11-2009

Trust-Oriented Composite Services Selection and Discovery

Lei LI

YAN WANG
Macquarie University

Ee Peng LIM
Singapore Management University, eplim@smu.edu.sg

DOI: https://doi.org/10.1007/978-3-642-10383-4_4

Citation

LI, Lei; WANG, YAN; and LIM, Ee Peng. Trust-Oriented Composite Services Selection and Discovery. (2009). *Proceedings of the 7th International Joint Conference on Service-Oriented Computing*. 50-67. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/494

Trust-Oriented Composite Service Selection and Discovery

Lei Li[1], Yan Wang[1], and Ee-Peng Lim[2]

[1] Department of Computing, Macquarie University, Sydney, Australia 2109
[2] School of Information Systems, Singapore Management University, Singapore 178902
`{leili,yanwang}@science.mq.edu.au,`
`eplim@smu.edu.sg`

Abstract. In Service-Oriented Computing (SOC) environments, service clients interact with service providers for consuming services. From the viewpoint of service clients, the trust level of a service or a service provider is a critical issue to consider in service selection and discovery, particularly when a 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 invocations in composite services greatly increase the complexity of trust-oriented service selection and discovery. In this paper, we propose novel approaches for composite service representation, trust evaluation and trust-oriented service selection and discovery. Our experiments illustrate that compared with the existing approaches our proposed trust-oriented service selection and discovery algorithm is realistic and more efficient.

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 a discovery service of registries, invoke and receive services from the rich service environments [18].

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 service technologies [18]. 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 service 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 [19].

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 [11]. Trust is also the subjective 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] [8].

Different from P2P information-sharing networks or eBay reputation management system, where a binary rating system is used [25], in SOC environments, a trust rating is usually a value in the range of [0,1] [19,20,21] given by a service client, representing

L. Baresi, C.-H. Chi, and J. Suzuki (Eds.): ICSOC-ServiceWave 2009, LNCS 5900, pp. 50–67, 2009.

the subjective belief of the service client on the satisfaction of a service or a service provider. The trust value of a service or a service provider can be calculated by a trust management authority based on the collected trust ratings representing the reputation of the service or the service provider.

However, trust management is a very complex issue in SOC environments. To satisfy the specified functionality requirement, a service may have to invoke other services forming composite Web services with complex invocations and trust dependencies among services and service providers [16]. Meanwhile, given a set of various services, different compositions may lead to different service structures. Although these certainly enrich the service provision, they greatly increase the computation complexity and thus make trustworthy service selection and discovery a very challenging task.

In the literature, there are some existing studies for service composition and quality driven service selection [3,16,24,28,30]. However, for trust-oriented composite service selection and discovery, some research problems remain open.

1. The definition of a proper graph representation of composite services including both probabilistic invocations and parallel invocations is still lacking. The corresponding data structure is also essential. Both of them are fundamental and important for deploying the global trust evaluation of composite services.
2. From the definitions in [8,11], trust can be taken as the *subjective probability*, i.e. *the degree of belief an individual has in the truth of a proposition* [4,5], rather than the *objective probability* or *classical probability*, which is *the occurrence frequency of an event* [5]. A subjective probability is derived from an individual's personal judgment about a specific outcome (e.g. the evaluation of teaching quality or service quality). It differs from person to person. Hence, the classical probability theory does not fit for trust evaluation. Instead, *subjective probability theory* deals with *subjective probability* [4,5] and should be adopted for trust evaluation.
3. Although there are a variety of trust evaluation methods in different areas [19,21,25], no proper mechanism exists for evaluating the global trust of a composite service with a complex structure from the trust values of all service components.
4. Taking trust evaluation and the complex structure of composite services into account, effective algorithms are needed for composite service selection and discovery, and are expected to be more efficient than the existing approaches [16,28].

In this paper, we first present the service invocation graph and service invocation matrix for composite service representation. In addition, we propose a trust evaluation method for composite services based on Bayesian inference, which is an important component in subjective probability theory. Furthermore, we propose a service selection and discovery algorithm based on Monte Carlo method. Experiments have been conducted on composite services with various sizes to compare the proposed model with the existing exhaustive search method [16]. The results illustrate that our proposed algorithm is realistic and more efficient.

This paper is organized as follows. Section 2 reviews existing studies in service composition, service selection and trust. Section 3 presents our proposed composite services oriented service invocation graph and service invocation matrix. Section 4 presents a novel trust evaluation method for composite services. In Section 5, a Monte Carlo method based algorithm is proposed for trust-oriented composite service selection and discovery. Experiments are presented in Section 6 for further illustrating the properties of our models. Finally Section 7 concludes our work.

2 Related Work

In SOC environments, the composition of services offered by different providers enriches service provision and offers flexibility to service applications. In [14,15], Medjahed et al present some frameworks and algorithms for automatically generating composite services from specifications and rules.

In real applications, the criteria of searching services should take into account not only functionalities but also other properties, such as QoS (quality of service) and trust. In the literature, a number of QoS-aware Web service selection mechanisms have been developed, aiming at QoS improvement in composite services [3,24,30]. In [30], Zeng et al present a general and extensible model to evaluate the QoS of composite services. Based on their model, a service selection approach has been introduced using linear programming techniques to compute optimal execution plans for composite services. The work in [3] addresses the selection and composition of Web services based on functional requirements, transactional properties and QoS characteristics. In this model, services are selected in a way that satisfies user preferences, expressed as weights over QoS and transactional requirements. In [24], Xiao et al present an autonomic service provision framework for establishing QoS-assured end-to-end communication paths across domains. Their algorithms can provide QoS guarantees over domains. The above works have their merits in different aspects. However, none of them has taken parallel invocation into account, which is fundamental and one of the most common existing invocations in composite services [16,28].

Menascé [16] adopts an exhaustive search method to measure service execution time and cost involving probabilistic, parallel, sequential and fastest-predecessor-triggered invocations. However, the algorithm complexity is exponential. Yu et al [28] study the service selection problem with multiple QoS constraints in composite services, and propose two optimal heuristic algorithms: the combinatorial algorithm and the graph-based algorithm. The former one models the service selection as a multidimension multichoice 0-1 knapsack problem. The latter one can be taken as a multiconstraint optimal path problem. Nevertheless, none of these works addresses any aspect of trust.

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 [20,29]. In Peer-to-Peer information-sharing networks, binary ratings work pretty well as a file is either the definitively correct version or not [27]. 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 [19].

