TruSSCom – Proposal for Trustworthy Service Representation, Selection and Negotiation for Integrating Software Systems

Lahiru S. Gallege
Department of Computer and Information Science,
Indiana University-Purdue University Indianapolis (IUPUI), Indianapolis, IN, USA.
Email: lspileth@cs.iupui.edu

Abstract

Integrating a distributed software system using publically available software services saves effort, time, and cost. One key step in this process is the service selection which identifies a relevant set of services for this integration. In an open service marketplace, it is hard to judge the trust of software services using a static view (consisting of service specifications) published by their developers. Instead, the concepts of trust in the context of services needs to be properly quantified, specified, negotiated, and then used in the selection process. Prevalent service selection and negotiation approaches do not consider the trust aspect of services. Trustworthy service representation, selection, and negotiation are challenging tasks due to the subjective and temporal nature of trust, lack of standards, and associated uncertainty. This proposal defines the trust of a service, quantifies the trust by monitoring and aggregating various evidences, represents the trust in the service specification and improves the service selection and negotiation process using this representation. Publically available software objects (from Java collection framework library) and mobile app services (from the Android marketplace) are used as datasets to empirically evaluate this proposal.

Categories and Subject Descriptors K.6.3 [**Management of Computing and Information Systems**]: Software Management - software selection.

Keywords : Trust; Service Representation; Service Selection; Service Negotiation; Software Services.

1. Motivation

Proper reuse of software services helps complex software projects to meet budgets and deadlines. The Service Oriented Architecture (SOA) provides concepts of reusing software services to integrate software systems. For example, consider a distributed tracking system that tracks the location of an object (of interest) using publically available sensor services such as Camera, Wi-Fi, and GPS. Such reuse of software services demands an establishment of trust between independent venders and consumers at the time of service selection [1]. Our previous survey [2] confirmed that the trust aspects of software entities (e.g., software objects, libraries, modules, services, etc.) in the context of publically hosted software services are yet to be standardized and that increases the complexity of software service reuse.

SPLASH '13, October 26–31, 2013, Indianapolis, Indiana, USA.
ACM 978-1-4503-1995-9/13/10.
http://dx.doi.org/10.1145/2508075.2508077

Publically available software services, along with associated information, are typically hosted in a cloud – Android market place is one such example. These marketplaces still follow the traditional way of representing the trust of a service, which is to indicate it as a quality of service (QoS) attribute (e.g., numerical star ratings). This naive quantification (or representation) has limited usage at the time of service selection, as it does not reflect important information such as, how the trust value was aggregated and its fluctuations over time. During the lifetime of a service, it is expected that the trust of a service deviates due to various factors such as, changes in operating conditions, new revisions, and feature updates. Hence, a fairly rudimentary and static way of providing a star rating for a publically available service is not a comprehensive solution. Instead there is a need for new approaches to define, represent trust aspects of software services and also to select and negotiate attributes of them based on their trustworthiness.

2. Problem Statement

The focus of this proposal is to "*provide a software framework and associated algorithmic modifications to perform trustworthy service representation, selection and negotiation in the context of service marketplaces*. The goal of this effort is to facilitate effective service reuse while integrating distributed software systems. To achieve this goal, the framework should identify the complete spectrum of the trust of a service at the time of service selection. It should also track and record fluctuations associated with both the service and its environment. This additional information provided by the framework will enhance the service selection process by basing its decision on the trustworthiness of the service. Proposing a software framework for service selection based on the trust of a service is a challenging task due to the disagreements about the definition of trust of a service, the subjective and temporal nature of trust, the heterogeneity associated with available evidences, and the complexity associated with aggregation and evolution of the trust-related aspects of services.

3. Our Approach

This proposal, therefore, focuses on tackling the above mentioned problems of the quantification and representation of trust in the context of software services and its usage in the service selection and negotiation processes. The proposed work is based on our preliminary efforts reported in [1, 2, 3, 4]. This paper initially presents our definition and quantification of the trust associated with a software service. Then it describes the preliminary trust model which helps to monitor and aggregate relevant evidences using suitable operators. Finally, the algorithmic modifications necessary to perform trustworthy service selection and negotiation using the model defined are briefly discussed along with proposed empirical validation scenarios.

