included in the 'NFP' aspect of the 'Capability' area. When evaluating these aspects, they are compared to the measured values for similar services as well as to any existing agreements (SLAs, etc.) for determining whether any agreement is violated.

'Response Time' represents the measure related to performance. It compares the real response time of the service with other services response time. Similarly, 'Data Volume' evaluates the amount of data exchanged between the service and service consumer, by comparing it to other similar services providing the same functionality.

The usage of the service is evaluated by means of 'Requests Number', so the interpretation is that good services are used by many users.

The aspect called ‘Scalability’ is measured in terms of response time variation when the number of requests increases and response time variation when the data volume grows, and then it is compared with other similar services.

While the ‘Availability’ aspect determines the percentage of errors because of failures accessing the service, ‘Robustness’ determines the percentage of errors because of malfunctioning of the web service, and which could not be controlled (fault messages are controlled errors). While the first one is related to the hardware and network infrastructures, the second is directly related to the service design and implementation. They are also compared with the values measured for other services.

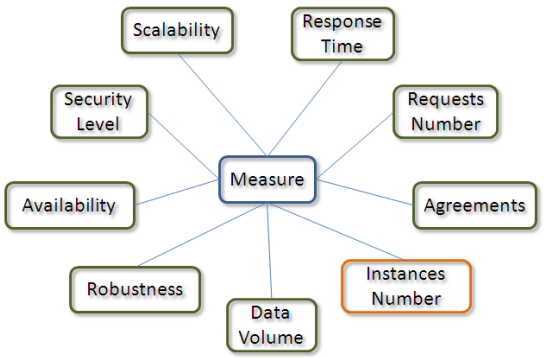

Fig. 4. Measure area of the Conceptual Model.

‘Instances Number’ represents the number of copies available of the service, so the calls received can be sent to different servers, balancing the load of the web service.

An important aspect is ‘Security Level’ which evaluates the mechanisms used for securing the message exchange. It includes the digital certificates used as well as the encryption mechanisms applied in comparison with other services and those mechanisms technically available but not used.

Finally, the aspect ‘Agreements’ evaluates whether the service is fulfilling its commitments by checking the measured values with existing SLAs and Trust Agreements (if any). It has a very important impact in trustworthiness.

All the aspects covered by this area can be evaluated by using the information which can be automatically gathered by monitoring the services, so it does not require manual provision of data.

4.4 Reference Area

This area represents the so called reputation of the web service. It aims at gathering external information about the service behavior and third parties’ opinions about the service according to the aspects presented in Figure 5.

This information is used in order to determine the trust in the service as seen by others with their own experience and sources of information.

The 'News' aspect refers to those news published which are related somehow to the web service or the service provider, as they may be a source of information about the service behavior and, about the way the service provider manages its business.

'Forums' is another source of opinions about the service and the service provider. Users may use forums to report problems or to ask help about a service.

The aspect 'Trusted Feedback' represents the aggregation of users' opinions about the service. They can rate the web service according to their experience using it. It will also depend on how much a user is trusted and the level of expertise with services (as consumers without experience using services may have negative opinions because of their lack of knowledge instead of because real problems in the service).

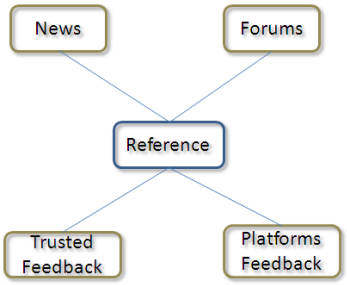

Fig. 5. Reference area of the Conceptual Model.

Finally, 'Platforms Feedback' represents the reputation of the service obtained from the aggregation of experiences from other platforms which may provide their rating for the service or the concrete measurement of some of the aspects, usually as part of a federation. It depends on the level of trust in the platforms sharing their information for avoiding problems with malicious platforms and for giving preference to those platforms which collaborate actively with the one evaluating the service.

In this case, all the information depends on external sources. In the case of 'News' and 'Forums', an administrator should introduce the information in the system for its evaluation. In the case of 'Trusted Feedback', users have to rate the service and that rating can be obtained from a web-based form. For 'Platforms Feedback', next section presents an ontology used for sharing information between platforms.