In general, the trust from a service client on a service or a service provider can be taken as an extent with which the service client *believes* that the service provider can satisfy the client's requirement with desirable performance and quality. Thus, as we point out in Section 1, trust is a *subjective belief* and it is better to adopt *subjective* *probability theory* [5] to deal with trust. In contrast, *classical probability theory* is actually more suitable to deal with objective occurrence frequency of an event.

There are some works to deal with subjective ratings [7,22]. Jøsang [7] describes a framework for combining and assessing subjective ratings from different sources based on Dempster-Shafer belief theory. Wang and Singh [22] set up a bijection from subjective ratings to trust values with a mathematical understanding of trust in a variety of multiagent systems. However, their models use either a binary rating (positive or negative) system or a triple rating (positive, negative or uncertain) systems that are more suitable for security-oriented or P2P file-sharing trust management systems.

As pointed in [27], in richer service environments such as SOC or e-commerce, a rating in $[0, 1]$ is more suitable. In [26], Xu et al propose a reputation-enhanced QoS-based Web service discovery algorithm for service matching, ranking and selection based on existing Web service technologies. Malik et al [13] propose a set of decentralized techniques aiming at evaluating reputation-based trust with the ratings from peers to facilitate trust-based selection and composition of Web services. However, in these works, no service invocation and composite service structure are taken into account. Taking the complex structure of composite services into account, effective algorithms are needed for trust-oriented composite service selection and discovery.

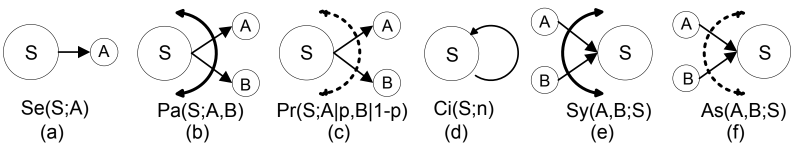
3 Service Invocation Model

In this section, we present the definitions of our proposed service invocation graph and service invocation matrix for representing the complex structures of composite services. They are essential for our trust-oriented composite service selection and discovery algorithm to be introduced in Section 5.

3.1 Composite Services and Invocation Relation

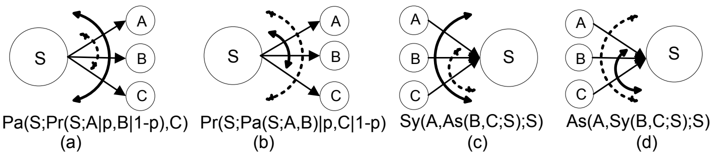
A *composite service* is a conglomeration of services with invocation relations between them. Six atomic invocations [16,28] are depicted as follows and in Fig. 1.

- *Sequential Invocation*: A service S invokes its unique succeeding service A. It is denoted as $\mathsf{Se}(S : A)$ (see Fig. 1(a)).
- *Parallel Invocation*: A service S invokes its succeeding 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)).
- *Probabilistic Invocation*: A service S invokes its succeeding service with a probability. E.g., if S invokes 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 itself n times [28]. Hence, it can be replaced by Se in advance.
- *Synchronous Activation*: A service S is activated only when all its preceding services have been 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 preceding services. E.g., if S has asynchronous predecessors A and B, it is denoted as $\mathsf{As}(A, B : S)$ (see Fig. 1(f)).



Se(S;A) (a) Pa(S;A,B) (b) Pr(S;A|p,B|1-p) (c) Ci(S;n) (d) Sy(A,B;S) (e) As(A,B;S) (f)

Fig. 1. Atomic invocations



Pa(S;Pr(S;A|p,B|1-p),C) (a) Pr(S;Pa(S;A,B)|p,C|1-p) (b) Sy(A,As(B,C;S);S) (c) As(A,Sy(B,C;S);S) (d)

Fig. 2. Complex invocations examples

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 $\mathsf{As}(A,\mathsf{Sy}(B,C:S):S)$.

3.2 An Example: Travel Plan

Here we introduce an example of composite services.

Smith in Sydney, Australia is making a travel plan to attend an international conference in Stockholm, Sweden. His plan includes conference registration, airline from Sydney to Stockholm, accommodation and local transportation.

Regarding conference registration *Reg*, Smith could pay *Online* or by *Fax* with a credit card *Ccard*. Regarding accommodation reservation *Acc*, Smith could make a reservation at Hotel *Ha*, *Hb* or *Hc* with credit card *Ccard*. According to the hotel choice, Smith could arrange the local transportation, e.g. take a *Taxi* to *Ha*, take a *Taxi* or a *Bus* to either *Hb* or *Hc*. Regarding airplane booking *Air*, Smith could choose from Airlines *Aa*, *Ab* and *Ac* with the credit card *Ccard* for the payment. Smith chooses the services according to their trust values. He will have a higher probability to choose the service with a better trust value.

In this example, with a starting service *START* and an ending service *END*, the composite services consisting of all possibilities of the travel plan can be depicted by a service invocation graph (*SIG*) (Fig. 3). One of all feasible travel plans is a service execution flow as depicted in Fig. 4.

3.3 Service Invocation Graph

The structure of a composite service can be represented by a service invocation graph (*SIG*), with the initial definition as follows.

