World Wide Web (2016) 19:807–832
DOI 10.1007/s11280-015-0362-0

Trust-based group services selection in web-based service-oriented environments

Xing Su[1] · Minjie Zhang[1] · Yi Mu[1]

Received: 20 November 2013 / Revised: 15 June 2015 /
Accepted: 17 June 2015 / Published online: 21 July 2015

Abstract Multi-agent technologies have been widely employed for the development of web-based systems, including web-based e-markets, web-based grid computing, e-governments and service-oriented software systems. In these service-oriented systems, service provider agents and service consumer agents are autonomous and rational entities, which can enter and leave the environments freely. For simplicity, we use terms 'provider' and 'consumer' to represent this two types of agents. How to select the most suitable service providers according to a service request from a consumer in such an open environment is a very challenging issue. In this paper, we propose an innovative trust model, called the GTrust model, for service group selection in general service-oriented environments. In the GTrust model, the trust evaluation for a service group is based on the functionality coverage of the group, the dependency relationships among individual services in the group, the ratings of individual services on the attributes of the service request and a similarity measurement of the extent to which reference reports can reflect the service request in terms of the priority distribution of attributes. The experimental results and analysis demonstrate the good performance of the GTrust model on the service group selection in service-oriented environments.

Keywords Intelligent agents · Trust model · Service-oriented environments

✉ Xing Su
xs702@uowmail.edu.au

Minjie Zhang
minjie@uow.edu.au

Yi Mu
ymu@uow.edu.au

[1] School of Computing and Information Technology, University of Wollongong,
Keiraville, NSW Australia

1 Introduction

In the last decade, agent and Multi-Agent Systems (MASs) technologies have been widely applied to develop web-based service-oriented systems such as Internet-based grid systems [4, 27], e-markets [2, 9, 11], pervasive computing systems [13, 15], and e-government [5, 24]. Web-based service-oriented environments are open environments where consumers and providers can enter and leave the environments freely. In such environments, how to select a trustworthy and suitable service provider to fulfill a service request for a consumer is a very challenging issue for most service-oriented applications.

'Agent trust' is one of the important research issues in MASs. There are a number of models, which have been developed to evaluate trusts in MASs, for example, the probabilistic theory-based model [1, 21], the certified reputation model [7] and the evidential trust model [23]. In the last decade, some trust models have been developed in service-oriented environments to help consumers evaluate the trust values of potential service providers from different perspectives [3, 6, 8, 10, 16, 25]. Zacharia et al. proposed a reputation-based trust evaluation model, called the SPORAS [26], to select single service provider based on their historical performance. Huynh et al. introduced a famous trust model, called the 'Certified Reputation' (CR) model, to evaluate a service provider's trust through third party references [7]. Su et al. extended the work of the CR model and developed a priority-based trust model to evaluate a full context trust value of a potential service provider based on third party references, the historical performance and the priority distribution of attributes [19]. Most current trust models focus on evaluating the trust value for a single provider; however, many complex service requests from consumers cannot be handled by an individual service from a single service provider and a service group consisting of a number of services from different providers are needed to satisfy these service requests [20, 22]. In such situations, trust models focusing on the trust evaluation for single service providers cannot be directly employed for the trust evaluation for service groups. How to select a suitable service group based on a complex service request has become a new problem.

The trust evaluation for a service group is different from that of a single service, because there are some specific factors that affect the trust value of a service group. Such factors mainly include: (1) the functionality coverage of the service group on the service request, which determines whether the group can satisfy all the attributes of the service request, (2) the workflow and dependency relationships among individual services of the group, (3) the historical performance of individual services in the group, and (4) the suitability about whether the historical performance of the group can reflect its performance on the service request based on the priority distribution of attributes.

In the literature, there are only a few models for group trust evaluation, but they do not take most of aforementioned factors into account. The REGRET model [12] is one of these models, which generated trust value of a service group from averaging all trust values of individual services in the group.

In this paper, we introduce an innovative trust model, the GTrust model, for trust evaluation of a service group by considering aforementioned four factors. The GTrust model has the following merits. (1) We use the 'functionality coverage' to measure the extent of functionalities of a service request satisfied by a service group. (2) We introduce the 'workflow description' and the 'degree of dependency' to describe the workflow and dependency relationships among individual services of a service group. (3) We borrow the concept of 'third party reference report' from the PBTrust model [19] to record the historical performance of individual services in a service group. (4) We use the concept of 'similarity' to measure the similarity of priority distribution between the service request and historical services in reference reports.

The rest of this paper is organised as follows. In Section 2, the problem description and definitions are presented. The basic components of the GTrust model are introduced briefly in Section 3. The modules of the GTrust model are introduced in detail in Section 4. In Section 5, we demonstrate experimental results and analysis. The related work and comparison are given in Section 6. In Section 7, we conclude the paper and outline the future work

2 Problem description and definitions

In general, a service in service-oriented environment can be described by a number of attributes such as price, time, quality, and their corresponding priority values. For different service requests, the priority distribution on the same attributes are different. In order to precisely describe a service request, we propose a service description formally.

Suppose that a service request includes n number of attributes and each attribute has a priority value to describe its importance in the request. A service can be described by n number of attributes and their priority values as follows.

Definition 1 A *service description* (*SDes*) is defined as a 2 × n matrix.

$$SDes = \begin{pmatrix} A_1 & A_2 & A_3 & ... & A_n \\ P_1 & P_2 & P_3 & ... & P_n \end{pmatrix}, \tag{1}$$

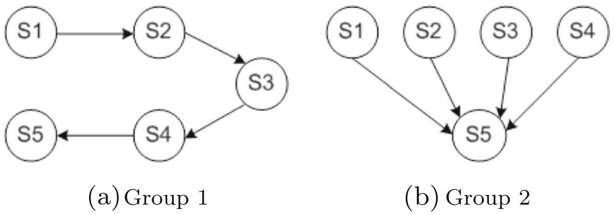
where A_j is the j^{th} attribute of the service request, P_j is the priority value of A_j and $\sum_{j=1}^{n} P_j = 1$.