4.5 The Model as an Ontology

As the presented approach wants to exploit semantics as much as possible, all the model is expressed as an ontology, defining all the main concepts (such as 'Area', 'Aspect' and 'Parameter) and instantiating them according to the aspects presented. It includes concepts for describing a 'Federation' and the entities related to it, focusing on the trust shared between them.

The ontology has been defined using WSML in order to facilitate its development, its usage with semantic services (as part of it could be used in services descriptions based on WSMO) and the creation of mediators whenever necessary. The ontology is presented in Figure 6, with the main concepts defined. Concrete aspects and parameters would be instances of the concepts presented in the ontology.

Those concepts related to Trust Concept represent areas, aspects and parameters which are taken into account when determining the trust of a service. They represent a hierarchy for organizing those aspects used. While some are atomic (such as Trust Parameter), others depend on the aggregation of more Trust Concepts (such as Trust Area and Trust Aspect). In its higher level, a group of Trust Concepts is a Trust Model, which will represent the way a Trust Provider evaluates trustworthiness.

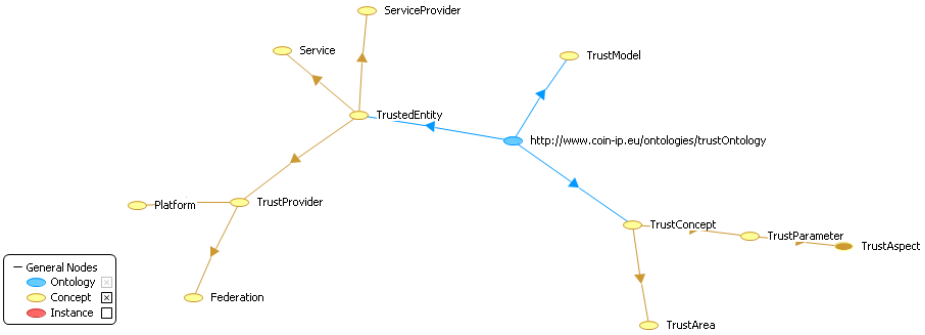

Fig. 6. Concepts of the Trust Ontology.

On the other hand, concepts related to Trusted Entity represent entities which take part somehow in a Federation, a group where information about trust is shared. There are entities which will have a trust level assigned (Trusted Entities, which can be evaluated) and there will be entities which provide trust evaluations (Trust Providers).

One of the advantages obtained is that the information can be shared between different platforms and federations easily, even enabling the possibility to define mediators whenever necessary (in case of interactions between heterogeneous platforms). Each Trust Provider will generate a trust evaluation for those known Trusted Entities, publishing certain information for others, which can map Trust Models between platforms and re-evaluate trustworthiness when necessary or desired.

Another advantage is the possibility to extend and modify the model in an easy way. New parameters, aspects and areas can be defined, assigning to them a weight representing their importance where they are allocated, so the rest of weights can be adapted. This allows administrators to customize the way trust is calculated.

Finally, semantics will be used to relate those aspects with some kind of relationship, meaning that a change in the value of one aspect will affect another one or that their values change in a similar way at the same time. These relationships will be used for the trust evaluation, as explained in the next section. In the ontology, this is modeled as three slots in Trust Parameter, which represent parameters which affect to other parameter, parameters affected by the instantiated parameter and parameters whose value is just related with the value of other parameters.

5 Trust Evaluation

Once aspects and parameters are defined, it is necessary to determine how so much information can be aggregated in order to provide a result which represents how much a web service can be trusted.

As it is necessary a way to normalize somehow the value of each aspect (so all the results are expressed in the same metric) a set of linguistic terms (and their corresponding fuzzy sets) have been defined (see Figure 7), so all the values obtained will be based on those terms and their membership functions. This is appropriate, as users will understand easily the meaning of the trust evaluation and, moreover, it will be possible to apply fuzzy logic theory and aggregate aspects in a meaningful way. Besides, it facilitates statistical analysis of all the aspects and parameters in the model, in order to find relationships between them.

