Measuring Trust of Components in Web Services

Hermina Duratovic-Field, Prof. Qi Shi, Dr Robert Askwith, Prof. Madjid Merabti
Liverpool John Moores University
H.duratovic@2009.ljmu.ac.uk, Q.Shi@ljmu.ac.uk, R.askwith@ljmu.ac.uk, m.merabti@ljmu.ac.uk

Abstract—Trust is a vast and subjective field that has been researched in areas such as spam, peer-to-peer networks and web services. However, very little research has been done into web service collaboration and how trust related metrics are propagated down to the component layers. Trust has the same core requirements; its representation, exchange, establishment and enforcement, as well as a unified way of measuring it through reliability, decision and reputation based mechanisms. This work will look at describing what efforts have been made in service collaboration, as well as describing the proposed framework to resolve the weaknesses highlighted in the report.

I. INTRODUCTION

Trust is used by society in every decision made. When trusting, we carefully assess other people’s intention, capabilities and motives in context to the situation and task at hand to decide whether a particular person or service is worth trusting [1][2][3]. In computer science and network security, the goals of trust are the same. When network connections are made, one must know that capabilities are in place for necessary actions to be performed, that the parties involved have positive motives and that they have shared intentions.

Web services allow users to execute cross platform applications in the hope that they can share a joint goal or objective. These web services can be integrated to provide a higher level application for the user [4]. When these collaborations occur, the trust management system needs to be able to assess each of the collaborating services in terms of benefits, risks and costs to the overall performance [5]. After the collaboration has ended the trust management system also needs to assign trust related performance scores to each collaborating service.

The existing solutions of trust management can be categorised as direct experience, trusted third parties, hybrid and negotiation [6]. Direct experience is the simplest of trust notions. It only deals with trusting people that we have directly collaborated with. This is because, once collaboration begins, the parties involved reveal what is requested of them in small iterative revelations until trust is established.

This paper will look at trusted third parties and experience based trust. It has the following format. The background chapter will look at the state of the art and examine some of the shortcomings of current work and trends. Following the background we will introduce our proposed solution with some preliminary results. Finally, the document will be concluded and summarised. A larger format of the figures in this document can be found at www.hermi.co.uk

II. BACKGROUND

A. Definition of Trust

Trust is a difficult notion to define. There are many definitions of trust, each tied into a certain concept or a certain subject. Trust has been researched in subject areas such as spam [8], peer to peer networks [9], and distributed systems [10]. For us to be able to measure trust we need to be able to define it in a context that is both relevant and explanatory of the topic at hand. Gambetta’s definition of trust states that trust is “A particular level of subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such an action and in a context in which it affects his own action” [11].

This generic definition of trust can then be evolved further into a specific definition based on the area we are working on. From Gambetta’s definition we can see that trust is a “subjective probability” meaning that trust is based on the security requirements of the user. Furthermore trust needs to be judged before we interact with a service, so that we know that our interactions are secure, timely and accurate. This means that we need to be able to judge the motives, capabilities and intentions [1][2] of the services that collaborate with in the composition. So from this we can evolve Gambetta’s definition to be:

Trust assesses the motives, capabilities and intentions of a component service based on the trust requirements of the user, so that our interactions with the collaboration are secure, successful and accurate.

B. The Problem

Trust in service composition has many dimensions. Figure 1 shows the dimensions of trust and how each of these layers participates in how trust is managed. The first dimension looks at how services are composed, taking into account all of the functional requirements of the cooperation and the Quality...
of Service. From here a list of service collaborations is created with scores that explain how well they fit into user requirements.

The assessment of trust can be categorised as reliability trust, decision trust, and reputation trust. Reliability trust is concerned with the probability that a “reliable” collaborating service will perform accordingly, given that the success of the collaboration is dependent on its performance. Decision trust extends on reliability trust in that it looks at the confidence of our decision to trust a highly dependent component of the collaboration whilst taking into account the cost of possible failing in the task. Finally, reputation trust looks at how well the collaborating component has performed in the past in other associations [12].

It is important that in every collaboration we can assess each service component in terms of trust. That is, it needs to be able to assess the intentions, motives and capabilities of the component so that we can choose the components of the collaboration that will not lower trustworthiness of the entire collaboration. It also needs to be able to assess the component in terms of the task, the role that the component plays in the task, the importance of the task and the consequences of a failing component. However, it is difficult to enforce any kind of trust notion in service collaboration as users deal with the collaboration as a whole and any failures reflect directly on the entire collaboration. For this reason further investigation is required into how trust information can be propagated down to the component layer so that informed trust decisions can be made.