Definition 2 A *reference report* (*Rf*) is defined as a two-tuple, $Rf =< SDes, Ratings >$, where $SDes$ is the service description of the service request from a historical consumer (third party) and $Ratings$ is further defined as a vector, $Ratings =< R_1, R_2, ..., R_n >$, where R_j represents the rating of the individual service on the j^{th} attribute of the historical service request and $R_j \in [0, 1]$, where 0 and 1 represent the worst and best performance of the individual service, respectively. The expansion of a reference report is described as follows.

$$Rf = \left\langle \begin{pmatrix} A_1 & A_2 & A_3 & ... & A_n \\ P_1 & P_2 & P_3 & ... & P_n \end{pmatrix}, < R_1, R_2, R_3, ..., R_n > \right\rangle \tag{2}$$

To deal with a complex service request, a number of individual services need to form a group with a certain workflow and dependency relationships among them. If two service groups have the same individual services but the workflows and dependency relationships among these individual services are different, the two service groups may have a different performance on a service request. For example, suppose that two service groups have the same individual services S_1, S_2, S_3, S_4 and S_5, but different workflows and dependency relationships among individual services, which are described as follows.

In Figure 1, Group 1 has a sequential workflow from S_1 to S_5, i.e., the latter individual services can begin to work when their former individual services finish their works. The workflow in Group 2 are different from the workflow in Group 1. In Group 2, S_1, S_2,



(a) Group 1 (b) Group 2

Figure 1 The workflows and dependency relationships among individual services of the two service groups

S_3 and S_4 can work concurrently and S_5 can only begin to work when they finish their works. In addition, the dependency relationships among individual services can affect the performance of the service group. For example, in Group 1, if S_4 has a bad performance and offers a low quality output to S_5, even if S_5 has a very good performance, the bad input received from S4 can reduce the actual performance of S_5.

In order to describe the extent of the dependency relationship between two individual services, we introduce the concept of degree of dependency.

Definition 3 A *degree of dependency* (λ) is defined as a value $\lambda \in [0, 1]$, where 0 and 1 represents no dependency relationship and the strongest dependency relationship between two individual services, respectively.

The workflow and dependency relationships among individual services of a service group can be described by a directed graph, which can be further described by an adjacency matrix as follows.

Definition 4 A *workflow description* (*WDes*) of a service group which consists of m number of individual services can be represented by an $m \times m$ adjacency matrix as (3).

$$WDes = \begin{pmatrix} \lambda_{11}, \lambda_{12}, \lambda_{13}, ..., \lambda_{1m} \\ \lambda_{21}, \lambda_{22}, \lambda_{23}, ..., \lambda_{2m} \\ ..., ..., ..., ..., ... \\ \lambda_{m1}, \lambda_{m2}, \lambda_{m3}, ..., \lambda_{mm} \end{pmatrix}, \tag{3}$$

where $\lambda_{ii'} \in [0, 1]$ represents the degree of dependency that the i'^{th} service (i.e., $S_{i'}$) depends on the i^{th} service (i.e., S_i). $\lambda_{ii'} = 0$ represents that there is no dependency relationship between S_i and $S_{i'}$. $0 < \lambda_{ii'} \leq 1$ represents that $S_{i'}$ depends on S_i and the value of $\lambda_{ii'}$ is the degree of dependency between S_i and $S_{i'}$.

In addition, cycles in a workflow (i.e., there are two services S_i and $S_{i'}$ in $WDes$, where S_i directly or indirectly depends on $S_{i'}$ and $S_{i'}$ directly or indirectly depends on $S_{i'}$) can cause infinite calculation loops in the final rating calculation and the individual rating calculation (i.e., see, Sections 4.3.3 and 4.4). Hence, the GTrust model can only calculate the trust value for a group of services with the acyclic workflow.

3 Basic modules of the GTrust model

The GTrust model consists of 4 modules which are the request module, the reply module, the priority-based group trust calculation module and the evaluation module. The working procedure of the 4 modules is shown as follows.

As shown in Figure 2. When a consumer needs a complex service, (1) the request module of the consumer generates a service request and broadcasts it to providers in the network; (2) potential providers form service groups and the reply module of each service group generates a service reply to the consumer; (3) the priority-based group trust calculation module of the consumer calculates the trust value for each replied service group and selects the service group with the highest trust value; (4) after the selected service group finishing the service request, the evaluation module of the consumer generates reference reports for individual services of the group to describe their performance on the service request, which are recorded by the individual services.

4 The principle of the GTrust model

In this section, the 4 modules of the GTrust model are introduced in detail.

4.1 The request module

The objective of the request module is to generate a service request (i.e., $SReq$) based on consumer's requirements. A service request is described as a two-tuple $SReq =< SDes, RN >$, where $SDes$ is the service description and RN is the number of reference reports required from each individual service. In the GTrust model, the value of RN has the significant influence on the trust evaluation of service groups. If the value of RN is high, the consumer will receive more reference reports from each service group, which can help the consumer evaluate the potential performance of each service group accurately. However, the high value of RN can also significantly reduce the number of service groups that can reply the service request. Many suitable service groups cannot reply the service request, since some individual services in these service groups do not have enough reference reports

