Trust Evaluation in Composite Services Selection and Discovery

Lei Li
Department of Computing
Macquarie University
Sydney, NSW 2109, Australia
leili@science.mq.edu.au

Yan Wang
Department of Computing
Macquarie University
Sydney, NSW 2109, Australia
yanwang@science.mq.edu.au

Abstract

In Serviced-Oriented Computing (SOC) environments, the trust level of a service or a service provider is a critical issue for a service client to consider, particularly when the client is looking for a service from a large set of services or service providers. However, a service may invoke other services offered by different providers forming composite services. The complex invocation relations significantly increase the complexity of trust evaluation in composite services. In this paper, we propose a novel algorithm for trust evaluation in composite services that takes all atomic invocations into account, which is essential for composite services selection and discovery.

Keywords: trust, trust evaluation, composite services

1. Introduction

In recent years, Service-Oriented Computing (SOC) has emerged as an increasingly important research area attracting much attention from both the research and industry communities. In SOC applications, a variety of services across domains are provided to clients in a loosely-coupled environment. Clients can look for preferred and qualified services via the discovery service of registries, invoke and receive services from the rich service environments [4].

In SOC, a service can refer to a transaction, such as selling a product online (i.e. the traditional online service), or a functional component implemented by Web services technologies [4]. However, when a client looks for a service from a large set of services offered by different providers, in addition to functionality, the reputation-based trust is also a key factor for services selection. It is also a critical task for service registries to be responsible for maintaining the list of reputable and trustworthy services and service providers, and bringing them to clients [5].

Trust is the measure by one party on the willingness and ability of another party to act in the interest of the former party in a situation [2]. Trust is also the probability by which, party A expects that another party B performs a given action if the trust value is in the range of [0,1] [1].

The trust issue has been widely studied in many applications. In e-commence environments, the trust management system can provide valuable information to buyers and prevent some typical attacks [6, 11]. In Peer-to-Peer information-sharing networks, binary ratings work pretty well as a file is either the definitively correct version or not [1, 8, 9]. In SOC environments, an effective trust management system is critical to identify potential risks, provide objective trust results to clients and prevent malicious service providers from easily deceiving clients and leading to their huge monetary loss [7].

However, trust management is a very complex issue in SOC environments. To satisfy the same specified functionality requirement, a service may have to invoke other services forming composite services with complex invocations and trust dependencies among services and service providers [3]. Meanwhile, given a set of various services, different compositions may lead to different trust values.

Though there are a variety of trust evaluation methods existing in different areas, no proper mechanism exists for evaluating the global trust of a composite service from the trust values of all service components. In this paper, we propose a novel algorithm for global trust evaluation in composite services, which is essential for composite services selection and discovery.

This paper is organized as follows. Section 2 presents the service invocation model. Section 3 proposes a novel global trust evaluation algorithm for composite services selection and discovery. An example of trust evaluation is presented in Section 4. Finally Section 5 concludes our work.

2. Service Invocation Model

In this section, the service invocation model is proposed to represent the composite services. In Section 2.1, the invocation relations in composite services are presented, after which a composite services example is introduced in Section 2.2.

2.1. Invocation Relations in Composite Services

A *composite service* is a conglomeration of services with invocation relations between them. Six atomic invocations [3, 10] are depicted as follows and in Fig. 1.

- *Sequential Invocation*: A service S invokes its unique successing service A. It is denoted as $\mathsf{Se}(S : A)$ (see Fig. 1(a)).
- *Parallel Invocation*: A service S invokes its successing services in parallel. E.g., if S has successors A and B, it is denoted as $\mathsf{Pa}(S : A, B)$ (see Fig. 1(b)).