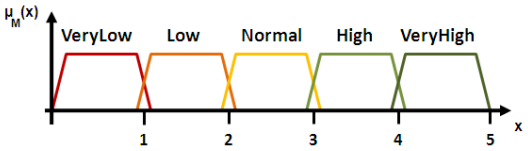

Fig. 7. Linguistic terms and their membership function.

The idea is to allow users and other platforms to access to the evaluation of concrete aspects facilitating interoperability and, moreover, to use those values to improve the model robustness and analyze consistency between aspects.

Each time any aspect in the model is updated, the whole trust evaluation will be calculated. This is done in a three-round process, which first calculates aspects, then amends the results according to consistency rules and, finally, aggregates all the values with a weighted mean which adapts weights according to the context.

5.1 First Round – Evaluate Each Aspect in the Model

As explained before, each aspect is represented by a value between 0 and 5 which determines how good is the aspect but, how is this value obtained for each aspect?

The way to calculate each value will depend on its nature and in what we can do with it. Given the space limitations of this paper, only the general calculation types are presented with some examples, and not how each different aspect is calculated.

Some aspects, such as response time and data volume, allow predicting next values from previous ones, so techniques like **simple exponential smoothing** are applied for determining the future expected value, which are compared with the prediction for other services which are similar in order to see how good obtained predictions are.

Other aspects can be evaluated by analyzing the **tendency** with respect to some variables (such as with scalability, using response time and data volume). In this case, it is possible to 'virtually' generate a graphic representing the service behavior and determine the tendency by calculating the slope of the curve in a concrete time.

There are other aspects which may depend on a context which is out of the control of the platform. For instance, with 'Location', the quality and price of network

infrastructures available, existing regulation about data protection and even resources spent for pursuing fraud are important inputs which are represented with **fuzzy sets** and combined with **fuzzy operations**.

Those aspects which are related to reputation using users' rating and other platforms as input are evaluated by applying the solution proposed in RATEWeb [7] because of its compatibility with the available information. Moreover, it can be used to predict reputation when users' feedback is not readily available, which is an expected situation when they have no good incentives.

Finally, whilst some aspects will be as simple as calculating a percentage and determine how good it is with respect to other services (such as in the case of robustness), others will be as complicated as determining the fuzzy set of a trust parameter by applying **statistical hypothesis tests**, like Student's t test for pair differences, which are combined later with fuzzy operators (such as in 'Release', where t test is used to evaluate the effectiveness of a new release by comparing old values of robustness and response time with new ones, obtained after updating the service).

5.2 Second Round – Amend Evaluations with Consistency Rules

As one of the key requirements for a trust model is to provide a mechanism guaranteeing robustness, the proposed model defines a way to check the information gathered for the evaluation in order to identify malicious manipulations of data.

This mechanism is built based on an analysis of the relationships between the aspects in the model. For example, the publication of new releases of the service may be good for the robustness of the service, as the number of errors is expected to decrease. But, it may happen that there are many releases in a short period of time because the service implementation is not good enough and the service experiences too many errors.

A fuzzy associative matrix is built according to the identified relationships, so a set of rules is defined. An example of rule, according to the previous example, would be: "*IF Robustness IS VeryLow THEN DecreaseRelease*". This way, it is possible to avoid that a malicious service provider tries to increase its trust level by publishing many releases without really doing meaningful modifications in the service.

There are two kinds of consistency rules for this purpose:

- *Basic rules*, as a result of the aspects analysis, especially for those relationships cataloged as cause-effect;
- *User defined rules*, as a customized configuration based on users' experience, requirements and desired constraints.

After the calculation of the value corresponding to each aspect in the model, the fuzzy rules are executed, in order to correct any incoherence in the aspects values. If one rule is activated because of an inconsistency, the value of one or more of the aspects will be amended, giving it an expected value defined in the rule.

5.3 Third Round – Aggregate Evaluations

Due to the heterogeneity of the aspects included in the model, probabilistic based approaches do not have too much sense. Since each aspect may be independent from others, that they measure different things and that they have different importance for the trust calculation, using a weighted average is the best solution. As aspects are grouped in areas, these areas are seen as blocks for calculating the trust, so each area is calculated first and, finally, the global value is calculated.