![Figure 1. Trust Management of Component Services](image)

From Figure 1 we can see that trust management in service collaborations are not as straight forward as many trust managements systems. In service collaboration we cannot just use reputation to judge whether the component is trustworthy. We need to look at all of the processes in service collaboration as well as trust management in order to create a solution that is fit for purpose.

C. Related Work

Service composition mechanisms use QoS (Quality of Service) and functional requirements to compose a list of possible collaborations [13][14]. Whilst these are useful and can give an indicator of trustworthiness, they do not always imply that the web service or component is worth trusting. Trust in web service collaboration has been investigated at the enforcement level in the component services. Each solution relies on some kind of reputation or experience information being stored for the collaborating service. They extend on the mechanism for collaboration of services by adding trust as an extra factor to be used when composing the list of possible collaborations. Yahyaoui [15] looks at using a game theory to compose a service collaboration. The aim of the so-called game is to provide an optimal collaboration based on the functional requirement with minimal trust cost. In this game component, services bid to participate in the collaboration and only services that perform the functionalities required with the lowest trust cost win their bid.

Further work looks at introducing a trust vector to each of the possible service collaborations [16] based on Yosang et al’s definitions [12]. Other solutions look at introducing a “Trust Database” to store reputation and experience data and then implement subjective logic [17] or the use of “trust aware” communities to help in the formation of trustworthy collaborations [18].

An interesting approach to service composition is that of X. Xie et al [19] who use the notion of social networks. The social networks include a user social network, a provider social network and a service social network. Trust for the collaborating components is then derived as indirect trust from the social networks or direct trust from direct experience with the components. Other social network based work is that of J. Bentahar and B. Khosravifar [20].

Another interesting approach is that of Y. Dai and G. Wang [21]. This work looks at using a collaboration ontology and finding optimal collaboration paths using trust requirements.

M. Sensoy and P. Yolum look at using ontologies to store ratings given of services in an open environment. The ontology captures the service’s description and the service in detail. It then allows the consumer to give detailed information about the experiences they received. When another consumer queries the ontology they can then evaluate the service in terms of their own requirements and the details that are given on experiences with the service [22].

Furthermore, negotiation has been used to establish trust amongst users and services. A.J Lee and M. Winslett look at encoding trust related issues in the pre-existing web service standards available. It claims that current trust negotiation involves the use of third party certificates and servers in understanding the trust worthiness of a service, and claims that being part of such services and obtaining certificates can diminish the openness of services and not release their full potential [23].

E.M. Maximilien and M.P. Singh propose a framework that focuses on trust using QoS ontologies in an open environment. The framework first captures (using a policy based language) the consumer’s requirements and provides profiles of them. An algorithm then selects services based on the policies and “representations to dynamically capture data about service performance in respect to customisable Quality of Web
service (QWS) dimensions.” This theoretically means that the services are configured at runtime to choose the most suited service for each client’s needs [24].

Finally, S. Cai et al propose that selection of web services can be achieved by creating a web of trust and then mining the data using the Kalman Filter algorithm to analyse the global trust value of each provider and user. Further work of trust in web services involves models based on the collection and derivation of evidence for Service Oriented Architectures (SOA) trust [25].

The main requirements of trust management systems have been defined as representation, exchange, establishment and enforcement. Whilst both establishment and exchange are important research topics, we have found that in service collaboration enforcement has had little work attached to it as most collaboration frameworks assume the existence of some kind of reputation score. When services collaborate the end user does not know this collaboration and any kind of experience related rating can affect every service in a positive or negative way regardless of their performance. In order for any of these techniques to be directly implementable one must have a mechanism in place to distribute ratings to each component based on their performance alone.

III. CURRENT WORK

There is insufficient understanding of trust related issues in component services. Services are composed using functional requirements and Quality of Service (QoS) metrics [21][22]. However, very little regard is given to how trustworthy a certain component is in terms of risks, costs and benefits. Whilst the QoS metrics provide information of how available, documented, and usable service components are [26][27][28][29], they do not provide enough information to decide whether the service will act accordingly.

When evaluating whether a service will act accordingly we need to look at reliability, decision trust and reputation trust [10]. Whilst there has been research into each of these sections of trust in a service, we have not found adequate research into specific trust scoring of component services. A significant amount of effort has been put into looking at reputation scoring of a whole collaborating service [23][24][25][30]. Very little work looks at using these reputation scores at the component layer to rate the collaborating services so that better informed decisions can be made, before collaborations are composed. Each work that looks at trust assessment in the component layer assumes the existence of some kind of reputation score of that component [16][17][18][19][20]. However, in reality, this is not the case. If any feasible solution is to be discovered, this assumption needs to be solved.

