AAAI Technical Report SS-12-04
Intelligent Web Services Meet Social Computing

Component Trust for Web Service Compositions

Mohammad Reza Motallebi †‡ and **Fuyuki Ishikawa** † and **Shinichi Honiden** †‡
† National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
‡ Department of Computer Science, Graduate School of IST, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
{mohammad, f-ishikawa, honiden}@nii.ac.jp

Abstract

The concept of trust in web services describes the degree of belief that a client or a group of clients have over services functioning satisfactorily and providing the expected results. As services are usually invoked in composition with other services, judging on their trustworthiness gets more complicated, yet computing their trustworthy becomes a desired goal. Existing work only take the trust of each individual service into account, regardless of the context of the composition. They also do not use the data gained from other clients for selecting the most trustful composition and preparing for possible service failures.

In our work we first introduce the concept of Combination Reputation, which reflects the commonness and popularity of invoaction of a pair or group of services among other clients. By interpreting the trust and reputation values as subjective probability, we define the Component Trust of the services in the composition, which reflects the degree of belief the client has over components of services performing satisfactorily. We model the web service composition as a Bayesian network and integrate the above trust values into the network and show how to compute the global trust of the composition.

Introduction

Web services are defined as "a software system designed to support interoperable machine-to-machine interaction over a network [which] has an interface described in a machine-processable format ...". They are seen as a workaround for resolving the limitations of conventional middleware and being the entry point to the local information systems of enterprises by providing a service-oriented architecture, set of protocols and standards (Alonso et al. 2004).

There are numerous work in the web services community trying to address the functional and non-functional requirements for web service selection. Although the semantic description of web services is a required step towards building functionally valid compositions, but it is not sufficient. Various problems and failures could rise within services and their orchestration that may lead to the breakdown of the composition. Here we focus on the class of problems that are caused by incorrect or incoherent results. Causes such as misbehaving execution flow and misunderstood behavior could lead to faults resulting the incompatibility between service invocations (Chan et al. 2009). More specifically, we consider the case where the response of one service would not produce expected results when provided as an input to another service in the composition, even though the input and output parameter types defined in their interfaces match in type and in semantics. We will use the concept of trust to address this problem.

Trust has been extensively studied in various fields of computer science and has been given different definitions. In the field of web services, Li et al. have defined trust as the extent which the service client believes that the service provider can satisfy the client's requirement with desirable performance and quality (Li, Wang, and Lim 2009). They use Bayesian inference to estimate the trust value of each service using ratings from different clients assuming that it follows a normal density. Then they provide a Monte-Carlo based algorithm for finding the services resulting the highest global trust. In (Li and Wang 2009), Li and Wang provide a method for mapping user ratings to subjective trust values and compute the global trust using a breadth-first search on the execution graph.

We use trust for web services to represent the individual opinion of service clients on how well the service functionally performs and provides expected results according to user ratings (Wang and Vassileva 2007). However, with respect to the existing work, we believe that using the trust value of individual services is not sufficient for computing the global trust of web service compositions. Furthermore, considering faults and failures that could rise in service compositions, obtaining the set of services that result highest global trust and also preparing for possible failures is a crucial matter.

In the next sections we will explain our proposed solution to these problems where we will first introduce the new concept of Component Trust based on composition reputation and service trust. Then we will provide a procedure for modeling a composition as a Bayesian Network and computing the trust of the composition using network queries. At the end we will provide some the results of our experiments and discuss the advantages and weaknesses of this work.

Background and Motivation

Trust is literally defined as “firm belief in the reliability, truth, or ability of someone or something” or “an assured reliance on the character, ability, or strength of someone or something”. Trust has also been studied extensively in different fields of computer science from the web to multi-agent systems and has been given different definitions.

The concept of trust has been studied in the context of semantic web. Artz and Gil (Artz and Gil 2007) have presented the results of their survey on the application of trust in semantic web and have divided it into four main areas: policy-based, reputation-based, general models, and the ones pertaining to information resources. However, they have mainly considered trust as a measurement of the authenticity of a party. Trust and reputation models are also common themes on the web. Websites such as eBay, Amazon and Google have their own methods of measuring reputation, reliability and trustworthiness of products and web pages. eBay’s reputation mechanism known as Feedback Forum and its rating system is known to be one of the reasons for its success.