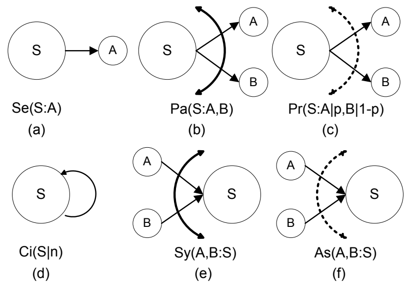

Figure 1. Atomic invocations

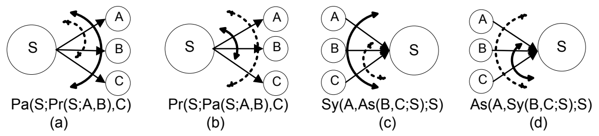

Figure 2. Complex invocations example

- *Probabilistic Invocation*: A service S invokes its successing service with a probability. E.g., if S has successors A with the probability p and B with the probability $1-p$, it is denoted as $\mathsf{Pr}(S:A|p,B|1-p)$ (see Fig. 1(c)).
- *Circular Invocation*: A service S invokes itself for n times. It is denoted as $\mathsf{Ci}(S|n)$ (see Fig. 1(d)). A circular invocation can be unfolded by cloning the service vertices involved in the cycle as many times as the cycle count [10].
- *Synchronous Activation*: A service S is activated only when all its predecessing services have completed. E.g., if S has synchronous predecessors A and B, it is denoted as $\mathsf{Sy}(A,B:S)$ (see Fig. 1(e)).
- *Asynchronous Activation*: A service S is activated as the result of the completion of one of its predecessing services. E.g., if S has asynchronous predecessors A and B, it is denoted as $\mathsf{As}(A,B:S)$ (see Fig. 1(f)).

With atomic invocations, some complex invocations can be depicted as Fig. 2, which are not clearly introduced in the existing works.

- *Probabilistic inlaid parallel invocation*, denoted as $\mathsf{Pa}(S:\mathsf{Pr}(S:A|p,B|1-p),C)$.
- *Parallel inlaid probabilistic invocation*, denoted as $\mathsf{Pr}(S:\mathsf{Pa}(S:A,B)|p,C|1-p)$.
- *Asynchronous inlaid synchronous activation*, denoted as $\mathsf{Sy}(A,\mathsf{As}(B,C:S):S)$.
- *Synchronous inlaid asynchronous activation*, denoted as $\mathsf{As}(A,\mathsf{Sy}(B,C:S):S)$.

2.2. An Composite Services Example

Here we introduce an example of composite services. In this example, with a starting service *START* and an ending service *END*, the composite services consisting of all possibilities of the invocation flows can be depicted by a *service invocation graph* (*SIG*) (see Fig. 3). One of all invocation flows from service *START* to service *END* is a *service execution flow* (*SEF*) as depicted in Fig. 4.

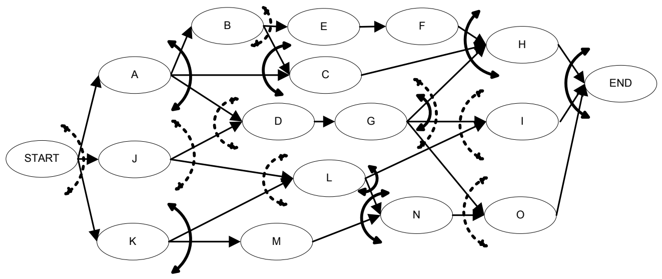

Figure 3. The service invocation graph example

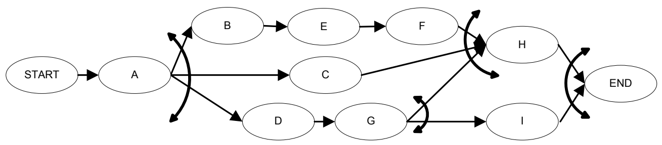

Figure 4. A service execution flow example

When a client looks for the optimal *SEF* with the maximal global trust value from multiple ones in an *SIG*, a proper mechanism is necessary for evaluating the global trust of an *SEF* from the trust values of all service components, which will be introduced in the next section.

3. Trust Evaluation in Composite Services

The global trust value of *SEF* is determined by the trust values of vertices and invocation relations between vertices in the *SEF*.

There are two kinds of atomic structures to determine the trust value of an *SEF*: Se (Fig. 1(a)) and Pa (Fig. 1 (b)). An Se in the *SEF* can be selected from the service invocation relation Se (Fig. 1(a)) or Pr (Fig. 1(c)) in the *SIG*. A Pa in the *SEF* can be selected from the service invocation relation Pa (Fig. 1 (b)) in the *SIG*.

With Se and Pa, Sy in an *SEF* can be determined. Since an *SEF* is an end-to-end graph, if in the *SEF* there is a Pa, with which a service invokes its successing services in parallel, there must be an Sy, with which a service is activated by its predecessing services in parallel (see Fig. 6(a)). Due to space constraint, the details are omitted.