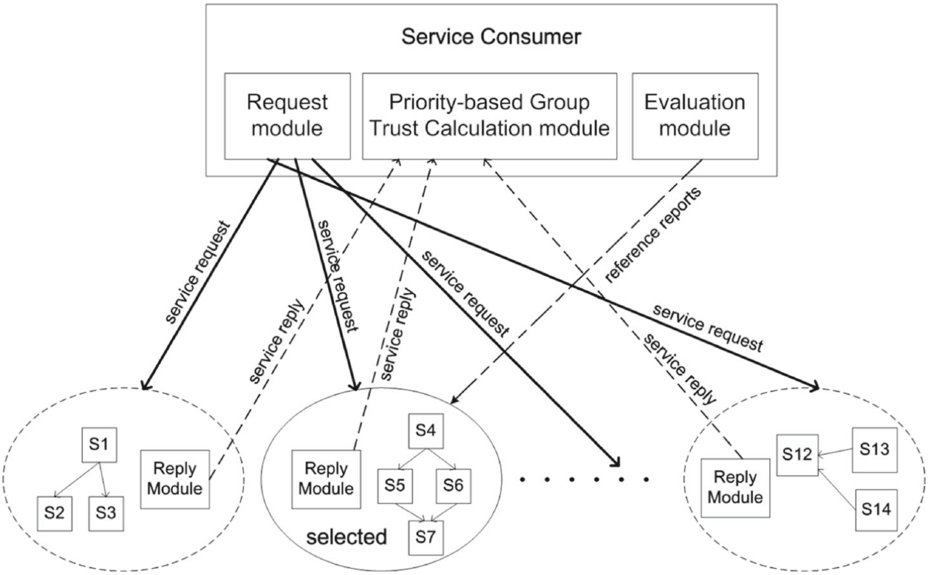


Figure 2 The working procedure of the GTrust model

to achieve the requirement of the consumer. To the contrary, if the value of RN is low, more service groups can reply the service request. However, the reference reports received from each service group might not be enough for the consumer to evaluate the potential performance of each service group accurately. In service-oriented networks, the suitable value of RN is related to many factors, such as the number of service providers in the network, the numbers of reference reports obtained by service providers, the functionality coverage of these reference reports on the current service request, the numbers of individual services in service groups and the workflows and dependency relationships among individual services of service groups. Therefore, the value of RN should be decided according to the circumstances of networks. We leave this for users to decide the value of RN based on the above factors and their applications.

Suppose that consumer C in an e-market environment needs a complex service including 5 attributes (i.e., cost, speed, quality, colour and warranty). The priority values of 5 attributes are 0.1, 0.4, 0.2, 0.1 and 0.2, respectively. In addition, C requires each individual service to provide 3 reference reports to indicate its historical performance. Based on the above requirements, the request module generates a service request $SReq$ as follows.

$$SReq = \left\langle \begin{pmatrix} Cost & Speed & Quality & Colour & Warranty \\ 0.1 & 0.4 & 0.2 & 0.1 & 0.2 \end{pmatrix}, 3 \right\rangle \tag{4}$$

Then, $SReq$ is broadcasted to all providers through the network. The above example will be used to explain the following modules in this section.

4.2 The reply module

The purpose of the reply module is to generate a service reply to describe the workflow, dependency relationships and historical performance of individual services in a service group. A service reply is described as a two-tuple $SR =< WDes, RfSet >$, where $WDes$ is the workflow and dependency relationships among individual services of the service group and $RfSet$ is the set of reference reports of individual services in the service group. For example, if a service group (SG) intends to conduct the service request, the reply module of SG collects the following information: the workflows and dependency relationships among individual services of SG and reference reports of individual services of SG.

Suppose that SG consists of 5 individual services named S_1, S_2, S_3, S_4 and S_5, with the workflow and dependency relationships shown as follows.

In Figure 3, the circles represent the individual services of SG, the arrows represent the dependency relationships between two individual services and the value on each arrow

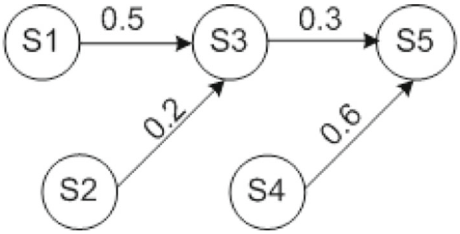


Figure 3 The workflow and dependency relationships among individual services of SG

indicates the degree of dependency (i.e., λ, see Definition 3) between two individual services.

The workflow description (i.e., $WDes$, see Definition 4) of SG can be represented in the format of Definition 4 as follows.

$$WDes = \begin{pmatrix} 0, 0, 0.5, 0, 0 \\ 0, 0, 0.2, 0, 0 \\ 0, 0, 0, 0, 0.3 \\ 0, 0, 0, 0, 0.6 \\ 0, 0, 0, 0, 0 \end{pmatrix} \tag{5}$$

In addition to the workflow description, the service provider of each individual service in SG also needs to provide RN number of reference reports (i.e., Rf, see Definition 2) to reflect its historical performance on similar service requests. The service providers are assumed to be rational entities and are familiar with the trust evaluation of the GTrust model so that they will provide their most suitable reference reports to the consumer that can maximise the trust value of SG. The suitability ($RSuit$) of a reference report on the service request is calculated based on two aspects: 1) the similarity of priority distribution in service descriptions between service request (i.e., $SDes_1$ in $SReq$, see Section 4.1) and the reference report (i.e., $SDes_2$ in Rf, see Definition 2) and 2) the ratings of the reference report (e.g. $Ratings$ in Rf, see Definition 2) on the service request, which can be calculated as follows.

$$RSuit = RSim \times RRating, \tag{6}$$

where $RSim$ is the similarity of priority distribution in service descriptions between the service request and the reference report; and $RRating$ is the ratings of the reference report on the service request.