3.1 Trustworthy Service Representation

Our formal definition of the trust of a service is based on the definition proposed by the Trusted Computing Group [5] (i.e., "An entity can be trusted if it always behaves in the expected manner for the intended purpose"). Hence, we define the trust of a software service as the "degree of conformance of its behaviour to its published specification."

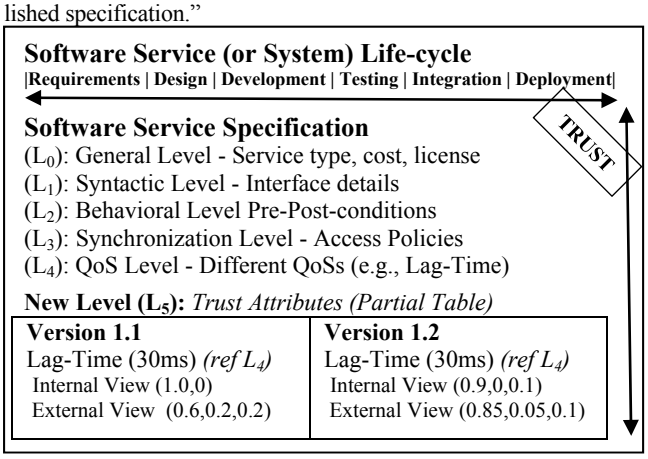

Figure 1. Trust aspects of the multi-level service specification [3]

The trust aspect, due to its definition, crosscuts the entire software lifecycle (Figure 1). At the same time, the software specification is structured as a collection of contracts at each level. Therefore, our trust model [1] focuses on different artefacts (i.e., conceptual or physical outcomes of a particular phase of the lifecycle) and their associated evidences. These evidences are also produced at each phase of software lifecycle of a service (e.g., test cases during the testing phase). For a particular artefact, the evaluations done by the developers indicate its internal view, while the evaluations done by the users/reviewers indicate its external view. These two views provide the necessary mechanisms for quantifying and representing trust of a service. The formal quantification of trust that we used is based on the theory of evidence [6] and theory of subjective logic [7]. Using these theories, our model associates a numerical tuple of Belief, Disbelief, and Uncertainty (B, D, U) to indicate the trust about each evidence. The numerical values of the B, D, and U are between 0 and 1 and their sum equals to 1. This BDU-based quantification is selected because of its inherent ability to capture the subjectiveness of the associated evidences and a lack of complete information (as represented by the "U" part) about evidences.

In general, the service selection process evaluates the service meta-data which are publically available via the service specifications. For example, a multi-level structure is proposed by Beugnard et al. [8] has four levels in the service specification (syntax, semantics, synchronization, and QoS (Figure 1 - $L_{1\text{-}4}$)). However, most prevalent approaches do not include trust aspects as a part of the specification. We have modified our previous version [9], an enhancement of the Beugard et al.'s specification (with L_0 indicating the inherent attributes), also to include trust aspects of a service (Figure 1 - L_5). The trust crosscuts all levels hence, the BDU tuples are computed for all the levels for each attribute. These values are computed as a reference to another level attribute (e.g., BDU of Lag-Time is referring to QoS attribute) and placed as a part of trust contract according to an agreed temporal aspect (e.g., versions). The calculation of the BDU tuples are done in both internal (i.e., developers) and external (i.e., users) perspectives with respective to that particular attribute of the specification [1, 3, 4]. The aggregation of these individual BDU tuples (represent each evidences with respect to a certain attribute in the specification) is done by subjective logic operators such as the conjunction and consensus [4, 7]. For example, when multiple opinions are present, such as (e.g., consistent or conflicting user reviews about Lag-Time of a data service) then the consensus operator is used to aggregate these opinions.

3.2 Trustworthy Service Selection

Prevalent service specifications do not include a separate trust-related section. The existing selection schemes also do not use operators which are needed for matching trust-attributes. According to the above scheme (Section 3.1), when the trust contracts are created for services, they are also used to perform the selection process. We have abstracted the generic structure (i.e., Table 1 without the terms with *) behind the prevalent service selection algorithms (such as [9]). Additionally, we propose the following modifications to this abstracted structure (shown in Table 1 with *) to perform trustworthy service selection [4].