Trust has also been considered in the multi-agent systems community. Castelfranchi and Falcone provide a socio-cognitive model of trust by defining three elements of “core trust”, “reliance” and “delegation” that trust is made of, in which an agent must have certain beliefs in, in order to trust another agent (Lim, Keung, and Griffiths 2010). Opposed to the cognitive view of trust in multi-agent systems, there exists a probabilistic (or computational) view, which utilizes an agent’s past behavior for calculating the subjective probability of showing a particular behavior in the future (Artz and Gil 2007). Huynh (Huynh 2006) considers trust for open multi-agent systems as “A measurable level of the subjective probability with which an agent a assesses that another agent b will perform a particular action in a favorable way to $a, \ldots$”. If we consider web services as a technology closely related to multi-agent systems, distributed systems and semantic web, it would not be an overstatement to claim that trust is essential for the web services too. The present work tries to address trust from a perspective similar to the ones given above.

The concept of trust has been studied to some extent in the services community and has been considered from different aspects. Some look at trust from a security-based viewpoint such as verifying a service’s authenticity and authority, and others see it from a behavioral aspect (Paradesi, Doshi, and Swaika 2009). One of the applications of trust is to ensure consistency between services in a composition and prevent failures (Nepal, Malik, and Bouguettaya 2009) (He et al. 2009) (Paradesi, Doshi, and Swaika 2009), since it can be used to represent the individual opinion of service consumers on how well the service functionally performs and provides expected results according to user ratings (Wang and Vassileva 2007). Using trust semantics, the service consumer would assign a trust value to each of the services it interacts with. This value can be measured as (or converted to) a value in the range of $[0, 1]$. Thus, a good approach, similar to Li and Wang (Li and Wang 2009) and others, would be to regard the trust value as a subjective probability representing the degree of belief the consumer has on a service performing in an expected manner and providing valid results.

With services being invoked in a composition, the problem would be to compute the trust value of the whole composition, i.e. the global trust value. This way, the concept of trust we used for each of the services would be extended to the whole composition, expressing the client’s degree of belief on whether the composition would perform correctly and provide expected results. Li et al. (Li, Wang, and Lim 2009) approach this problem by defining the trust value of service compositions based on their invocation types and provide a shortest-path algorithm and a Monte-Carlo based algorithm for computing the global trust value. In another work, Li and Wang (Li and Wang 2009) compute the global trust by multiplying the trust dependencies on the shortest-path of the execution graph.

While utilizing the trust value of each of the services is the key component to measuring the trust of the composition, we believe it is still insufficient and the context of each service in the composition should also be taken into account for computing the global trust value. In this paper, we approach this problem by first introducing the new concept of Component Trust, and utilize it to compute the global trust value using Bayesian Networks.

Trust for Web Service Composition

System Design

We first give an overall view of the design of our proposed system. We propose a tool that performs as a middleware in the service composition system. It would act as a layer on top of other service selection and recommendation middleware. Therefore, we assume that the candidate services provided to our tool already satisfy the non-functional requirements; i.e. they have similar QoS values or there are no hard constraints on some of the QoS metrics of those services. Since there are many existing works proposed for such purposes, we will not go into the details of addressing the non-functional requirements and assume that such a setting is practical.

Our proposed middleware consists of 4 stages. At first, the service composition is passed to the tool, for example in the form of a BPEL file. The system then is provided with a set of concrete services for each of the abstract services in the composition. In the next stage, the trust value of each of those services is assessed. As it will be explained in the next section, this part could be done using various trust measurement tools, as long as the trust value is in the range of [0,1] which could be applied to our method. The trust value obtained in this stage would represent the direct trust of the service composition invokee on each of the individual services. Next, the system would query for the frequency of service invocations among other clients. The usage statistics, similar to web page hits and referrals, could be reported and queried from a central server, or the client could keep a list of other reliable clients to query from. Once these usage statistics are obtained, they are passed to the Bayesian network engine, along with the trust values for each of the