3.1. Global Trust Evaluation of Se

Considering an Se structure (see Fig. 5 (a)), since S and A are independent, the probability that both S and A occur is equal to the product of the probability that S occurs and the probability that A occurs. When the trust value is taken as a probability [1], we have the following definition.

Definition 1: The global trust value T_g of an Se structure where service S uniquely invokes service A (see Fig. 5 (a)) can be computed by

$$T_g = T_S \cdot T_A, \tag{1}$$

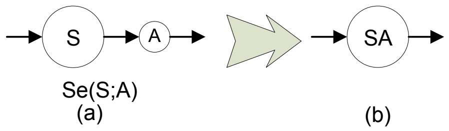

Figure 5. Se invocation

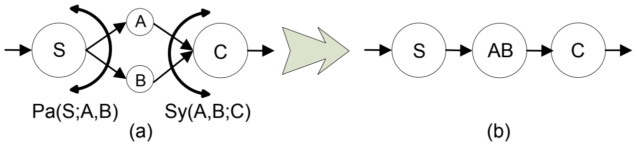

Figure 6. Pa invocation

where T_S and T_A are the trust values of S and A respectively.

The global trust value of an Se (see Fig. 5 (a)) can be taken as the trust value of a new vertex SA (see Fig. 5 (b)), which is merged from vertices S and A.

3.2. Global Trust Evaluation of Pa

Definition 2: The global trust value T_g of a Pa structure where service S invokes services A and B in parallel (see Fig. 6 (a)) can be computed from T_S and the merged trust value T_{AB} by Definition 1, and

$$T_{AB} = \frac{\omega_1}{\omega_1 + \omega_2} \cdot T_A + \frac{\omega_2}{\omega_1 + \omega_2} \cdot T_B, \qquad (2)$$

where T_S, T_A and T_B are the trust values of S, A and B respectively. ω_1 and ω_2 are weights for A and B respectively which are specified in a requesting client's preference or specified as the default values by the service trust management authority.

Based on the above computation, in a Pa structure (see Fig. 6(a)), vertices A and B can be merged as a new vertex AB (see Fig. 6 (b)) with trust value T_{AB}, leading to Se structures where S uniquely invokes AB and AB uniquely invokes C (see Fig. 6 (b)). The global trust value of an Se structure is computed according to Definition 1. Therefore, the global trust value of Pa can be evaluated.

3.3. Global Trust Evaluation Algorithm of *SEF*

According to Definitions 1 & 2, each atomic structure Se or Pa can be converted to a single vertex. Hence, in the process of trust evaluation, since an *SEF* only consists of Se and Pa structures, an *SEF* can be incrementally converted to a single vertex with its trust value taken as the global trust value of the *SEF*. Therefore, the global trust evaluation of *SEF* algorithm have the following steps:

Step 1 The trust value of each atomic Se structure in the *SEF* is evaluated based on Definition 1. Each evaluated atomic Se structure is taken as a vertex in the *SEF*.

Step 2 The trust value of each atomic Pa structure is evaluated based on Definition 2. Each evaluated atomic Pa structure is then taken as a vertex in the *SEF*.

Step 3 If the *SEF* contains more than one vertex, go to Step 1. Otherwise, the trust value of the single vertex is the global one.

The details of global trust evaluation of *SEF* are illustrated in Algorithm 1.