Equation (1) is used for calculating the trust level associated for each area and for the model in general, where factor *k* represents the weight to be applied, while *T* represents the trust level obtained for the aspect *i* in the model. The value of *T* will be always between 0 and 5 because the values are normalized with those fuzzy sets already mentioned.

$$T_B = \frac{\sum_{i=1}^{n} k_i * T_i}{\sum_{i=1}^{n} k_i} \quad (1)$$

Although each aspect has a weight associated, there are certain cases where other aspects are somehow related to it, and thus need to be taken into account for refining that weight, so the factor *k* is not exactly the original weight assigned to each parameter.

One of the factors to take into account is when an aspect has been updated. Although it can be considered a cognitive bias, it is quite representative in a SOA context, as last updates of the service are the most representative of the service behavior. This is because good results in old requests do not guarantee good results during the next requests (for example, the availability and response time could decrease because of problems with the servers and networks). The conclusion is that it is necessary to increase the weight of those aspects which have been updated recently.

The other factor to have in mind is the effect of each aspect in the service behavior. Even if some aspects are not updated recently, they might have a lot of importance in the way the service behaves. 'Release' is a good example, as a good robustness is directly related to the last updates in the design and implementation. A service which was not updated recently may obtain a low rating in the 'Release' aspect but, if the service is performing fine, that value can be increased, as it was really important. In this case, we can say that 'Robustness' (in 'Measure' area) and 'Release' (in 'General' area) are closely related and so we annotate it semantically. The semantic distance between two aspects will determine how much they affect each other.

According to the described principles, factor *k* will be calculated by applying (2), where *w* represents the original weight for aspect *i*, *t* represents the factor related to time and *D* the factor related to the semantic distance with last aspects updated.

$$k_i = w_i * t_i * D_i \quad (2)$$

The factor related to time (*t*) should give more importance to newest values and decrease the weight of old values smoothly. For calculating the effect of the time in the weight, all the aspects of the same block are ordered in a list according to the last time they were updated. Two aspects will be in the same position if they were updated

at the same time, so the classification is some kind of grouping of aspects according to the time when they were updated. Then, (3) is applied, since the curve it produces is similar to an exponential smoothing, which is the effect to be produced in the weights of the parameters. In this equation *p* is the position of the aspect *i* in the list.

$$t_i = e^{1/p_i} - 1 \quad (3)$$

Finally, the factor related to the importance of an aspect in the behavior of the service (*D*) should increase a bit the weight of those aspects which are more relevant, but should not penalize those aspects which may be more independent. For doing so, (4) is used, where *min(d)* represents the minimum distance between an updated aspect and aspect *i*. The distance is obtained thanks to the semantic relationship declared in the instances of the ontology. The reason to apply equation (4) is that it performs an exponential smoothing as well, but its effect is softer and the difference between first values and next ones is not so pronounced.

$$D_i = \left(\frac{1}{\min(d_i) * Ln(10)} \right) + 1 \quad (4)$$

As recently updated aspects are rewarded when calculating the factor *t*, in this case, we will consider that *min(d)=1* for most recently updated aspects and for those aspects which are directly related to them (as a cause, effect or normal relationship).

5.4 Initial Calculation

When a service is published in a platform, there is no information about the service behavior. The only information available is the service description and other information claimed by the service provider about its web service. This means that the first time the trust for a given service is calculated, it has to be a slightly different.

It is not possible to use information from 'Measure', as it will not be available (although it could be gathered with some invocations for testing purposes). For that reason, the calculation is focused mainly in the information gathered in 'General' and 'Capability' areas, which will give us a good idea about the potential of the service.

More information can be gathered by means of the 'Reference' area but, as it will not be possible to validate the information received (being weaker against malicious manipulations), its weight will be lower.

Having all these considerations in mind, the recommended weights to be used is 40 for 'Capability' and for 'General', while 20 for 'Reference' and 0 for 'Measure'.

6 Conclusions and Future Work

Given the importance of trust in those systems that perform an intensive use of web services, the presented approach provides a wide and complete conceptual model for determining the trust associated to a web service.

Although reputation is an important part for determining the trust level of a service, there are other aspects which may be even more important, such as the measures done during direct interaction with services and the service capabilities.