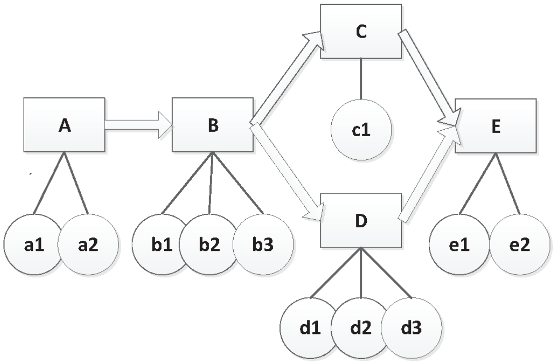

Figure 1: Service Composition

services. The engine then computes the trust of the composition and provides the set of services that result highest global trust. By using this tool and the methods described in this paper, the service consumer can measure the trustworthy of the instantiation of services in the composition and while recording these values for the highest one, be assured that the computed global trust satisfies minimum requirements. At last, the selected set of concrete services is passed to the execution engine.

Web Service Composition

Here we provide a model of web services in order to implement the notion of trust of a web service composition. We model a web service composition W as a set of service classes C_i linked in one of the *sequential*, *parallel* and *activation* invocation types (Li, Wang, and Lim 2009). Each service class C_i has a set of $n(i)$ service candidates S_{ij} ($j \leq n(i)$), where all candidates, given their WSDL interface, are functionally similar and have the same type and number of required request parameter types and given the same preconditions, provide the same response parameter types and effects. We assume that each service class is used only once in the composition (or otherwise with a different name) and that each service candidate belongs to one service class only. As an example, in Fig. 1 the service class B is invoked sequentially and has 3 candidates. Service class E is activated using services of class C and D and has 2 candidates that fulfill the service functionality. Furthermore, services S_{ij} in each class class, are previously selected to satisfy our initial QoS requirements, so we can say they have similar QoS values. We assume that all these constraints are satisfied beforehand by preceding tools and the set of such services are passed onto our middleware to select the most trustful ones in terms of functionality and execution of the services in a composition.

Having a composition work-flow W, an execution E is a set of service candidates S_{ij}, one from each class C_i, which the client uses for the composition at a point in time: $E = \{S_{ij} \in C_i \,|\, j \leq n(i)\,;\, \forall S_{ij} \forall S_{i'j'} : C_i \neq C_{i'}\,;\, \nexists C_i \forall S_{i'j'} \in E : S_{i'j'} \notin C_i\}$. Given this definition, the main goal of this paper is to measure the trust value of the service executions, in order to select the execution having the highest global trust.

Component Trust

Various causes could lead to failures in a web service composition and using trust for services is one of the approaches in addressing this problem. Causes such as misbehaving execution flow and misunderstood behavior could lead to faults resulting the incompatibility between service invocations (Chan et al. 2009). However, the value assigned to each service representing its trust, would be most useful when the client is willing to invoke a single service and wishes to know how correctly the service will perform based on its own previous experience or information gained from other clients. When the service is invoked in a composition, it is receiving input from other services and providing input to others. Thus, the performance of a specific service, measured using the the trust value, would depend on other services preceding and succeeding it in the execution. Thus, we can say that trust in services is context-specific (Wang and Vassileva 2007) and extends beyond the trust value given to each of the services. For web service composition, the context of a service's invocation from the consumer's point of view is the services providing input to it and the services receiving input from this specific service and the performance of the service would depend on them.

A special case of this dependency that would enlighten our claim of trust being context-specific would be when two services having high trust values and reported to function correctly with other services, fail when invoked in combination with each other (for example in a sequential composition). The cause of these failures could be simple inconsistency between the request/response values that cannot be handled by the other party. So we believe that solely relying on the trust values of the services for computing the trust of the composition is insufficient. Therefore we introduce the new concept of *Component Trust (CT)* to address this shortcoming.