Algorithm 1 Global Trust Evaluation Algorithm of *SEF*

```
Input: an SEF, trust value for each vertex.
Output: the global trust value of SEF $T_{global}$.
1: let the starting service of SEF be root, and the ending service of SEF be terminal;
2: while there is more than one vertices in SEF do
3:     initialize vector $Container$ to contain root;
4:     while $Container \neq \emptyset$ do
5:         select a vertex $v$ in $Container$;
6:         remove $v$ from $Container$;
7:         let vectors $Se$ and $Pa$ be the Se and Pa structures from $v$;
8:         if vector $Se \neq \emptyset$ then
9:             if only $v$ invokes $Se$ then
10:                // global trust evaluation of Se (lines 11-17)
11:                let vSe be the vertex which is merged from $v$ and $Se$;
12:                change the predecessors of $v$ to those of vSe;
13:                change the successors of $Se$ to those of vSe;
14:                remove all the edges to $v$ in SEF;
15:                remove all the edges from $Se$ in SEF;
16:                let the weight of $v$ be that of vSe;
17:                let $T_{vSe}$ be the trust value of vSe based on Definition 1;
18:                $T_{global} \leftarrow T_{vSe}$
19:                add vSe into $Container$;
20:            else
21:                if $Se$ is not terminal and $Se$ is not in $Container$ then
22:                    add $Se$ into $Container$;
23:                end if
24:            end if
25:        end if
26:        if vector $Pa \neq \emptyset$ then
27:            for all $Pa(i)$ in $Pa$ do
28:                if $Pa(i)$ is not terminal and $Pa(i)$ is not in $Container$ then
29:                    add $Pa(i)$ into $Container$;
30:                end if
31:            end for
32:            for all $Pa(i)$ in $Pa$ do
33:                let $Se_i$ and $Pa_i$ be the Se and Pa structures from $Pa(i)$;
34:                for all $Pa(j)$ in $Pa$ and $j > i$ do
35:                    let $Se_j$ and $Pa_j$ be the Se and Pa structures from $Pa(j)$;
36:                    if $Se_i$=$Se_j$ and $Pa_i = \emptyset$ and $Pa_j = \emptyset$ then
37:                        // global trust evaluation of Pa (lines 38-44)
38:                        let vPa be the vertex merged from $Pa(i)$ and $Pa(j)$;
39:                        change the successors of $Se_i$ to those of vPa;
40:                        change the predecessors of $Pa(i)$ and $Pa(j)$ to those of vPa;
41:                        remove all the edges from $v$ to $Pa(i)$ and $Pa(j)$;
42:                        remove all the edges from $Pa(i)$ and $Pa(j)$ to $Se_i$;
43:                        let the sum of weights of $Pa(i)$ and $Pa(j)$ be that of vPa;
44:                        let $T_{vPa}$ be the trust value of vPa based on Definition 2;
45:                        $T_{global} \leftarrow T_{vPa}$
46:                        if $Pa_i$ or $Pa_j$ is in $Container$ then
47:                            remove $Pa_i$ or $Pa_j$ from $Container$;
48:                        end if
49:                    end if
50:                end for
51:            end for
52:        end if
53:    end while
54: end while
55: return $T_{global}$
```

3.4. Composite Services Selection and Discovery

In the literature, the exhaustive search method is used to enumerate all *SEF*s in a composite service [3]. Then the trust values of *SEF*s can be evaluated according to Algorithm 1. After comparing these trust values, the optimal

Table 1. Trust values of each service in the example

Service	*START*	*A*	*B*	*C*	*D*	*E*
Trust value	1	0.7	0.9	0.8	0.8	0.9
Service	*F*	*G*	*H*	*I*	*END*	
Trust value	0.7	0.9	0.8	0.9	1	

Table 2. Weights of service components in Pa

B	*C*	*D*	*H*	*I*
0.1	0.3	0.6	0.6	0.4

SEF with the maximal global trust value can be discovered. Since the composite service selection and discovery is an NP-complete problem [10], to improve the efficiency, a polynomial approximation algorithm is expected to find the optimal *SEF* with the maximal global trust value.

4. An Example of Trust Evaluation

In this section, taking the *SEF* in Fig. 4 as an example, we will illustrate how our proposed global trust evaluation algorithm works. The corresponding trust values of each service component are listed in Table 1. The weights of service components in all Pa structures of the composite services are listed in Table 2.

The evaluation process of Algorithm 1 is as follows. Taking Fig. 7 (a) as an example, firstly, *B*, *E* and *F* form Se structures, and they are merged as *BEF* with $T_{BEF} = 0.567$ based on Definition 1. Similarly, *START* and *A* are merged as *STARTA* with $T_{STARTA} = 0.7$, and *D* and *G* are merged as *DG* with $T_{DG} = 0.72$. So Fig. 7 (b) is obtained, where *STARTA*, *BEF* and *C* form a Pa structure, and *BEF*, *C* and *H* form an Sy structure. Then, *BEF* and *C* are merged as *BCEF* with $T_{BCEF} = 0.7414$ based on Definition 2 (Fig. 7 (c)). Similarly, *H* and *I* are merged as HI with $T_{HI} = 0.84$ (Fig. 7 (d)). Because *HI* and *END* form an Se structure, they are merged as *HIEND* with $T_{HIEND} = 0.84$. After that, as Fig. 7 (e) has a Pa structure, we obtain the merged vertex *BCDEFG* with $T_{BCDEFG} = 0.7287$ (Fig. 7 (f)). Since the *SEF* in Fig. 7 (f) only consists of Ses, the final vertex is obtained and $T_{global} = T_{STARTA} \cdot T_{BCDEFG} \cdot T_{HIEND} = 0.4285$ (Fig. 7 (g)).