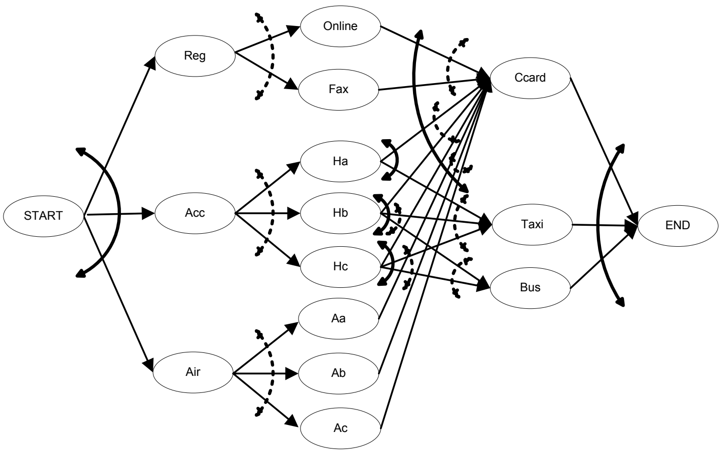


Fig. 3. The *SIG* for the travel plan of Smith

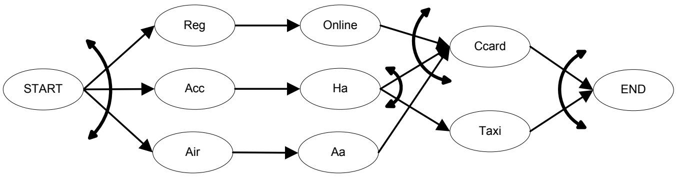


Fig. 4. A service execution flow

Definition 1. The *service invocation graph* (*SIG*) is a directed graph $G = (V, E, R)$, where V is a finite set of vertices, E is a finite set of directed edges and R is the set of atomic invocations Se, Pa, Pr, Ci, Sy and As. In G, each vertex $v \in V$ represents a service. $\forall e = (v_1, v_2) \in E$ $(v_1, v_2 \in V)$ is a directed edge, where v_1 is the *invoking vertex* and v_2 is the *invoked vertex*. Here v_1 is the *direct predecessor* of v_2 and v_2 is the *direct successor* of v_1. It is denoted as $v_1 \succeq v_2$.

Definition 2. Given a service invocation graph $G = (V,\ E, R)$, vertex $v_2 \in V$ is *invocational* from vertex $v_1 \in V$ if $(v_1, v_2) \in E$ or there is a directed path P in G where v_1 is the staring vertex and v_2 is the ending vertex. If v_2 is invocational from v_1, it is denoted as $v_1 \succ v_2$.

In addition, if $v_1 \succ v_2$, v_1 is the predecessor of v_2 and v_2 is the successor of v_1. Obviously, the *invocational* relation is transitive, i.e. if $v_1 \succ v_2$, $v_2 \succ v_3$, then $v_1 \succ v_3$.

Definition 3. In a service invocation graph, the *service invocation root* is the entry vertex without any predecessors, and the *service invocation terminal* is the exit vertex without any successors.

Based on the above definitions, *SIG* is well-defined as follows.

Definition 4. A composite service can be represented by a *service invocation graph*

$$SIG = (V, I_p, R_p, I_s, R_s), \tag{1}$$

where

- In an *SIG*, there are only one service invocation root *START* and only one service invocation terminal *END*;
- $V = \{v_i | v_i$ is a vertex, $v_i =$*START* or *START* $\succ v_i\}$;
- $I_p = \{I_{p_i}\}$ and I_{p_i} is a set of direct predecessors invoking v_i, i.e. $I_{p_i} = \{p_{ij} | p_{ij}, v_i \in V$ and $p_{ij} \succeq v_i\}$;
- R_p represents a set of activation relations between I_p and V, which includes atomic activations Sy and As;
- $I_s = \{I_{s_i}\}$ and I_{s_i} is a set of direct successors invoked by v_i, i.e. $I_{s_i} = \{s_{ij} | v_i, s_{ij} \in V$ and $v_i \succeq s_{ij}\}$;
- R_s represents a set of invocation relations between V and I_s, which includes atomic invocations Se, Pa, Pr and Ci.

Let $\emptyset$ denote the empty invocation relation set. In an *SIG*, if $I_{p_i} = \emptyset$, then $v_i =$ *START*. Similarly, if $I_{s_i} = \emptyset$, then $v_i =$ *END*.

Definition 5. A *service execution flow* (*SEF*) of an *SIG G* is a graph $G' = (V', E', R')$, where R' contains Se, Pa, Sy and Ci, $V' \subseteq V$ and $E' \subseteq E$. In addition, $\forall v' \in V'$, v' is invocational from service invocation root *START* of G, and service invocation terminal *END* of G is invocational from v'.

3.4 Service Invocation Matrix

In Section 3.3, *SIG* provides a clear picture of service invocation relations in composite services. However, an underneath data structure is essential to represent and store vertices and invocation relations. Here we propose *service invocation matrix* - an algebraic representation of composite services.

Definition 6. A composite service can be represented by a *service invocation matrix*

$$SIM = (M_{ij})_{1 \leq i \leq n, 1 \leq j \leq n}, \tag{2}$$

where

- n is the number of vertices in the composite services;
- $M_{ij} = 0$ *iff* there is no invocation from vertex i to vertex j;
- $M_{ij} =< M_{ij}^{(1)}, M_{ij}^{(2)}, \ldots, M_{ij}^{(k)} > (i \neq j)$ represents the invocations from vertex i to vertex j, and k is the number of all invocations from i to j;
- $M_{ij}^{(h)}$ $(1 \leq h \leq k)$ is an integer which represents an invocation type from vertex i to vertex j;
  - If it is a parallel invocation, $M_{ij}^{(h)} = 2m_1$ $(m_1 = 1, 2 \ldots)$, where m_1 increases from 1 continuously and different m_1 values indicate different parallel invocations Pas;

  - If it is a probabilistic invocation, $M_{ij}^{(h)} = 2m_2 - 1$ $(m_2 = 1, 2 \ldots)$, where m_2 increases from 1 continuously and different m_2 values indicate different probabilistic invocations Prs;
- M_{ii} is an integer to represent the number of circular times of Ci in vertex i.

According to Definition 6, we have the following property.

Property 1. $< M_{ij}^{(1)}, M_{ij}^{(2)} > = < M_{ij}^{(2)}, M_{ij}^{(1)} >$

Taking Travel Plan (Fig. 3) in Section 3.2 as an example, non-zero entities of the *SIM* are listed in Table 1. Our proposed *SIM* can cover all atomic invocation structures and the complex invocation structures derived from them.