In order to solve trust related issues, there are requirements that need to be adhered to. Firstly we need to have a way of representing trust, that is characterising the notion of trust so that it may be understood by all parties involved [31]. We need to be able to describe the credentials (capabilities), behaviours (motives) and intentions of each of the parties [32].

Next we need to look at the exchange of trust, that is how securely we can interchange experiences of services based on trust related issues. Furthermore we need to look at how to establish the trust, and how long this trust can be established for. Finally we need to look at enforcing the trust management system, so that we can continue to establish and utilise the trust management system [31].

From the research we have conducted, we have been able to develop a high level design of a trust management framework, which looks at using knowledge discovery techniques to judge reliability, decision and reputation trust of component services. We have also looked at defining the different stages of collaboration that are included in trust management.

Figure two shows the high level design of the proposed solution. Trust data of collaborations will be put through reputation analysis; from this a reputation score for an individual service will be calculated. This reputation score will then be used with the QWS data to calculate a reliability score for each of the components. Once both the reputation score and the reliability score is calculated, they will be used with the user requirements to calculate a decision score, based on which a collaboration will be calculated.

A. Reputation Analysis

Almost all trust management systems rely on rating a collaboration as a whole. When the user evaluates its interaction, it does not know of the component services that were used to perform the collaboration, nor does it know to what capacity they contributed to the collaboration. However, it is very important that collaborations are formed using trustworthy components. This will allow for better rating for the collaboration, and better rating for the services, also this would allow a new composed service (not used previously) to have a trust rating derived from its components.

This item in the design deals with deriving a trust score for an individual service that has taken part in service collaboration. A repository with all service collaboration data, and trust scores is required. From this data, data mining algorithms will be run in order to provide details of trends within the data; these trends will then be interpreted into trust scores for component services.

This is a novel aspect to trust management of collaborating services. As mentioned previously, most trust management techniques assume the existence of an individual trust score for the component service. However this is not the case. Whilst a lot of the work based on forming services with high
trust metrics exists, they cannot realistically be implemented without the real presence of the trust score.

**B. Reliability Analysis**

Through the research we have conducted we have found that some collaborations are formed using only QoS metrics and functional requirements. These collaborations do not look at the reputation of the component services. Therefore they cannot be measured in terms of trust. If trust reputation was added, it would allow for the existing mechanisms to be extended to incorporating trust, making it a simple and feasible solution.

Reliability trust is concerned with collecting QoS data such as availability, documentation, etc. This data is then combined with the reputation score that is derived using methods in the previous section. It is then mined for trends within the data and its association to the reputation score. These trends are then analysed to provide a reliability score for the component service.

As mentioned previously, the use of "reputable and reliable" mechanisms is not a difficult concept, and it does not form too many integration issues with existing solutions. This mechanism can provide many on the fly results that can make service composition quicker without lowering the overall accuracy of the trust mechanisms. It however does require the existence of individual trust score for the component services.

**C. Decision Analysis**

This work allows the user to set their trust security requirements, and for these to be included in the evaluation of a component’s individual trust score. There has been a lot of cross domain work, which allows the user to specify these requirements using their ontologies security requirements.

However, there is no work that incorporates component reputation scores and reliability scores. The work that frequents the domain looks at specifying the overall reputation scores, and if it is used in component collaboration techniques, again, the assumption of a component reputation score exists.

This mechanism looks at comparing both the reputation and reliability scores on the data, to provide a final score, which will be given a percentage as a confidence score for trusting a particular service within the collaboration. This will be analysed using both scores, functional requirements and trust requirements of the user. This score will be the final tool to judge whether a certain component should be used in a service collaboration for that particular user.

This work will look at allowing the user to specify the trust and functional requirements to provide a detailed requirement specific trust score for a component participating in the collaboration. Only after that will a list of all the suitable services for the collaboration be given. The algorithm will allow for the selection of services without too much user interference. This work is reliant on both reliability and a reputation score being present.

**IV. INITIAL RESULTS**

The initial work implements the QoS metrics to establish if there are any relevant or interesting patterns in the data. This implementation looks at the second layer of collaborative trust management, and in particular reliability trust. We have assumed that the functional requirements have been given, and the data that is in the dataset is a list of possible services to be used in the collaboration.