5. Conclusions

When a client looks for the optimal *SEF* with the maximal global trust value from multiple ones in an *SIG*, a proper mechanism is necessary for evaluating the global trust of an *SEF* from the trust values of all service components.

There are only two kinds of atomic structures Se and Pa in an *SEF*, and each of them can be converted to a single vertex by our global trust evaluation algorithm. Hence, in the process of trust evaluation, an *SEF* can be incrementally converted to a single vertex with its trust value taken as the global trust value of the *SEF*. Therefore, our proposed algorithm can compute the global trust value of an *SEF*, which is essential for composite services selection and discovery.

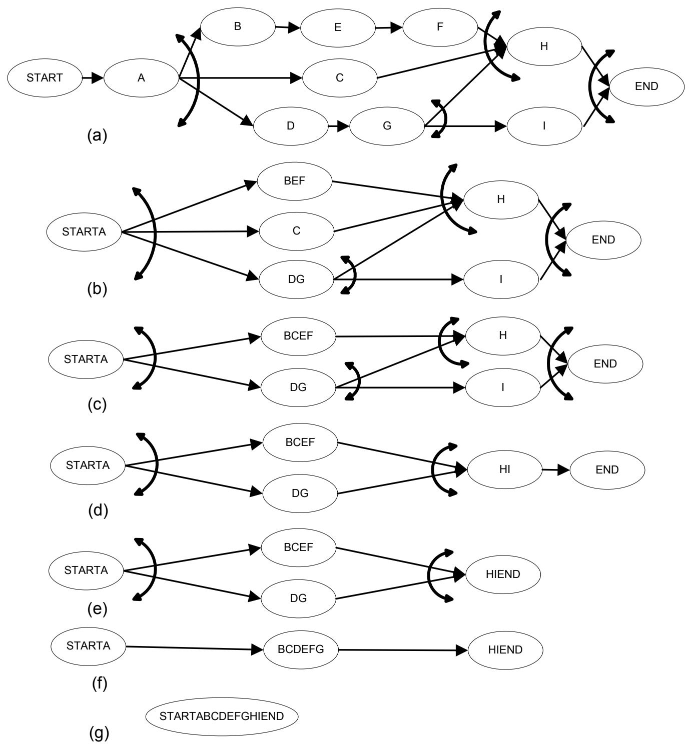

Figure 7. The evaluation process of Algorithm 1

References

[1] A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, 2007.

[2] D. H. Knight and N. L. Chervany. The meaning of trust. Technical Report WP9604, University of Minnesota, Management Information Systems Research Center, 1996.

[3] D. A. Menascé. Composing web services: A qos view. *IEEE Internet Computing*, 8(6):88–90, 2004.

[4] M. P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann. Service-oriented computing: a research roadmap. *Int. J. Cooperative Inf. Syst.*, 17(2):223–255, 2008.

[5] L.-H. Vu, M. Hauswirth, and K. Aberer. QoS-based service selection and ranking with trust and reputation management. In *CoopIS 2005*, pages 466–483.

[6] Y. Wang and E.-P. Lim. The evaluation of situational transaction trust in e-service environments. In *ICEBE 2008*, pages 265–272.

[7] Y. Wang, K.-J. Lin, D. S. Wong, and V. Varadharajan. Trust management towards service-oriented applications. *Service Oriented Computing and Applications journal*, 3(1), 2009.

[8] L. Xiong and L. Liu. Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Trans. Knowl. Data Eng.*, 16(7):843–857, 2004.

[9] B. Yu, M. P. Singh, and K. Sycara. Developing trust in large-scale peer-to-peer systems. *2004 IEEE First Symposium on Multi-Agent Security and Survivability*, pages 1–10, 2004.

[10] T. Yu, Y. Zhang, and K.-J. Lin. Efficient algorithms for web services selection with end-to-end qos constraints. *TWEB*, 1(1), 2007.

[11] G. Zacharia and P. Maes. Trust management through reputation mechanisms. *Applied Artificial Intelligence*, 14(9):881–907, 2000.