To calculate the similarity ($RSim$) of priority distribution in service descriptions (i.e., $SDes_1$ and $SDes_2$), the order of attributes in $SDes_2$ is arranged as same as that of in $SDes_1$., and then, $RSim$ is calculated based on the distance of two vectors, which is described as follows.

$$RSim = e^{-\sqrt{\sum_{j=1}^{n}(P_{1j}-P_{2j})^2}}, \tag{7}$$

where $RSim$ is the similarity of priority distributions between $SDes_1$ and $SDes_2$ and $RSim \in [0, 1]$, where 0 and 1 represent the lowest and highest similarity of priority distributions, respectively; P_{1j} and P_{2j} are the priority values in $SDes_1$ and $SDes_2$ on the j^{th} attribute of the service request, respectively. In addition, some attributes of the service request might not be covered by the reference report (i.e., $P_{2j} = \varnothing$). In this situation, the priority values of these uncovered attributes equal 0 (i.e., $P_{2j} = 0$) in this calculation.

To calculate the ratings ($RRating$) of the reference report on the service request, we let each rating in the reference report multiply its corresponding priority value in the service description of the service request (i.e., $SDes_1$) first. Then, we add up all the calculation results together, which is described as follows.

$$RRating = \sum_{j=1}^{n} P_{1j} \cdot R_j, \tag{8}$$

where $RRating$ is the ratings of the reference report on the service request; P_{1j} is the priority value of the j^{th} attribute of the service request; and R_j is the rating of the reference report on the j^{th} attribute of the service request. In addition, some attributes in the service request might not be covered by the reference report (i.e., $R_j = \varnothing$). In this situation, the ratings of these uncovered attributes equal 0 (i.e., $R_j = 0$) in this calculation.

Finally, the RN number of reference reports with the highest suitability values ($RSuit$) in each individual service are chosen to be the suitable reference reports. After collected all necessary information, the reply module will create a service reply $SR =< WDes, RfSet >$ for SG, where $WDes$ is the workflow description of SG and $RfSet$ consists of b number of reference reports and $b = m \times RN = 15$, where $m = 5$ is the number of individual services in SG and $RN = 3$ is the number of reference reports required from each individual service.

4.3 The priority-based group trust calculation module

The purpose of this module is to evaluate the trust value of each service group based on its service reply SR. This module produces the trust value for a service group based on three factors: 1) the functionality coverage of a service group on the attributes of the service request, 2) the similarity of priority distribution in service descriptions between the service request and the reference reports of the group and 3) the ratings of the group on the attributes of the service request. Since a service group is composed of different individual services owned by different providers, the group abilities to handle the service request depends on the abilities of individual services in the group. We use a group service description to formally describe the abilities of a service group by extracting service descriptions from reference reports of the group.

Definition 5 A *group service description* (*GSDes*) is represented by a $b \times n$ matrix, where b is the number of reference reports of a service group and n is the number of attributes in the service request. $GSDes$ is described as follows.

$$GSDes = \begin{pmatrix} P_{11} & P_{12} & \dots & P_{1n} \\ P_{21} & P_{22} & \dots & P_{2n} \\ \dots & \dots & \dots & \dots \\ P_{b1} & P_{b2} & \dots & P_{bn} \end{pmatrix} \tag{9}$$

In (9), the r^{th} row represents the priority distribution of the r^{th} reference report and P_{rj} represents the priority value of the r^{th} reference report on the j^{th} attribute of the service request. If the reference report does not cover the j^{th} attribute of the service request, $P_{rj} = \varnothing$; otherwise $P_{rj} \in [0, 1]$, where 0 and 1 represent the lowest and highest priority values, respectively. By using (9), the abilities of a service group can be described comprehensively.

4.3.1 Functionality coverage calculation

The purpose of functionality coverage calculation is to measure whether the abilities provided by a service group can cover all the attributes of the service request. The functionality coverage of a service group on the service request is defined as follows.

Definition 6 A *functionality coverage* (*FCov*) is defined as an n-tuple $FCov =< ACov_1$, $ACov_2$, $ACov_3$,...$ACov_n >$, where $ACov_j \in [0, 1]$ represents the functionality coverage value of the service group on the j^{th} attribute of the service request, which can be calculated based on the group service description (i.e., $GSDes$, see Definition 5) as follows.

$$ACov_j = \frac{b - MS_j}{b}, \tag{10}$$

where $ACov_j$ represents the functionality coverage value of a service group on the j^{th} attribute of the service request, b represents the number of reference reports of the group and MS_j represents the number of '$\varnothing$' (i.e., how many reference reports cannot cover the j^{th} attribute of the service request) in the j^{th} column of $GSDes$. If the functionality coverage value on the j^{th} attribute equals 0, we can say that this service group is not able to conduct the service request.

4.3.2 Group similarity calculation

The objective of the group similarity calculation is to measure the similarity of priority distributions between the service request and a service group. In the GTrust model, the priority distribution of the service request is a vector, which can be easily obtained through extracting priority values P_i from $SDes$ (see Definition 1). However, the priority distribution of a service group is a matrix (i.e., $GSDes$, see Definition 4), which is obtained through extracting priority values P_i from b number of reference reports of the group. In order to evaluate the similarity of priority distribution between the service request and a service group, we must transfer the priority distribution of a service group (i.e., the matrix $GSDes$) to a vector $GPV =< GP_1, GP_2, GP_3, ...GP_n >$, where GP_j is the priority value of the service group on the j^{th} attribute of the service request. GP_j in GPV is calculated as follows.

$$GP_j = \frac{\sum_{r=1}^{b} P_{rj}}{b}, \tag{11}$$

where b is the number of reference reports of the service group and P_{rj} is the priority value of the r^{th} reference report on the j^{th} attribute of the service request. In addition, some attributes of the service request might not be covered by some reference reports (i.e., $P_{rj} = \varnothing$). In this situation, the priority values of these attributes equal 0 (i.e., $P_{rj} = 0$) in this calculation.

With (11), we can calculate each element in vector GPV and transfer the priority distribution of the service group to a vector. After that, we can calculate the similarity ($GSim$) of priority distribution between the service request and the service group based on the distance between two vectors (i.e. see (8)), which is described as follows.

$$GSim = e^{-\sqrt{\sum_{j=1}^{n}(P_j - GP_j)^2}}, \tag{12}$$

where $GSim$ is a value in [0, 1], where 0 and 1 represent the lowest and highest similarities of priority distributions, respectively; P_j is the priority value of the j^{th} attribute of the service request; and GP_j is the priority value of the service group on the j^{th} attribute of the service request.