The trustworthy of a pair of services a and b invoked in a composition (for example in a sequential manner), can be obtained by combining the trust value of the services a and b and a value representing the consistency of the composition. We represent their consistency with the collective opinion of other clients on the combination of these two services and call it *Combination Reputation (CR)*. Similar to the trust value of the services, various methods can be used to measure *CR* of pairs (or groups) of services. Here we use the invocation frequency of those services among similar users as the measure for representing the invocation's popularity and commonness to explain the reputation of the combination of services a and b in the composition.

For measuring *CR*, we take an approach similar to Rong et al. (Rong, Liu, and Liang 2009), which have proposed a web service ranking framework inspired by collaborative filtering. However, we consider the frequency of an invocation as

$$freq(a,b,T) = \sum_{u_k} n(a,b,T,u_k) \,/\, \sum_{u_k} \sum_{a,b} n(a,b,T,u_k) \tag{1}$$

which defines the frequency that consumers use service a of service class A and service b of service class B in an invocation of type T (e.g. *sequential*) among all other service within the same classes A and B and in the same invocation type. We should mention that in *sequential* and *parallel* invocation types, we will measure the reputation of the combination for groups of two succeeding services, since each service receives parameters from its preceding service in the composition. But for *activation* invocations, because one service relies on the response of a few services and its performance would depend on all its preceding services providing input to it, the context of trust for that case is defined for a group of services. Therefore, we will be measuring the invocation rate of a group of services in invocations of type *activation*, i.e. $freq(\{m, n, \ldots\}, z, Act, u_k)$. Having obtained the frequency of service invocations from different sources, the client could build the *Combination Reputation (CR)* of a pair (or group) of service candidates by taking the average of the frequency of invocations.

It should be noted that here we do not talk about the architecture of the system. The numbers for frequency of invocation could be obtained either from a central server, or from other clients themselves. However, when receiving the values from multiple sources, the client would receive the values of $\sum_{u_k} n(a, b, T, u_k)$ and $\sum_{u_k} \sum_{a,b} n(a, b, T, u_k)$ separately so it can compute the average across queries from different sources.

Next, a service consumer should integrate the *CR* values with its own trust values given to each web service. Various trust computation methods, such as a mapping from rating space to trust space as provided in (Li and Wang 2009), could be used as long as they provide a continuous (possibly scaled) value in $[0, 1]$, that reflects the user's subjective opinion on the trustworthy of the service. For integrating the *CR* and the trust value of services, a client should use a *Trust-Combination-Operator* (Ω) (Yang et al. 2006). Using it we will have

$$CT(a, b, T) = \Omega(Trust(a), Trust(b), CR(a, b, T)) \quad (2)$$

which gives us the *Component Trust (CT)* of service a of class A and service b of class B in an invocation of type T. A common choice for the Trust-Combination-Operator is multiplication. Given that each service would be part of two components (one preceding and one succeeding), we choose the product of the square root of the trust of services in the component and their CR as the *Component Trust* value:

$$CT(a, b, T) = \sqrt{Trust(a)} \cdot \sqrt{Trust(b)} \cdot CR(a, b, T) \quad (3)$$

Having the trust of services and the reputation of their combination as a value in $[0, 1]$, we reach the CT value which would represent the subjective belief on the trustworthiness of the composition of services a and b in an invocation of type T. This too being a value in $[0, 1]$, can be interpreted as a subjective probability, i.e. the degree of belief of the consumer on the trust of this component. Therefore we can benefit from algorithms for subjective or Bayesian probability.

Bayesian Network

Bayesian network provides the means to compactly represent the joint probability of a set of variables. It provides a systematic and localized method for structuring probabilistic information about a situation into a coherent whole. It also provides a suite of algorithms that allow one to automatically derive many implications of this information, which can form the basis for important conclusions and decisions about the corresponding situation (Darwiche 2010). It is a compact representation of a probability distribution that is usually too large to be handled using traditional specifications from probability and statistics such as tables and equations (Jensen and Nielsen 2007).