Table 1. Non-zeros of *SIM* in Travel Plan

i	j	M_{ij}	i	j	M_{ij}	i	j	M_{ij}	i	j	M_{ij}
START	Reg	$< 2 >$	Air	Ab	$< 1 >$	Hc	Ccard	$< 2 >$	Fax	Ccard	$< 1 >$
START	Acc	$< 2 >$	Air	Ac	$< 1 >$	Hc	Taxi	$< 2, 1 >$	Ha	Ccard	$< 1 >$
START	Air	$< 2 >$	Reg	Online	$< 1 >$	Hc	Bus	$< 2, 1 >$	Ha	Taxi	$< 1 >$
Acc	Ha	$< 1 >$	Reg	Fax	$< 1 >$	Aa	Ccard	$< 1 >$	Ccard	END	$< 1 >$
Acc	Hb	$< 1 >$	Hb	Ccard	$< 2 >$	Ab	Ccard	$< 1 >$	Taxi	END	$< 1 >$
Acc	Hc	$< 1 >$	Hb	Taxi	$< 2, 1 >$	Ac	Ccard	$< 1 >$	Bus	END	$< 1 >$
Air	Aa	$< 1 >$	Hb	Bus	$< 2, 1 >$	Online	Ccard	$< 1 >$			

4 Trust Evaluation in Composite Services

In this section, we introduce our trust evaluation models for composite services. In Section 4.1, a trust estimation model is proposed to estimate the trust value of each service component from a series of ratings according to Bayesian inference[4,5], which is an important component in subjective probability theory. These ratings are provided by service clients and stored by the service trust management authority. In Section 4.2, a global trust computation model is proposed to compute the global trust value of a composite service based on the trust values of all service components.

4.1 Trust Estimation Model

Since subjective probability is a person's degree of belief concerning a certain event [4,5], the trust rating in $[0, 1]$ of a service given by a service client can be taken as the *subjective possibility* with which the service provider can perform the service satisfactorily. Hence, *subjective probability theory* is the right tool for dealing with trust ratings. In this paper, we adopt *Bayesian inference*, which is an important component in *subjective probability theory*, to estimate the trust value of a provided service from a set of ratings. Each rating is a value in $[0, 1]$ evaluated from the subjective judgements of a service client on multiple attributes of the provided service, such as availability, security, execution time and cost [8,23].

The primary goal of *Bayesian inference* [4,5] is to summarize the available information that defines the distribution of trust ratings through the specification of probability density functions, such as: prior distribution and posterior distribution. The *prior distribution* summarizes the subjective information about the trust prior to obtaining the ratings sample $x_1, x_2, \ldots, x_n$. Once the sample is obtained, the prior distribution can be updated. The updated probability distribution on trust ratings is called the *posterior distribution*, because it reflects probability beliefs posterior to analyzing ratings.

According to [6], if all service clients give ratings for the same service, the provided ratings conform to normal distribution. The complete set of ratings can be collected based on honest-feedback-incentive mechanisms [9,10]. Let μ and σ denote the mean and the variance of ratings in the normal distribution. Thus, a sample of ratings $x_1, x_2, \ldots, x_n$ ($x_i \in [0,1]$) has the normal density with mean μ and variance σ. In statistics, when a ratings sample with size n is drawn from a normal distribution with mean μ and variance σ, the mean of the ratings sample also conforms to a normal distribution which has mean μ and variance $\sigma/\sqrt{n}$ [4]. Let $\delta \in [0,1]$ denote the prior subjective belief about the trust of a service that a client is requesting for. We can assume that the prior normal distribution of μ has mean δ and variance $\sigma/\sqrt{n}$, i.e.

$$f(\mu) = \begin{cases} \frac{\sqrt{n}}{\sigma\sqrt{2\pi}} e^{\frac{n(\mu-\delta)^2}{-2\sigma^2}}, & 0 < \mu < 1; \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Given μ, the joint conditional density of the ratings sample is

$$f(x_1, x_2, \ldots, x_n|\mu) = \frac{1}{\sigma^n (2\pi)^{\frac{n}{2}}} e^{\frac{\Sigma(x_i-\mu)^2}{-2\sigma^2}} = \frac{1}{\sigma^n (2\pi)^{\frac{n}{2}}} e^{\frac{\Sigma x_i^2 - 2\mu \Sigma x_i + n\mu^2}{-2\sigma^2}}. \tag{4}$$

Hence, the joint density of the ratings sample and μ is

$$f(x_1, \ldots, x_n; \mu) = \frac{\sqrt{n}}{\sigma^{n+1} (2\pi)^{\frac{n+1}{2}}} e^{\frac{\Sigma x_i^2 - 2\mu n\bar{x} + n\mu^2 + n(\mu-\delta)^2}{-2\sigma^2}}. \tag{5}$$

Based on Eq. (5), the marginal density of the ratings sample is

$$\begin{aligned} f(x_1, x_2, \ldots, x_n) &= \frac{\sqrt{n}}{\sigma^{n+1} (2\pi)^{\frac{n+1}{2}}} e^{\frac{\Sigma x_i^2 + n\delta^2}{-2\sigma^2}} \int_{-\infty}^{\infty} e^{\frac{n\mu^2 - (n\bar{x}+n\delta)\mu}{-\sigma^2}} \mathrm{d}\mu \\ &= \frac{\sqrt{n}}{\sigma^{n+1} (2\pi)^{\frac{n+1}{2}}} e^{\frac{\Sigma x_i^2 + n\delta^2 - \frac{n(\bar{x}+\delta)^2}{2}}{-2\sigma^2}} \int_{-\infty}^{\infty} e^{\frac{n(\mu - \frac{\bar{x}+\delta}{2})^2}{-\sigma^2}} \mathrm{d}\mu \\ &= \frac{1}{\sqrt{2}\sigma^n (2\pi)^{\frac{n}{2}}} e^{\frac{\Sigma x_i^2 + n\delta^2 - \frac{n(\bar{x}+\delta)^2}{2}}{-2\sigma^2}}, \end{aligned} \tag{6}$$

since a normal density has to integrate to 1.

Thus, the posterior density for μ is

$$f(\mu|x_1, x_2, \ldots, x_n) = \frac{f(x_1, x_2, \ldots, x_n; \mu)}{f(x_1, x_2, \ldots, x_n)} = \frac{\sqrt{n}}{\sigma\sqrt{\pi}} e^{\frac{n(\mu - \frac{\bar{x}+\delta}{2})^2}{-\sigma^2}}. \tag{7}$$

Therefore, the posterior distribution of μ is normal with mean $\frac{\bar{x}+\delta}{2}$ and variance $\sigma/\sqrt{2n}$. If the loss function is squared error [4,5], the mean of the posterior normal distribution can be used as the estimation of trust value from ratings. Hence,

Theorem 1. The Bayesian estimation of the trust value of a service with n ratings $x_1, x_2, \ldots, x_n$ ($x_i \in [0,1]$) is

$$T(x_1, x_2, \ldots, x_n, \delta) = \frac{\bar{x}+\delta}{2} = \frac{\Sigma_{i=1}^{n} x_i + n\delta}{2n}, \tag{8}$$

where $\delta \in [0,1]$ denotes the requesting client's prior subjective belief about the trust.