4.3.3 Group rating calculation

The purpose of group rating calculation is to predict the performance of a service group on the attributes of the service request. The rating of the service group on the j^{th} attribute of the service request is calculated as follows.

$$GRating_j = \frac{\sum_{i=1}^{m} FRating_{ij}}{m}, \tag{13}$$

where m is the number of individual services in the service group and $FRating_{ij}$ is the final rating of the i^{th} individual service on the j^{th} attribute of the service request, after considering the degree of dependency with other individual services in the group. $FRating_{ij}$ is calculated as follows.

$$FRating_{ij} = \frac{\sum_r R_{rj}}{RN} \times \left(1 - \frac{\sum_{k=1}^{d} \lambda_{ki} \cdot (1 - FRating_{kj})}{d}\right), \tag{14}$$

where $\frac{\sum_i R_{rj}}{RN}$ is the average rating of the i^{th} individual service on the j^{th} attribute of the service request, which is calculated from the RN number of reference reports provided by the i^{th} individual service; R_{rj} is the rating value of the r^{th} reference report on the j^{th} attribute of the service request; d represents the number of individual services on which the i^{th} individual service (i.e., S_i) depends and λ_{ki} is the degree of dependency that S_i depends on its k^{th} depending individual service and $FRating_{kj}$ is the final rating of the k^{th} depending individual service of S_i on the j^{th} attribute of the service request. In addition, since cycles in a workflow (i.e., see, Definition 4) can cause infinite calculation loops in the final rating calculation, the GTrust model can only calculate final ratings of a group of services with the acyclic workflow.

The final rating calculation begins from the individual service without depending on any other individual service and then move to the individual services depending on the individual services whose final ratings already known. In addition, if some ratings in reference reports on an attribute do not exist, the average value of existing ratings on the same attribute is used to represent the missing ratings during the group rating calculation. The reason for this setting is that these rating must be in [0, 1], where 0 and 1 represent the worst and best historical performance of individual services on corresponding attributes, respectively. First of all, it is not reasonable to set these missing ratings to 1. In addition, we do not set these missing ratings to 0 for two reasons. First, for the missing ratings, the trust value of a service group has been reduced in the functionality coverage calculation (i.e., $ACov_j$, see Section 4.3.1). If we set the values of the missing ratings to 0, it means that the trust value of the service group has been reduced for the missing ratings again in the group rating calculation, which would increase the influence of the missing ratings on the trust value of the service group. Secondly, the missing ratings in the reference reports do not mean that individual services have the worst performance on the corresponding attributes. Therefore, the average value of existing ratings on the same attributes means that the missing ratings do not have any influence on the group rating calculation of the service group, since we have considered them in the functionality coverage calculation (i.e., see Section 4.3.1).

4.3.4 Final trust calculation

After the functionality coverage calculation, the group similarity calculation and the group rating calculation, we can calculate the final trust value *Trust* of a service group as follows.

$$Trust = GSim \cdot \sum_{j=1}^{n} P_j \cdot ACov_j \cdot GRating_j, \tag{15}$$

where $GSim$ is the group similarity value, P_j is the priority value of the j^{th} attribute in the service request, $ACov_j$ is the functionality coverage of the service group on the j^{th} attribute of the service request and $GRating_j$ represents the rating of the service group on the j^{th}

attribute of the service request after considering the workflow and dependency relationships among individual services of the group.

4.4 The evaluation module

After completing the service request, the consumer uses the evaluation module to generate the group reference report based on the performance of the group on the service request in the format of Definition 2, which is described as follows.

$$Rf =< SDes, < GR_1, GR_2, GR_3, ..., GR_n >>, \tag{16}$$

where $SDes$ is the service description of the service request, $GR_j \in [0, 1]$ represents the group rating on the j^{th} attribute of the service request.

The group ratings represent the performance of the service group on the final product of the service request. Due to the dependency relationships among individual services, some individual services use the outputs of other individual services as their inputs so that the performance of these individual services might be reduced by the low quality of inputs from other individual services. After the group reference report generation, the consumer generates an individual reference report for each individual service in the group, which trends to recover the actual performance of the individual service. The individual ratings are adapted by taking the group ratings, the workflow and dependency relationships among individual services into account, which are calculated as follows.

$$R_{ij} = GR_j \times \left(1 + \frac{\sum_{k=1}^{d} \lambda_{ki} \cdot (1 - R_{kj})}{d}\right), \tag{17}$$

where $R_{ij} \in [0, 1]$ is the individual rating of the i^{th} individual service on the j^{th} attribute of the service request, GR_j is the group rating on the j^{th} attribute of the service request, d represents the number of the individual services on which the i^{th} individual service (i.e., S_i) depends and λ_{ki} is the degree of dependency that S_i depends on its k^{th} depending individual service and R_{kj} is the individual rating of the k^{th} depending individual service of S_i on the j^{th} attribute of the service request. In addition, since cycles in a workflow (i.e., see, Definition 4) can cause infinite calculation loops in the individual rating calculation, the GTrust model can only calculate individual ratings of a group of services with the acyclic workflow.

We use the same example in the request module and the reply module to demonstrate how to generate the group and individual reference reports in the evaluation module. After completing the service request, Consumer C generates the group reference report based on the performance of the service group SG on attributes of the service request, which is described as follows.

$$\left\langle \begin{pmatrix} Cost & Speed & Quality & Colour & Warranty \\ 0.1 & 0.4 & 0.2 & 0.1 & 0.2 \end{pmatrix}, < 0.6, 0.4, 0.9, 0.9, 0.3 > \right\rangle \tag{18}$$

From the consumer's group ratings, we can see that the consumer was satisfied with the 'cost' attribute of the service provided by SG (i.e., the first attribute of $SDes$), was not satisfied with the 'speed' and 'warranty' attributes of the service (i.e., the second and fifth attributes of $SDes$), and was very satisfied with the 'quality' and 'colour' attributes of the service (i.e., the third and forth attributes of $SDes$).