A Bayesian Network for a set of variables is formally defined as a pair of (G, Θ), where: G is a directed acyclic graph (DAG) over variables Z, called the network structure, and Θ is a set of Conditional Probability Tables (CPTs), one for each variable in Z, called the network parameterization (Darwiche 2009). The parameter G representing the DAG in the Bayesian network, encodes the qualitative part of the model, showing how variables influence their descendant variables and how each variable is conditionally independent of its non-descendants given the state of its parents. On the other hand, the parameter Θ, represents the quantitative parameters of the network, which are described in a manner which is consistent with the Markovaian property between each variable and its parent (Darwiche 2009) (Jensen and Nielsen 2007).

Using the *Component Trust* value for combination of web services that we previously measured, we can compute the global trust value of the composition. For this, we provide the required procedure for modeling the web service composition as a Bayesian network, by incorporating the previously explained trust values into the conditional probability tables (CPTs) of the nodes in the network. This model, which was inspired by the network for Reliability Block Diagrams, has some advantages: providing a graphical model depicting the dependency between the trust of the services and invocations and benefiting from the set of algorithms and queries available for Bayesian probabilities are among them.

Consider a composition consisting of three service classes $S1$, $S2$, and $S3$, having 2, 2 and 1 candidates each that are invoked sequentially. First, for each of the candidates $s11$ to $s21$, we add a variable representing whether the service candidate is invoked in the composition or not. These nodes, shown as u_s11, etc. in Fig. 2 , act as the root nodes of our network and have two outcomes: `Used` and `NotUsed`. We evenly distribute the probability of candidates of one class being invoked. So, for the two services $s11$ and $s12$ of class $S1$, each have the probability of 0.5 for being invoked. If class S had n service candidates, then $P(u\_Si = Invoked) = 1/n$. Obviously, the probability of the candidates not being invoked would be the complement of them being invoked, $P(u\_Si = NotInvoked) = 1 - P(u\_Si = Invoked)$. Next we add nodes representing the trust value of the service candidates and add an incoming link from the root nodes. These variables, shown as $S11$, etc. in Fig. 2, take two values, `Trusted` and `NotTrusted`. Based

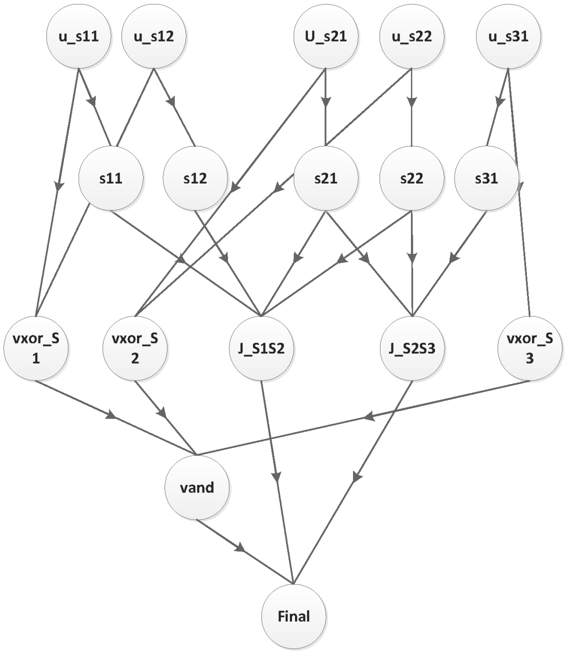

Figure 2: Sample Bayesian network

on the value of the preceding node, it will either report the trust value of that candidate or 0 if that candidate is not invoked; $P(s11 = Trusted \,|\, u\_s11 = Invoked) = Tr(s11)$, $P(s11 = Trusted \,|\, u\_s11 = NotInvoked) = 0$. The probability value for the case that the node in not trusted, i.e. $s11 = NotTrusted$ would be the complement of it being trusted.

Having the variables for all candidates in the composition, we then add a node representing the trust of the joint invocation. This node, shown as J_S1S2 and representing the composition of service classes $S1$ and $S2$, would have links incoming from all service candidates of $S1$ and $S2$. The output would be either `Trusted` or `NotTrusted` and for each pair of service candidates sij and $si'j'$, the value for their *CR* is inserted into the CPT of this node as the trusted value. For candidates from the same service class, the trusted value of the node would be zero.