If the requesting client has no prior subjective information about the trust of the requested service, by default, let $\delta = \frac{1}{2}$ since $\frac{1}{2}$ is the middle point of $[0,1]$ representing the neutral belief between distrust and trust. After the Bayesian inference, the Bayesian estimation of the trust can be taken as the requesting client's prior subjective belief about the trust for the Bayesian inference next time.

Now we can estimate the trust of a requested service by combining the requesting client's prior subjective belief about the trust and ratings. Since trust is subjective, it is more reasonable to include the requesting client's prior subjective belief about the trust.

4.2 Global Trust Computation in Composite Services

Our goal is to select the optimal one from multiple *SEF*s (service execution flows) in an *SIG* aiming at maximizing the global trust value of *SEF*, which is determined by the trust values of vertices and invocation relations between vertices in the *SEF*.

According to Definition 5, in *SEF* we only need consider Se (Fig. 1 (a)), Pa (Fig. 1 (b)) and Sy (Fig. 1 (e)) . From Se and Pa, Sy in *SEF* can be determined. Due to space constraint, the details are omitted. Hence, there are two kinds of atomic structures to determine the trust value of an *SEF*: Se and Pa. Se in the *SEF* can be selected from the service invocation relation Se (Fig. 1(a)) or Pr (Fig. 1(c)) in the *SIG*. Pa in the *SEF* can be selected from the service invocation relation Pa (Fig. 1 (b)) in the *SIG*.

Definition 7. The global trust value T_g of an Se structure where service S uniquely invokes service A (see Fig. 1 (a)) can be computed by

$$T_g = T_S \cdot T_A, \tag{9}$$

where T_S and T_A are the trust values of S and A respectively, which are evaluated from Theorem 1. Since S and A are independent, the probability that S and A both occur is equal to the product of the probability that S occurs and the probability that A occurs.

Definition 8. The global trust value T_g of a Pa structure where service S invokes services A and B in parallel (see Fig. 1 (b)) can be computed from T_S and the combined trust value T_{AB} by Definition 7, and

$$T_{AB} = \frac{\omega_1}{\omega_1+\omega_2} \cdot T_A + \frac{\omega_2}{\omega_1+\omega_2} \cdot T_B, \tag{10}$$

where T_S, T_A and T_B are the trust values of S, A and B respectively, which are evaluated from Theorem 1. 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 value by the service trust management authority according to QoS.

According to Definitions 7 & 8, each atomic structure Se or Pa can be converted to a single vertex. Hence, in the process of trust computation, an *SEF* consisting of Se and Pa structures can be incrementally converted to a single vertex with its trust value computed as the global trust. Due to space constraint, we briefly introduce the following global trust computation algorithm. For details, please refer to [12].

Global Trust Computation Algorithm. In order to obtain the global trust value of an *SEF*, firstly the trust value of each atomic Se structure in the *SEF* should be computed by Definition 7. Each computed atomic Se structure is then taken as a vertex in the *SEF*. After that, the trust value of each atomic Pa structure is computed by Definition 8. Similarly, each computed atomic Pa structure is then taken as a vertex in the *SEF*. Thus, the computation can repeat until the final *SEF* is simplified as a vertex, and the global trust value is obtained.

5 Composite Service Selection and Discovery

Here we assume that a service trust management authority stores a large volume of services with their ratings. In response to a client's request, the service trust management authority first generates an *SIG* containing all relevant services and invocation relations. Then, the trust-oriented service selection and discovery algorithm is applied to find the optimal *SEF* with the maximized global trust value.

5.1 Longest *SEF* Algorithm

If there are only Pr (probabilistic invocation) structures in an *SIG* (i.e. there are only Se (sequential invocation) structures in the *SEF*), the *SEF* is a path in the *SIG*. In this case, the longest *SEF* algorithm is applied for searching the optimal *SEF*. By extending Dijkstra's shortest path algorithm [1], the longest *SEF* algorithm is to find an execution flow (path) from *START* to *END* so that the multiplication of trust values of all vertices in the path is the maximal according to Definition 7. Formally, given a weighted graph consisting of set V of vertices and set E of edges, find a flow (path) P from the service invocation root *START* $\in V$ to the service invocation terminal *END* $\in V$ so that

$$\prod_{v_j \in P, v_j \in V} (T(x_1(v_j), x_2(v_j),\ \ldots,\ x_n(v_j), \delta_j)) \tag{11}$$

is the maximal among all flows (paths) from *START* to *END*, where $x_i(v_j)$ denotes a rating for vertex v_j and δ_j denotes the requesting client's prior subjective belief about the trust of vertex v_j. Due to space constraint, we ignore the details of this algorithm.

5.2 Monte Carlo Method Based Algorithm (MCBA)

If there are only Pa structures in an *SIG*, the unique *SEF* is the same as the *SIG*.

If an *SIG* consists of both Prs and Pas, finding the optimal *SEF* is an NP-complete problem [28], and we propose a *Monte Carlo method based algorithm* (*MCBA*) to find the optimal *SEF*.

Algorithm 1. *MCBA* for Composite Service Selection and Discovery

```
Input: Simulation times $l$; SIM, and service ratings Reputation.
Output: The optimal SEF with maximum global trust value $Trust_{global}$.
Let Trust be the trust value for each service evaluated from Reputation by Theorem 1;
for all $i$ such that $1 \leq i \leq l$ do
    Initialize $active = [root]$, SEF$= [root]$;
    while $active \neq \emptyset$ do
        Select a vertex $vertex$ from $active$, and remove $vertex$ from $active$;
        Let vectors $Pr$ and $Pa$ be the Pr and Pa structures from $vertex$;
        if vector $Pa \neq \emptyset$ then
            if $vertex$ is in SEF then
                for all $Pa(j)$ in $Pa$ do
                    if $Pa(j)$ is not in SEF then
                        Add $Pa(j)$ into SEF
                    end if
                end for
            end if
            for all $Pa(j')$ in $Pa(j)$ do
                if $Pa(j')$ is not $terminal$ and $Pa(j')$ is not in $active$ then
                    Add $Pa(j')$ into $active$
                end if
            end for
        end if
        if vector $Pr \neq \emptyset$ then
            if $vertex$ is in SEF then
                if none of $Pr$ is in SEF then
                    for all $Pr(k)$ in $Pr$ do
                        Generate a uniform distributed random number $rand$ in $[0, 1]$;
                        Select the smallest $k'$ such that $rand <Trust(k')/sum(Trust(k))$
                    end for
                    Add $Pr(k')$ in SEF
                end if
            end if
            if $Pr(k')$ is not $terminal$ and $Pr(k')$ is not in $active$ then
                Add $Pr(k')$ into $active$
            end if
        end if
    end while
    Let $Trust_{SEF}$ be the trust value of SEF according to Global Trust Computation Algorithm
    $Trust_{global} = \max Trust_{SEF}$;
end for
return Optimal SEF and $Trust_{global}$.
```