The off-line phase of the service selection algorithm periodically collects the evidences about trust attributes published by a service contract. The time frame of the aggregation can be hourly, daily, or based on versions. Figure 1 shows the trust aggregation of one attribute (i.e., Lag-Time of a data service) based on versioning. A significant divergence between the B, D, U tuples for the attribute in Version 1.1 is noted with then a convergence between the views in Version 1.2. Possible reason for such a change of opinion between versions is that, in the former version the developers have not considered the external opinions and later updated the service in the next version by considering them. These progressions of trust views provide additional information to the on-line service selection phase. In this phase, the service selection process requires additional matching operators such as conjunction and ordering [4, 7] to match new information available through the proposed trust contract. Having these trust related aggregations collected over a period of time provides additional knowledge to the service selection process.

Table 1. Service selection algorithm modifications

Service Query	List of service (*trust**) requirements
Off-Line Phase	Periodically aggregate (*trust**) attributes of services
Online Phase	Perform (*trustworthy**) service matching
Output	Ordered list of Services (*based on trustworthiness**)

3.3 Trustworthy Service Negotiation

As indicated earlier, most prevalent service selection schemes do not include a trust establishment based on mutual evaluations between the service vendors and consumers. Hence, these service negotiation schemes operate only based on quality of service agreements such as, service level agreements (SLAs). Online services are affected by the change of the environment conditions, which result in the fluctuations of the quality of service offered. Therefore, we propose a negotiation establishment based on the trust between vendors and consumers at the time of service selection. These establishments are based on the aggregated views of a trust of the services (Figure 1) and their future rules of engagements to address the fluctuations. For example, trust aspects of the negotiation contact should include information about which evidences to consider in quantifying the effects of a service outage. Considering the prevalent negotiation strategies, we are currently investigating on how the algorithms can be modified based on two phases (as described in Table 2). An active phase of negotiation is initiated at the end of the service selection. This is based on the current views of the trust of that service which establishes a mutual agreement of negotiation for current and future variations. After selection process is completed, as new evidences are available (such as new feature updates and releases), the passive phase periodically monitors the

agreement. If any violations are noted then the negotiation switches back to the active phase.

Table 2. Service negotiation algorithms modifications	
Active Phase	Initiates the trust establishments during service selection. Agreements on future fluctuations in trust establishments.
Passive Phase	Periodically monitors the trust establishments after selection. Updates trust establishments (both vendors and consumers).

4. Evaluation Methodology

To evaluate the above approach, we propose to use two datasets namely, the Java Collection Framework and Android Marketplace. Selected objects from the collection framework and selected apps from the marketplace are assumed to behave as services within our prototypical (TruSSCom) environment.

4.1 Case Study using the Android Marketplace

We evaluated, as a preliminary study, a subset of Android apps in relation to a trip planning case scenario [4]. External reviews available about these apps are mainly considered for aggregating evidences over time. The selected service dataset includes 36204 apps and 1108343 reviews. We used a default value for the internal view of a service because the evidences are not available. When a service query is submitted, the prototype displays a list of possible instances (for each type of service needed, e.g., travel related services). This process includes the prevalent selection process plus the above described trustworthy service selection.

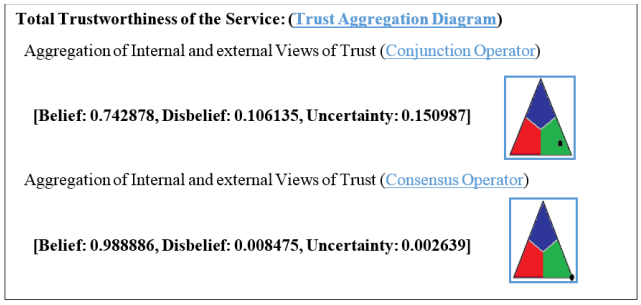

Figure 2. Total trustworthyness of a service is presented usig two operators (conjunction and consensus).