When determining the real weight of an aspect, its importance is not only determined by the weight provided by users, but also by when it was updated and by the effect it really has in the web service, which may not be visible at a first sight. It is also important to share information between platforms in an easy way (as it is used for determining reputation as well), always trying to guarantee a good level of robustness for the used model. For doing so, exploiting semantics is a very good tool, as they are becoming a widely used solution in the SOA context and allow modeling knowledge.

Future work will be focused in improving the model by analyzing all the relationships between the aspects defined in the model through deep statistical analysis, so they can be fully exploited.

After that, it will be possible to extend the approach in order to determine the trust level in Cloud Computing environments. Moreover it will be possible to determine the trust level which can be achieved by a business process (implemented as service compositions) by analyzing the trust related to the services which will be used. It will require mechanisms for analyzing the workflow and determining how the presented approach can be applied in this context.

Acknowledgments. This research was supported by the COIN project (http://www.coin-ip.eu/) and has been partly funded by the European Commission's IST priority of the 7th Framework Programme under contract number 216256.

References

1. Jøsang, A., Ismail, R., and Boyd, C.: A survey of trust and reputation systems for online service provision. Decis. Support Syst. 43, 2, pp. 618-644 (Mar. 2007)
2. Q. Zhang, T. Yu, K. Irwin: A classification scheme for trust functions in reputation-based trust management, in: Proceedings of ISWC Workshop on Trust, Security, and Reputation on the Semantic Web (2004)
3. Sabater, J. and Sierra, C.: Review on Computational Trust and Reputation Models. Artif. Intell. Rev. 24, 1, pp. 33-60 (Sep. 2005)
4. S. Lee, R. Sherwood, and B. Bhattacharjee.: Cooperative Peer Groups in NICE. In IEEE Infocom, San Francisco, CA, Apr. (2003)
5. Zacharia, G.: Collaborative Reputation Mechanisms for Online Communities. Master's thesis, Massachusetts Institute of Technology (1999)
6. Li, L., Wang, Y., and Lim, E.: Trust-Oriented Composite Service Selection and Discovery. In Proceedings of the 7th international Joint Conference on Service-Oriented Computing (Stockholm, November 24 - 27, 2009). L. Baresi, C. Chi, and J. Suzuki, Eds. Lecture Notes In Computer Science, vol. 5900. Springer-Verlag, Berlin, Heidelberg, pp. 50-67(2009)
7. Malik, Z., Akbar, I., and Bouguettaya, A.: Web Services Reputation Assessment Using a Hidden Markov Model. In Proceedings of the 7th international Joint Conference on Service-Oriented Computing (Stockholm, November 24 - 27, 2009). L. Baresi, C. Chi, and J. Suzuki, Eds. Lecture Notes In Computer Science, vol. 5900. Springer-Verlag, Berlin, Heidelberg, pp. 576-591 (2009)
8. Sabater, J. and Sierra, C.: REGRET: A reputation model for gregarious societies. In Proceedings of the 4th Int. Workshop on Deception, Fraud and Trust in Agent Societies, in the 5th Int. Conference on Autonomous Agents (AGENTS'01), pp. 61-69, Montreal (2001)
9. Huang, J. and Fox, M. S.: An ontology of trust: formal semantics and transitivity. In Proceedings of the 8th international conference on Electronic commerce: The new e-

commerce: innovations for conquering current barriers, obstacles and limitations to conducting successful business on the internet (ICEC '06). ACM, New York, NY, USA, pp. 259-270 (2006)
10. Carbo, J., Molina, J., and Davila, J.: Comparing predictions of SPORAS vs. a Fuzzy Reputation Agent System. In: 3rd International Conference on Fuzzy Sets and Fuzzy Systems, Interlaken. pp. 147—153 (2002)
11. Common Criteria for Information Technology Security Evaluation, Version 3.01 Revision 3, Final, July 2009. http://www.commoncriteriaportal.org/
12. OASIS: Web Services Transaction (WS-TX) Technical Committee; http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ws-tx
13. OASIS: Web Services Reliable Messaging TC WS-Reliability 1.1, http://docs.oasis-open.org/wsrm/wsreliability/v1.1/wsrm-ws_reliability-1.1-spec-os.pdf (2004)