Monte Carlo method [2] is a computational algorithm which relies on repeated random sampling to compute results. It tends to be adopted when it is infeasible to compute an exact result with a deterministic algorithm. Monte Carlo method is useful for modeling phenomena with significant uncertainty in inputs, such as the calculation of risk in business [2]. The specific areas of application of the Monte Carlo method include computational physics, physical chemistry, global illumination computations, finance and business, and computational mathematics (e.g. numerical integration and numerical optimization) [2,17]. It is also one of the techniques for solving NP-complete problems [2,17]. Generally, Monte Carlo method consists of four steps: (1) defining a domain of inputs, (2) generating inputs randomly, (3) performing a computation on each input, and (4) aggregating the results into the final one.

The main strategy in *MCBA* is as follows. In an *SIG*, the direct successors of a service need to be selected according to their trust values. Usually, the direct successor with a larger trust value is preferred, which indicates higher probability to be invoked, and vice versa. Then, according to this, a uniform distributed random number is generated to decide which succeeding service is selected.

When determining the optimal *SEF* from an *SIG*, we only need *MCBA* for Pr structures. Let's take Pr in Fig. 1(c) as an example to explain the details of our *MCBA*. If successor A has a trust value T_A from Theorem 1 and successor B has a trust value T_B from Theorem 1, the probability for vertex S to choose successor A is

$$P_A = \frac{T_A}{T_A + T_B}. \tag{12}$$

Similarly, the probability to choose successor B is

$$P_B = \frac{T_B}{T_A + T_B}. \tag{13}$$

Obviously, $0 < P_A, P_B < 1$. Then a uniform distributed random number r_0 in $(0, 1)$ is generated to decide which successor is chosen. In detail, if $r_0 < P_A$, successor A is chosen; If $P_A < r_0 < P_A + P_B = 1$, successor B is chosen.

Therefore, given an *SIG*, an *SEF* could be obtained by repeating *MCBA* from the service invocation root *START* until the service invocation terminal *END* is reached. Once an *SEF* is generated, its global trust value can be calculated by global trust computation algorithm in Section 4.2. By repeating this process for l simulation times, a set of *SEF*s can be generated, from which the locally optimal *SEF* with the maximal global trust value can be obtained. A high value of l is necessary to obtain the optimal solution. *MCBA* for composite service selection and discovery is illustrated in Algorithm 1.

In Theorem 1, the trust estimation algorithm has a complexity of $O(n)$ with n ratings. Hence, in global trust computation algorithm in Section 4.2, the complexity of trust evaluation for a composite service with N services is $O(nN)$. Therefore, *MCBA* with l simulations incurs a complexity of $O(nlN)$.

6 Experiments

In this section, we will illustrate the results of conducted experiments for studying our proposed *MCBA*.

6.1 Comparison on Travel Plan Composite Services

In this experiment, we compare our proposed *MCBA* with the exhaustive search method by applying them to the travel plan composite services (with 16 vertices and 30 *SEF*s). The corresponding ratings and Smith's prior subjective belief of each service component are listed in Table 2. The weights of service components in all Pa structures of the composite services are listed in Table 3.

The exhaustive search method is inefficient as it aims to enumerate all solutions. In the work by Menascé [16], the exhaustive search method is adopted to calculate execution time and cost of all *SEF*s in a composite service.

According to global trust computation algorithm in Section 4.2, the global trust value T_i of *SEF* i $(i = 1, 2, \ldots, 30)$ can be calculated. Let *trust-based SEF optimality* be

$$O_T(T_i) = \frac{T_i}{\max(T_i)}. \tag{14}$$

Table 2. Ratings and subjective belief of each service component in the travel plan

	Reg	*Acc*	*Air*	*Online*	*Fax*	*Ha*	*Hb*	*Hc*	*Aa*	*Ab*	*Ac*	*Ccard*	*Taxi*	*Bus*
x_1	0.88	0.83	0.78	0.92	0.51	0.17	0.35	0.89	0.30	0.95	0.25	0.95	0.94	0.32
x_2	0.84	0.82	0.87	0.92	0.38	0.18	0.32	0.86	0.36	0.98	0.30	0.95	0.86	0.37
x_3	0.97	0.85	0.77	0.94	0.25	0.22	0.46	0.82	0.34	0.91	0.24	0.96	0.86	0.34
x_4	0.87	0.82	0.83	0.96	0.40	0.12	0.34	0.87	0.29	0.91	0.31	0.96	0.89	0.18
x_5	0.91	0.74	0.79	0.95	0.41	0.16	0.28	0.88	0.41	0.97	0.29	0.96	0.90	0.35
δ	0.92	0.85	0.91	0.95	0.32	0.20	0.50	0.91	0.32	0.92	0.51	0.98	0.89	0.33

Table 3. Weights of service components in Pa

Reg	*Acc*	*Air*	*Ccard*	*Taxi*	*Ccard*	*Bus*
0.1	0.3	0.6	0.6	0.4	0.6	0.4

The corresponding histograph of $O_T(T_i)$ values of 30 *SEF*s is plotted in Fig. 5. From it, we can observe that 80% of $O_T(T_i)$ values are less than 0.8, implying that if we choose an *SEF* randomly, it is very likely to obtain an *SEF* with a low trust value .

In *MCBA*, there are multiple simulations, in each of which an *SEF* is generated and its global trust value is calculated. After l simulations, a locally optimal *SEF* can be obtained from l generated *SEF*s. In order to study the distribution of global trust of locally optimal *SEF*s, we take l simulations as a repetition and repeat for m times.