The initial work looks at establishing reliability trust, the first of the three trust metrics. When deciding whether a component is trustworthy we first need to establish how reliable a service is that is the probability that the chosen component service will perform accordingly. This section evaluates the Quality of Service metrics using knowledge discovery techniques to give certain probabilities of how the component is expected to perform based on other components and their quality of service results. This information provides us with a trend of Quality of Service (QoS) metrics that are important when it comes to assessing reliability trust. This reliability trust score can further be used to help with the assessment of the decision that is made and the decision trust metric.

The data that we will use is part of a QWS data set [26][27][29]. For the purpose of the experiment we will be using attribute 1-10 including the WsRF. As stated previously reliability trust is further analysed to decision trust. We will be using WsRF as the predictable column for the data mining algorithm. WsRF is the reliability metric, because it holds a percentage for the Quality of the service. That means that this metric gives an indication that the service will behave accordingly.

From the data collected we have found that the WsRF has many dependencies but the key dependencies are highlighted in Figure 3. This means that for a composing service to be reliable, it must be available, have good description tags, throw very few errors and produce the highest percentage of responses.

These dependencies show that quality of service data can be assessed to show which of the attributes of the data provide the most information about how reliable a service component is in terms of trust. This allows us to focus on rules that can be developed on the data to find which service component has the highest trust reliability scores. Additionally these can also
be used to help with the assessment of our decision to trust a said component service.

Furthermore the data can be clustered into the categories described in Figure 4. This shows us how the dependencies fit into the difference among WsRF scores, and why the dependencies have been derived. The columns represent the WsRF range of values on the attributes. For example, for the overall population of the dataset, only 137 of the services have availability greater than or equal to 91. However, 93% of those 137 have a high WsRF score.

From the table we can see that if availability is high, documentation is high, reliability is high and successability is high, so our decision to trust a service based on the WsRF is going to be high. This chart also highlights a weak dependency on throughput. From figure four we can also see that even though these dependant attributes of the data are high, we have a low WsRF. That is why the decision metrics will need to be introduced. In the example we can see that WsRF was high even though documentation was less than 11%. In fact 13% of the high WsRF had poor documentation.

These ranges show the distribution of data on the WsRF. From these distributions it is possible to deduce rules on the data so that we can predict future reliability trust scores of component services. Furthermore these distributions of data on the WsRF can be used as an investigative tool of reliability trust on the system. We can see from the distributions how well the component services are performing, and to find weaknesses in the trust management system. In addition this distribution can be further used to help judge decision trust of a component service.

![Figure 4. Clustering of data around WsRF](image)

From this we need to analyse some of the rules that produce high accuracy when it comes to predicting WsRF. These rules will help us establish a high reliability trust in the system. The high accuracy rules are shown in Figure 5. The diagram also shows the importance of these rules in terms of how often they appear in the data. So for example we have the rule:

**Successability < 28, Reliability < 29.9 (1dp) -> WsRF < 52**

(Importance 1.12)

We know from the importance this rule is the most occurring in the data. Because its probability is high we know that any data, which we get in the future that fits in this category, has a very high chance of not producing results for the collaboration.

Another example is that of the following rule:

**Best Practice >= 86, Documentation >= 84 -> WsRF = 69-79**

(Importance 0.28)

This rule occurs least in the dataset. Also because of its mediocre WsRF prediction, we will need other WsRF rules on other data in the set and to add reliability trust in order to make an informed trust decision.

These rules show how we judge the reliability trust of a component service. They can be used to predict reliability scores of future services, where WsRF scores do not exist. This allows for more accurate trust scores of new component services, based solely on quality of service data, when no user ratings have been produced. Furthermore, this can also help
when judging the decision trust of the collaboration, by providing further attributes to evaluate.

From the initial results we have found that rules within the QoS data exist for us to be able to decide reliability and decision trust. When a list of service collaborations is created we can decide which services are trustable and which are not based on the reliability. However it is not sufficient enough as some rules may show earlier given mediocre results in terms of reliability. For this reason reputation and decision metrics will need to be introduced. This reputation score will be taken from the users of the service, and delegated to the various components based on their performance. The decision metric will be a derivation from the two.

V. CONCLUSIONS AND FUTURE WORK

This paper has discussed the background to trust related issues in service composition. We have highlighted the problem and the weaknesses that exist in current solutions. We have also introduced our proposed solution with some preliminary results. As further work we will look at working on analyzing reputation trust of a collaboration service and bringing the relevant trust scores to the component layer. Furthermore we will reassemble these trusts scores as a means of testing that the correct trust scores are given to the component services.

VI. REFERENCES