Based on the group reference report, the individual reference reports can be generated as follows. First, from Figure 3, it can be seen that S_1, S_2 and S_4 in SG do not have other individual services to depend on (i.e., $d = 0$, see (17)) so that the individual ratings of S_1, S_2 and S_4 are the same as the group ratings in (18). Then, since S_3 depends on S_1 and S_2, the individual ratings of S_3 are calculated based on (17), which are described as follows.

$$\left\langle \begin{pmatrix} Cost & Speed & Quality & Colour & Warranty \\ 0.1 & 0.4 & 0.2 & 0.1 & 0.2 \end{pmatrix}, < 0.68, 0.48, 0.93, 0.93, 0.37 > \right\rangle \tag{19}$$

Since S_5 depends on S_3 and S_4, the individual ratings of S_5 are calculated based on (17), which are described as follows.

$$\left\langle \begin{pmatrix} Cost & Speed & Quality & Colour & Warranty \\ 0.1 & 0.4 & 0.2 & 0.1 & 0.2 \end{pmatrix}, < 0.70, 0.50, 0.94, 0.94, 0.39 > \right\rangle \tag{20}$$

5 Experiments and analysis

Two experiments are conducted to evaluate the performance of the GTrust model on the service group selection. After examining the related research, we found the REGRET model [12] to compare with the GTrust model. In the REGRET model, the average trust value of all individual services in a service group is used as the trust value for the group, which does not consider the workflow and dependency relationships among individual services of the group.

5.1 Experimental settings of experiment 1

In Experiment 1, 1 consumer and 20 individual services have participated. The 20 individual services form 4 service groups (i.e., G_1 to G_4) and each service group contains 5 individual services.

The consumer sends a service request containing 5 attributes (i.e., *cost*, *speed*, *quality*, *colour* and *warranty*) with different priority values. The service description of the service request is described as follows.

$$SDes' = \begin{pmatrix} Cost & Speed & Quality & Colour & Warranty \\ 0.04 & 0.08 & 0.2 & 0.34 & 0.34 \end{pmatrix}. \tag{21}$$

In order to comprehensively test the trust calculation of the GTurst model and the REGRET model, the reference reports of individual services in 4 service groups are generated to distinguish the advantages of different service groups, where G_1 has the highest functionality coverage on the attributes of the service request, G_3 has the highest similarity with the priority distribution of the service request, G_4 has the highest ratings on the attributes of the service request and G2 has the second highest functionality coverage, similarity, and ratings among 4 service groups. Each individual service is required to provide 3 reference reports (i.e., $RN = 3$) so that a service group needs to provide 15 reference reports to the consumer. The reference reports of 20 individual services are shown in Table 1.