Our experiments are using Matlab 7.6.0.324 (R2008a) running on a Dell Vostro V1310 laptop with an Intel Core 2 Duo T5870 2.00GHz CPU and a 3GB RAM. l, the number of simulation times, is set from 1 to 100. m, the number of repetition times, is set from 1 to 100. The experiment results are plotted in Fig. 7. We could observe that with a fixed number of repetitions, the more simulations, the closer to 1 O_T becomes. Namely more simulations lead to a higher probability to obtain the optimal *SEF*.

Furthermore, we compare the execution time of *MCBA* with that of the exhaustive search method. Each CPU time in this paper is the average of ten independent

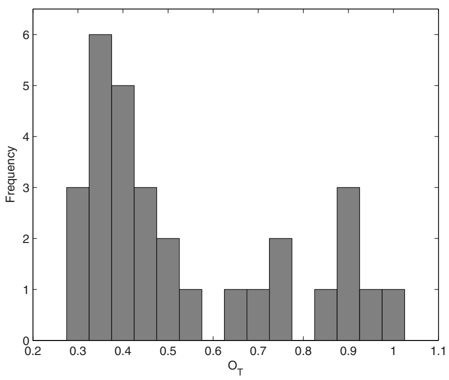


Fig. 5. Histograph of O_T for each *SEF*

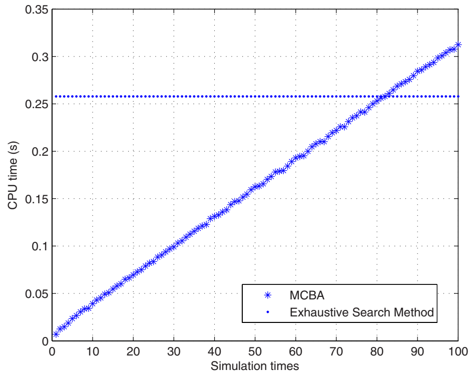


Fig. 6. CPU time of simulation times

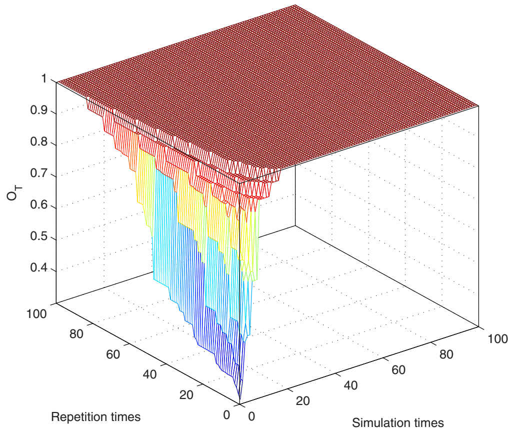


Fig. 7. O_T in the travel plan example

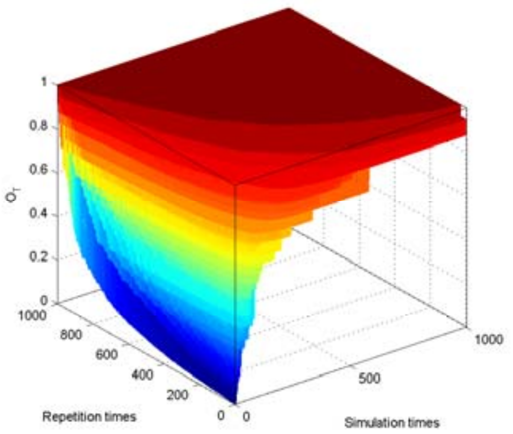


Fig. 8. O_T in composite service of 100 vertices

executions. In Fig. 6, we can observe that when the number of simulation times $l \leq 82$, our *MCBA* is faster than the exhaustive search method. From Figs 6 and 7, we can see that the probability to obtain the optimal *SEF* is 97% when there are 20 simulations. Meanwhile, the execution time of our *MCBA* is 27% of the one of the exhaustive search method. According to Table 2, theoretically the probability to obtain the optimal *SEF* for each simulation in *MCBA* is 17.8%, due to *SIG* and the strategy in *MCBA* in Section 5.2. Hence after 20 simulations theoretically *MCBA* has the probability of 98.04% to obtain the optimal *SEF*. Hence the experiment result about the probability to obtain the optimal *SEF* confirms to the theoretical conclusion.

With this simple travel plan example, *MCBA* outperforms the exhaustive search method. More significant performance differences can be observed with some complex composite services to be introduced in the next section.

6.2 Comparison on Complex Composite Services

In this experiment, we further compare our proposed Monte Carlo method based algorithm (*MCBA*) and the exhaustive search method on three more complex composite services. The number of vertices of these composite services is 35, 52 and 100 respectively. The numbers of Ses, Pas, Prs, Sys, Ass and *SEF*s in corresponding composite services are listed in Table 4.

Table 4. Structure of complex composite services

Number of vertices	Ses	Pas	Prs	Sys	Ass	*SEF*s
35	17	8	11	4	11	1.8×10^3
52	24	13	16	7	16	5.4×10^4
100	51	24	32	12	32	2.92×10^9

Table 5. CPU time in seconds of different examples

Number of vertices	16	35	52	100
Probability to obtain the optimal *SEF* for each simulation	17.84%	14.31%	5.71%	0.33%
Number of simulation times in *MCBA*	20	20	52	925
Probability to obtain the optimal *SEF* for *MCBA*	98.04%	95.45%	95.29%	95.12%
CPU time (seconds) of *MCBA*	0.0695	0.3219	0.8625	34.51
CPU time (seconds) of exhaustive search method	0.2578	17.09	–	–

In this experiment, we use the same platform as the experiment in Section 6.1. In the case of composite service with 35 vertices, the *MCBA* takes 0.3219 second to finish 20 simulations with the probability of 95.45% to obtain the optimal *SEF*, while the exhaustive search method uses 17.09 seconds. When the number of vertices becomes 52, our *MCBA* takes 0.8625 second to finish 52 simulations, with which the probability to obtain the optimal *SEF* is 95.29%. However, when taking the same time, the exhaustive search method can only search 0.42% of 5.4×10^4 *SEF*s. When taking 1000 times of the *MCBA* CPU time, it can only search approximately 1% of all *SEF*s. We further apply our *MCBA* to a composite service with 100 vertices. It takes 34.51 seconds to finish 925 simulations with a probability of 95.12% to obtain the optimal *SEF*. In contrast, when taking the same time, the exhaustive search method can only search $(9.56 \times 10^{-6})\%$ of 2.92×10^9 *SEF*s. When taking 100 times of the *MCBA* CPU time, it can only search $(1.01 \times 10^{-5})\%$ of all *SEF*s. The above results are listed in Table 5.