So far we can compute the trust value of a simple composition, consisting of only two service classes. For cases where the size of the composition is greater, we would have to add nodes to join the joint invocation nodes, J_SiSj. Either more nodes would be added between every two J_i, or one node would be added between all J_is. They would have an AND CPT, giving a `Trusted` value for only the case where all inputs are `Trusted`. Also, we add nodes showing the validity of the selected service candidates. The nodes $vxor\_Si$ shown in Fig. 2, act as an XOR function to ensure that no two services from the same class are selected. Its incoming links are from all nodes u_sij from a class S_i and the CPT has the value 1 for cases where only one of the u_sij has value `Selected`. Finally we link all validity nodes $vxor\_Si$ and the joint node to a new AND node to ensure that services selection in all classes are valid.

The trust value of the composition would then be the joint marginal for the final node, $Pr(Final = Trusted, e)$, where the evidence is the valuation for all u_sij of all service classes s_i. It should be noted that the probabilistic service invocation type, where a few services are invoked alternatively using a certain probability value could also be shown in our model, by inserting their selection probability in the CPT of u_sij nodes, instead of assigning equal selection chances.

Calculating the posterior marginals for variables in a network is one of the basic queries in Bayesian networks. There are a few algorithms and approaches to infer the probability of the whole network given the CPT of each variable. Inference by variable elimination, inference by factor elimination and inference by conditioning are the main approaches for solving the problem. Each of these approaches have their own benefits, yet they differ mainly in their space and time complexity. While the details of these approaches are out of the scope of this paper, we will use them in our experiments and compare their performance.

Evaluation

For evaluating our Bayesian network model, we used the SamIam tool (http://reasoning.cs.ucla.edu/samiam/). SamIam is a comprehensive tool developed in Java for modeling and reasoning on Bayesian networks, which consists of a graphical user interface and a reasoning engine. The composition configuration files, for example for the BPEL engine, can be encoded in the BIF XML file format which can be used with SamIam. We generated sample workflows to simulate the web service compositions. Next we transformed these workflows to their corresponding Bayesian networks and measured the performance of our method using this tool.

Our tests were carried out on an AMD Phenom(tm) II X4 955 machine with 8GBs of memory and running Ubuntu 11.04. It should be noted that the SamIam inference engine (inflib.jar) was observed to perform most of its computation in one thread and did not utilize more than one of the CPU cores. In order to evaluate the service composition network model, compositions of different sizes were generated and different number of service candidates were assigned to each of the service classes in the composition.

For our evaluation, we were able to reduce some of the constraints and simplify the network. By assuming that only a single service candidate from each class will be selected, we can remove the validity nodes V_i, all the edges going out from them and the nodes joining them. Furthermore, given the same assumption, we can have duplicates of the nodes representing the service candidates and their corresponding usage nodes, ai and u_ai. Each one of the ai nodes would then be connected to one end of the joint, and the evidence for the u_ai should be set accordingly, i.e., a service candidate is either used in both joints (invoked by a preceding service and invoking its successor), or not used in both. This way, we would have a tree-shaped network (network with treewidth of 1), and the the posterior marginals can be computed in linear time.

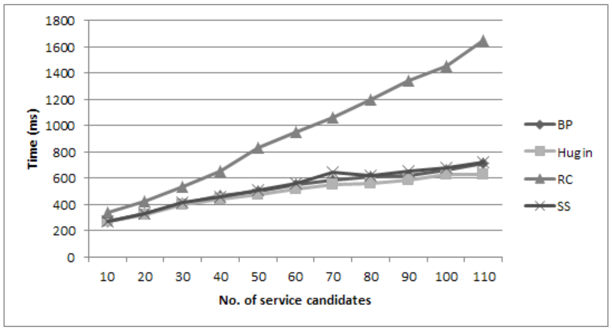

Figure 3: Probability query of the trust value of compositions

Web service compositions that were generated consisted of 2 to 12 service classes, each having up to 10 service candidates. We ran the tests using four algorithms that were provided by SamIam for computing posterior marginals: Recursive Conditioning (RC), Hugin, Shenoy-Shafer (SS) and Belief Propagation (BP). We observed that the the performance of the probability query algorithm mostly depends on the number of service candidates in the Bayesian network, rather than the number of classes, as it would introduce more nodes and states into the network. We queried the trustworthy of the compositions having the number of service candidates shown in Fig. 3 and measured its average processing time. As it can be seen, all algorithms maintained their linear running time, with the Recursive Conditioning method having a higher runtime constant compared to the Hugin, Shenoy-Shafer and Belief Propagation algorithms.