The TruSSCom prototype calculates the trustworthiness of each service based on associated evidences. It also tracks the progression of trust for that service based on temporal intervals (i.e., versions, weeks). It can switch the algorithms which calculates the trust of services and compare different selection results. For example, the current algorithm used to compute the trust tuple is based on the external reviews about a service. It is calculated using the services': 1) average user star ratings, 2) each review, using a suitable techniques (such as keyword counting or term-frequency-inverse document frequency), 3) the reputation of the reviewer by using the recommendation operator from [7], and 4) the expiration of the reivew using the distence to the review date from today. The consensus operator is used to aggregate these different views of the service to obtain a single quantification of the trust of the service. Figure 2 presents a screencapture of the prototype indicating the computed trust value of a selected service. The prototype follows two methods for calculating the total trustworthiness of a service (Figure 2). The consensus operator is preferred when the evaluated views fall within a range. Otherwise the conjunction operator is preferred. Final aggregation values are displayed using both numerical and graphical representations. We also use a BDU triangle which is simmlar to the visual representation used by Josang et al. in [7].

4.2 Case Study using the Java Collection Framework

Currently we are evaluating a subset of classes from the Java Collection Framework. We run them in different virtual machines to simulate as services hosted in marketplaces. In our experiment, the serializable interfaces of the objects together with their JavaDoc are treated as their service specifications. The scenario is assumed to be the selection of data services (e.g., data structure operations related to List, Queue, Set, Vector and Map). We base our study from the start of the collection framework (i.e., from pre-JDK 1.2 libraries) to the new Java-7 libraries. Our first goal is to identify the limitations of the initial versions of the operations by investigating bugs and features reported by users. Naturally, they are fixed and updated in later releases. Our trust evaluations are based on those evidences and we experiment the service selection negotiation algorithms against them. Since the collection framework is defined as a hierarchy of interfaces, we can study the effects of service substitution (e.g., super class operation – Sub class operation) based negotiation and their effect in the trust of the integrated system.

5. Conclusion and Future Work

Based on the concepts of theory of evidence and multi-level specifications, the trust aggregation of services using a contract improves the service selection process. Future directions include: further experimentation of the TruSSCom prototype and find ways to improve the certainty in predicting trust values of services.

Acknowledgments

This work is supported in part by the US Air Force Research Laboratory (AFRL) through the Security and Software Engineering Research Center (S^2ERC). Our research group members are Dimuthu U. Gamage, Dr. James H. Hill, and Dr. Rajeev R. Raje.

References

[1] L. S. Gallege, D. U. Gamage, J. H. Hill, and R. R. Raje, "Towards a comprehensive method for integrating trust into enterprise DRE systems," in *Proceedings of the Embedded and Real-Time Computing Systems and Applications (RTCSA)*, IEEE Computer Society, 2011.

[2] L. S. Gallege, D. U. Gamage, J. H. Hill, and R. R. Raje, "A Study of Trust in Distributed Systems," Department of Computer and Information Science Technical Report (TR-CIS-0420-03), 2012.

[3] L. S. Gallege, D. U. Gamage, J. H. Hill, and R. R. Raje, "Trust contract of a service and its role in service selection for distributed software systems," in *Proceedings of 8th Cyber Security Information Intelligence Research Workshop (CSIIRW '13)*, ACM, 2013.

[4] L. S. Gallege, D. U. Gamage, J. H. Hill, and R. R. Raje, "Trustworthy service selection using long-term monitoring of trust contract," to appear in *Proceedings of 17th EDOC Conference*, IEEE, 2013.

[5] S. Pearson, "Trusted Computing Platforms: TCPA Technology in Context," Prentice Hall PTR, Upper Saddle River, NJ, USA. 2002.

[6] G. Shafer, A Mathematical Theory of Evidence. Princeton University Press, Princeton, 1976.

[7] A. Jøsang, "Artificial Reasoning with Subjective Logic," in *Proceedings of the Second Australian Workshop on Common-sense Reasoning*, Perth, Australia, 1997.

[8] A. Beugnard, J. M. Jezequel, N. Plouzeau, and D. Watkins, "Making components contract aware," *IEEE Computer*, pp (32), July 1999.

[9] L. S. Gallege, K. Pradhan, and R. R. Raje, "Experiments with a multi-level discovery system," in *Proceedings of the International Conference in Computing (ICC)*, New Delhi, India, 2010.