In the case of composite service with 100 vertices, the results of *MCBA* are plotted in Fig. 8. When there are $l = 925$ simulation times, *MCBA* can reach the optimal solution with the probability 95.2% . Also it has a great chance to obtain the near-optimal one, even when l is as small as 200. For example, in Fig. 8, when l is 200, the probability for the trust-based *SEF* optimality to be $O_T \geq 0.82$ is about 95.7%.

In summary, our proposed *MCBA* can obtain a near-optimal *SEF* after some simulations. As the CPU time for a single simulation in *MCBA* is extremely short, our experiments have illustrated that the overall performance of *MCBA* is good even with complex composite services. In addition, *MCBA* is suitable for parallel computing since each simulation in *MCBA* is independent. This can greatly speed up computations and shorten the overall CPU time. Thus, our proposed *MCBA* is realistic and efficient.

7 Conclusions

In this paper, we first propose our service invocation graph and service invocation matrix for composite service representation. In addition, a novel trust evaluation approach based on Bayesian inference has been proposed that can aggregate the ratings from other clients and the requesting client's prior subjective belief about the trust. Based on them, a Monte Carlo method based trust-oriented service selection and discovery algorithm has been proposed. Experiments have illustrated that our proposed approach can discover the near-optimal composite services efficiently.

In our future work, strategies for optimizing the Monte Carlo method based algorithm will be studied to further improve the efficiency. We will also study some heuristic approaches for trust-oriented optimal service selection and discovery.

References

1. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik 1, 269–271 (1959)
2. Gentle, J., Härdle, W., Mori, Y.: Handbook of Computational Statistics. Springer, Heidelberg (2004)
3. Haddad, J.E., Manouvrier, M., Ramirez, G., Rukoz, M.: QoS-driven selection of web services for transactional composition. In: ICWS 2008, pp. 653–660 (2008)
4. Hamada, M.S., Wilson, A.G., Reese, C.S., Martz, H.F.: Bayesian Reliability. Springer, Heidelberg (2008)
5. Hines, W.W., Montgomery, D.C., Goldsman, D.M., Borror, C.M.: Probability and Statistics in Engineering. John Wiley & Sons, Inc., Chichester (2003)
6. Hu, N., Pavlou, P.A., Zhang, J.: Can online reviews reveal a product's true quality?: empirical findings and analytical modeling of online word-of-mouth communication. In: ACM EC 2006, pp. 324–330 (2006)
7. Jøsang, A.: Subjective evidential reasoning. In: IPMU (2002)
8. Jøsang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. Decision Support Systems 43(2), 618–644 (2007)
9. Jurca, R., Faltings, B.: Collusion-resistant, incentive-compatible feedback payments. In: ACM EC 2007, pp. 200–209 (2007)
10. Jurca, R., Faltings, B.: Minimum payments that reward honest reputation feedback. In: ACM EC 2006, pp. 190–199 (2006)
11. Knight, D.H., Chervany, N.L.: The meaning of trust. Technical Report WP9604, University of Minnesota, Management Information Systems Research Center (1996)
12. Li, L., Wang, Y.: Trust evaluation in composite services selection and discovery. In: SCC 2009, Bangalore, India, September 21-25 (2009)
13. Malik, Z., Bouguettaya, A.: RATEWeb: Reputation assessment for trust establishment among web services. VLDB J. 18(4), 885–911 (2009)
14. Medjahed, B., Bouguettaya, A.: A multilevel composability model for semantic web services. IEEE Trans. Knowl. Data Eng. 17(7), 954–968 (2005)
15. Medjahed, B., Bouguettaya, A., Elmagarmid, A.K.: Composing web services on the semantic web. VLDB J. 12(4), 333–351 (2003)
16. Menascé, D.A.: Composing web services: A QoS view. IEEE Internet Computing 8(6), 88–90 (2004)
17. Morton, D.P., Popova, E.: Monte-Carlo simulations for stochastic optimization. In: Encyclopedia of Optimization, pp. 2337–2345. Springer, Heidelberg (2009)
18. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: Service-oriented computing: a research roadmap. Int. J. Cooperative Inf. Syst. 17(2), 223–255 (2008)
19. Vu, L.-H., Hauswirth, M., Aberer, K.: QoS-based service selection and ranking with trust and reputation management. In: Meersman, R., Tari, Z. (eds.) OTM 2005. LNCS, vol. 3760, pp. 466–483. Springer, Heidelberg (2005)
20. Wang, Y., Lim, E.-P.: The evaluation of situational transaction trust in e-service environments. In: ICEBE 2008, pp. 265–272 (2008)
21. Wang, Y., Lin, K.-J., Wong, D.S., Varadharajan, V.: Trust management towards service-oriented applications. Service Oriented Computing and Applications journal 3(1) (2009)

22. Wang, Y., Singh, M.P.: Formal trust model for multiagent systems. In: Proceedings 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), pp. 1551–1556 (2007)
23. Wang, Y., Wong, D.S., Lin, K.-J., Varadharajan, V.: Evaluating transaction trust and risk levels in peer-to-peer e-commerce environments. Inf. Syst. E-Business Management 6(1), 25–48 (2008)
24. Xiao, J., Boutaba, R.: QoS-aware service composition and adaptation in autonomic communication. IEEE Journal on Selected Areas in Communications 23(12), 2344–2360 (2005)
25. Xiong, L., Liu, L.: PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities. IEEE Trans. Knowl. Data Eng. 16(7), 843–857 (2004)
26. Xu, Z., Martin, P., Powley, W., Zulkernine, F.: Reputation-enhanced QoS-based web services discovery. In: ICWS 2007, pp. 249–256 (2007)
27. Yu, B., Singh, M.P., Sycara, K.: Developing trust in large-scale peer-to-peer systems. In: 2004 IEEE First Symposium on Multi-Agent Security and Survivability, pp. 1–10 (2004)
28. Yu, T., Zhang, Y., Lin, K.-J.: Efficient algorithms for web services selection with end-to-end Qos constraints. TWEB 1(1) (2007)
29. Zacharia, G., Maes, P.: Trust management through reputation mechanisms. Applied Artificial Intelligence 14(9), 881–907 (2000)
30. Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J., Sheng, Q.Z.: Quality driven web services composition. In: WWW 2003, pp. 411–421 (2003)