Table 1 The references reports of the twenty individual services

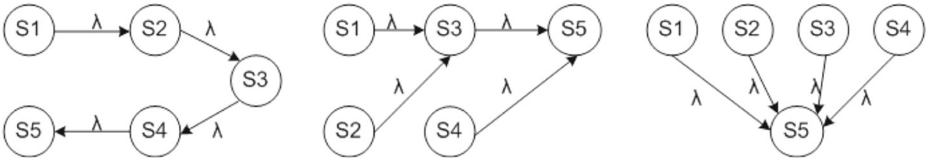
Report ID	Group ID	Provider ID	Priority distribution	Ratings
1	G_1	P_1	$(0.5, 0.2, 0.2, 0.1, 0)$	$< 0.2, 0.2, 0.2, 0.2, 0.2 >$
2	G_1	P_1	$(0.7, 0.1, 0.1, 0.1, \varnothing)$	$< 0.2, 0.2, 0.2, 0.2, 0 >$
3	G_1	P_1	$(0.9, 0, 0, 0.1, 0)$	$< 0.2, 0.2, 0.2, 0.2, 0.2 >$
4	G_1	P_2	$(0.6, 0, 0.2, 0, 0.2)$	$< 0.2, 0.2, 0.2, 0.2, 0.2 >$
5	G_1	P_2	$(0.4, \varnothing, 0.1, 0.1, 0.4)$	$< 0.2, 0, 0.2, 0.2, 0.2 >$
6	G_1	P_2	$(0.8, \varnothing, 0, 0.2, 0)$	$< 0.2, 0, 0.2, 0.2, 0.2 >$
7	G_1	P_3	$(\varnothing, 0.4, 0, 0.4, 0.2)$	$< 0, 0.2, 0.2, 0.2, 0.2 >$
8	G_1	P_3	$(\varnothing, 0.6, 0, 0.2, 0.2)$	$< 0, 0.2, 0.2, 0.2, 0.2 >$
9	G_1	P_3	$(\varnothing, 0.8, \varnothing, 0, 0.2)$	$< 0, 0.2, 0, 0.2, 0.2 >$
10	G_1	P_4	$(0.9, 0.1, \varnothing, 0, \varnothing)$	$< 0.2, 0.2, 0, 0.2, 0 >$
11	G_1	P_4	$(0.8, 0.2, 0, \varnothing, 0)$	$< 0.2, 0.2, 0.2, 0, 0.2 >$
12	G_1	P_4	$(0.7, \varnothing, \varnothing, 0.3, 0)$	$< 0.2, 0, 0, 0.2, 0.2 >$
13	G_1	P_5	$(0.5, 0.2, 0.3, \varnothing, \varnothing)$	$< 0.2, 0.2, 0.2, 0, 0 >$
14	G_1	P_5	$(0.7, 0.1, 0.2, 0, 0)$	$< 0.2, 0.2, 0.2, 0.2, 0.2 >$
15	G_1	P_5	$(0.6, 0.3, 0.1, \varnothing, 0)$	$< 0.2, 0.2, 0.2, 0, 0.2 >$
16	G_2	P_6	$(0.2, \varnothing, 0.5, 0.3, \varnothing)$	$< 0.6, 0, 0.6, 0.6, 0 >$
17	G_2	P_6	$(0.1, \varnothing, 0.6, 0.3, \varnothing)$	$< 0.6, 0, 0.6, 0.6, 0 >$
18	G_2	P_6	$(0.3, 0, 0.4, 0.3, \varnothing)$	$< 0.6, 0.6, 0.6, 0.6, 0 >$
19	G_2	P_7	$(\varnothing, 0.2, \varnothing, 0.8, \varnothing)$	$< 0, 0.6, 0, 0.6, 0 >$
20	G_2	P_7	$(0.1, 0.3, \varnothing, 0.6, \varnothing)$	$< 0.6, 0.6, 0, 0.6, 0 >$
21	G_2	P_7	$(0.2, 0.1, \varnothing, 0.7, \varnothing)$	$< 0.6, 0.6, 0, 0.6, 0 >$
22	G_2	P_8	$(0, 0.3, 0.2, \varnothing, 0.5)$	$< 0.6, 0.6, 0.6, 0, 0.6 >$
23	G_2	P_8	$(\varnothing, \varnothing, 0.4, \varnothing, 0.6)$	$< 0, 0, 0.6, 0, 0.6 >$
24	G_2	P_8	$(\varnothing, 0.3, 0, 0, 0.7)$	$< 0, 0.6, 0.6, 0.6, 0.6 >$
25	G_2	P_9	$(\varnothing, 0.2, \varnothing, 0.5, 0.3)$	$< 0, 0.6, 0, 0.6, 0.6 >$
26	G_2	P_9	$(\varnothing, 0.1, 0, \varnothing, 0.9)$	$< 0, 0.6, 0.6, 0, 0.6 >$
27	G_2	P_9	$(\varnothing, \varnothing, \varnothing, 0.4, 0.6)$	$< 0, 0, 0, 0.6, 0.6 >$
28	G_2	P_{10}	$(0.4, \varnothing, 0.2, \varnothing, 0.4)$	$< 0.6, 0, 0.6, 0, 0.6 >$
29	G_2	P_{10}	$(0.2, 0, \varnothing, \varnothing, 0.8)$	$< 0.6, 0.6, 0, 0, 0.6 >$
30	G_2	P_{10}	$(0, \varnothing, 0.4, \varnothing, 0.6)$	$< 0.6, 0, 0.6, 0, 0.6 >$
31	G_3	P_{11}	$(\varnothing, \varnothing, \varnothing, 1, \varnothing)$	$< 0, 0, 0, 0.4, 0 >$
32	G_3	P_{11}	$(\varnothing, \varnothing, \varnothing, 1, \varnothing)$	$< 0, 0, 0, 0.4, 0 >$
33	G_3	P_{11}	$(0.3, \varnothing, \varnothing, 0.7, \varnothing)$	$< 0.4, 0, 0, 0.4, 0 >$
34	G_3	P_{12}	$(\varnothing, \varnothing, \varnothing, \varnothing, 1)$	$< 0, 0, 0, 0, 0.4 >$
35	G_3	P_{12}	$(0, \varnothing, 0.2, \varnothing, 0.8)$	$< 0.4, 0, 0.4, 0, 0.4 >$
36	G_3	P_{12}	$(\varnothing, \varnothing, 0.1, \varnothing, 0.9)$	$< 0, 0, 0.4, 0, 0.4 >$
37	G_3	P_{13}	$(\varnothing, 0, 0, \varnothing, 1)$	$< 0, 0.4, 0.4, 0, 0.4 >$
38	G_3	P_{13}	$(0, 0.4, \varnothing, \varnothing, 0.6)$	$< 0.4, 0.4, 0, 0, 0.4 >$
39	G_3	P_{13}	$(\varnothing, 0, \varnothing, \varnothing, 1)$	$< 0, 0.4, 0, 0, 0.4 >$
40	G_3	P_{14}	$(\varnothing, 0.3, \varnothing, 0.7, \varnothing)$	$< 0, 0.4, 0, 0.4, 0 >$
41	G_3	P_{14}	$(\varnothing, 0, \varnothing, 1, \varnothing)$	$< 0, 0.4, 0, 0.4, 0 >$
42	G_3	P_{14}	$(0, 0.3, \varnothing, 0.7, \varnothing)$	$< 0.4, 0.4, 0, 0.4, 0 >$
43	G_3	P_{15}	$(0.3, \varnothing, 0.7, \varnothing, \varnothing)$	$< 0.4, 0, 0.4, 0, 0 >$

Table 1 (continued)