Conclusion

In this paper, we reviewed the concept of trust for web services and identified the context of services' invocation in the composition as one of the points that cannot be addressed using current solutions. We introduced the notion of Component Trust which uses the frequency of invocation of a pair or group of services as the measure representing their reputation. Next we described a procedure for modeling a service composition as a Bayesian network. By integrating the Component Reputation and service trust values into the network, we can then compute the global trust of the composition using the posterior marginal query of the network. We carried out experiments using simulated test cases and showed the results obtained using the SamIam tool.

Although computing the global trust value of the composition was not a heavy task, however choosing the service execution with the highest trust value is a relatively complex process which the client has to record and pre-compute the global trust values of executions in order to come up with the most trustful one. As future work, we wish to overcome this limitation by applying other Bayesian network queries such as Maximum A Posteriori (MAP) to our model. Other matters such as the unwillingness of some clients to provide the actual frequency of invocations, or the disjunction of the service trust computation method and the Component Reputation method could be mentioned as the shortcomings of the current paper which we plan to study in our future work.

References

Alonso, G.; Casati, F.; Kuno, H.; and Machiraju, V. 2004. *Web Services: Concepts, Architectures and Applications*. Berlin: Springer.

Artz, D., and Gil, Y. 2007. A survey of trust in computer science and the Semantic Web. *Web Semantics: Science, Services and Agents on the World Wide Web* 5(2):58–71.

Chan, K.; Bishop, J.; Steyn, J.; Baresi, L.; and Guinea, S. 2009. A fault taxonomy for web service composition. In *Service-Oriented Computing-ICSOC 2007 Workshops*, 363–375. Springer.

Darwiche, P. A. 2009. *Modeling and Reasoning with Bayesian Networks*. New York, NY, USA: Cambridge University Press, 1st edition.

Darwiche, A. 2010. Bayesian networks. *Communications of the ACM* 53(12):80.

He, Q.; Yan, J.; Jin, H.; and Yang, Y. 2009. ServiceTrust: supporting reputation-oriented service selection. *Service-Oriented Computing* 269–284.

Huynh, T. 2006. *Trust and Reputation in Open Multi-Agent Systems*. Ph.D. Dissertation.

Jensen, F., and Nielsen, T. 2007. *Bayesian networks and decision graphs*. Springer Verlag.

Li, L., and Wang, Y. 2009. Subjective Trust Inference in Composite Services. *AAAI 2010* 11–15.

Li, L.; Wang, Y.; and Lim, E. 2009. Trust-Oriented Composite Service Selection and Discovery. *Service-Oriented Computing* 50–67.

Lim, S. N.; Keung, C.; and Griffiths, N. 2010. *Agent-Based Service-Oriented Computing*. London: Springer London.

Nepal, S.; Malik, Z.; and Bouguettaya, A. 2009. Reputation Propagation in Composite Services. In *2009 IEEE International Conference on Web Services*, 295–302. IEEE.

Paradesi, S.; Doshi, P.; and Swaika, S. 2009. Integrating Behavioral Trust in Web Service Compositions. In *2009 IEEE International Conference on Web Services*, 453–460. IEEE.

Rong, W.; Liu, K.; and Liang, L. 2009. Personalized Web Service Ranking via User Group Combining Association Rule. In *2009 IEEE International Conference on Web Services*, 445–452. IEEE.

Wang, Y., and Vassileva, J. 2007. A Review on Trust and Reputation for Web Service Selection. In *27th International Conference on Distributed Computing Systems Workshops (ICDCSW'07)*, 25–25. IEEE.

Yang, S. J.; Hsieh, J. S.; Lan, B. C.; and Chung, J.-Y. 2006. Composition and evaluation of trustworthy web services. *International Journal of Web and Grid Services* 2(1):5.