Report ID	Group ID	Provider ID	Priority distribution	Ratings
44	G_3	P_{15}	$(\varnothing, \varnothing, 1, \varnothing, \varnothing)$	$< 0, 0, 0.4, 0, 0 >$
45	G_3	P_{15}	$(0, \varnothing, 1, \varnothing, \varnothing)$	$< 0.4, 0, 0.4, 0, 0 >$
46	G_4	P_{16}	$(1, \varnothing, \varnothing, \varnothing, \varnothing)$	$< 0.8, 0, 0, 0, 0 >$
47	G_4	P_{16}	$(\varnothing, \varnothing, 1, \varnothing, \varnothing)$	$< 0, 0, 0.8, 0, 0 >$
48	G_4	P_{16}	$(1, \varnothing, \varnothing, \varnothing, \varnothing)$	$< 0.8, 0, 0, 0, 0 >$
49	G_4	P_{17}	$(\varnothing, \varnothing, \varnothing, \varnothing, 1)$	$< 0, 0, 0, 0, 0.8 >$
50	G_4	P_{17}	$(\varnothing, \varnothing, 1, \varnothing, \varnothing)$	$< 0, 0, 0.8, 0, 0 >$
51	G_4	P_{17}	$(\varnothing, \varnothing, \varnothing, 1, \varnothing)$	$< 0, 0, 0, 0.8, 0 >$
52	G_4	P_{18}	$(\varnothing, \varnothing, 1, \varnothing, \varnothing)$	$< 0, 0, 0.8, 0, 0 >$
53	G_4	P_{18}	$(\varnothing, \varnothing, \varnothing, 1, \varnothing)$	$< 0, 0, 0, 0.8, 0 >$
54	G_4	P_{18}	$(1, \varnothing, \varnothing, \varnothing, \varnothing)$	$< 0.8, 0, 0, 0, 0 >$
55	G_4	P_{19}	$(\varnothing, \varnothing, \varnothing, \varnothing, 1)$	$< 0, 0, 0, 0, 0.8 >$
56	G_4	P_{19}	$(1, \varnothing, \varnothing, \varnothing, \varnothing)$	$< 0.8, 0, 0, 0, 0 >$
57	G_4	P_{19}	$(\varnothing, \varnothing, \varnothing, 1, \varnothing)$	$< 0, 0, 0, 0.8, 0 >$
58	G_4	P_{20}	$(1, \varnothing, \varnothing, \varnothing, \varnothing)$	$< 0.8, 0, 0, 0, 0 >$
59	G_4	P_{20}	$(\varnothing, \varnothing, \varnothing, \varnothing, 1)$	$< 0, 0, 0, 0, 0.8 >$
60	G_4	P_{20}	$(1, \varnothing, \varnothing, \varnothing, \varnothing)$	$< 0.8, 0, 0, 0, 0 >$

4 service groups are tested in three scenarios with different workflows and dependency relationships among individual services, which are shown in Figure 4.

In Figure 4, the workflows in three scenarios represent three common workflows of 5 individual services in real-life applications, respectively. The workflow of Scenario 1 represents the sequential workflow, in which each individual service depends on the output of its former individual service. The workflow in Scenario 2 represents the hierarchical workflow, in which two individual services are at the low dependency level, two individual services are at the middle dependency level and one individual service is at the high dependency level. The workflow in Scenario 3 represents the concurrent workflow, in which one individual service depends on the other 4 individual services. In each scenario, we calculate the trust values for 4 service groups based on 5 different degrees of dependency (i.e., 0, 0.1, 0.5, 0.7 and 1). The reason for this setting is to comprehensively test influence of the degree of dependency on the trust value of the service group.



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3

Figure 4 The workflows and dependency relationships among individual services in three scenarios

5.1.1 Trust value transfer function

Since the GTrust model and the REGRET model use different presentations of trust values, we define the trust value transfer function to transfer trust values of the GTrust model to trust values of the REGRET model. The trust value transfer function is described as follows.

$$Trans(Rf) = \frac{\sum_{j=1}^{n} R_j}{n - \sum_{j=1}^{n} MS_j}, \tag{22}$$

where Rf is a reference report of the GTrust model (see Definition 2), R_j is the rating of the reference report on the j^{th} attribute of the service request, n is the number of attributes in the service request and MS_j is the number of missing attributes of the reference report on the j^{th} attribute of the service request.

5.2 Experimental results and analysis of experiment 1

This subsection gives the experimental results and analysis of Experiment 1 in detail.

5.2.1 Experimental results and analysis in scenario 1

Table 2 shows the results of the functionality coverage, group similarities and ratings of 4 service groups in the GTrust model with 5 degrees of dependency in Scenario 1 (Figure 4a). Since the results of a service group on 5 attributes of the service request are the same, we use only one value to represent them all. Tables 4 and 6 are the same as Table 2.

Table 3 shows the trust values of 4 service groups calculated by the GTrust model and the REGRET model in 5 degrees of dependency of Scenario 1. The trust values with (C) indicate the service groups selected by the GTrust model and the REGRET model, respectively.

Figure 5 demonstrates the trust values of G_2 and G_4 evaluated by the GTrust model and the REGRET models in 5 degrees of dependency in Scenario 1. In Figure 5, the X-axis represents the degrees of dependency, while the Y-axis represents the trust values of service groups.

From Table 3, we can see that all of the service groups except G_2 have their advantages in regard to the service request. G_1 has the highest functionality coverage on the attributes of the service request, G_3 has the highest similarity of priority distribution with the service request, and G_4 has the highest ratings on the service request. In principle, the REGRET model always selects the service group based on the average rating of reference reports.

Table 2 The functionality coverage, similarities and ratings of 4 service groups

	G_1	G_2	G_3	G_4
$ACov$	0.8	0.6	0.4	0.2
$GSim$	0.559	0.956	1.000	0.731
$Ratings$				
$\lambda = 0$	0.2	0.6	0.4	0.8
$\lambda = 0.1$	0.187	0.580	0.380	0.786
$\lambda = 0.5$	0.131	0.477	0.287	0.711
$\lambda = 0.7$	0.100	0.409	0.231	0.655
$\lambda = 1$	0.050	0.277	0.132	0.538

Table 3 The trust values of 4 service groups

	GTrust				REGRET			
	G_1	G_2	G_3	G_4	G_1	G_2	G_3	G_4
$\lambda = 0$	0.0894	0.3442 (C)	0.1600	0.1170	0.2	0.6	0.4	0.8(C)
$\lambda = 0.1$	0.0836	0.3326 (C)	0.1521	0.1150	0.2	0.6	0.4	0.8(C)
$\lambda = 0.5$	0.0585	0.2739 (C)	0.1150	0.1039	0.2	0.6	0.4	0.8(C)
$\lambda = 0.7$	0.0447	0.2346 (C)	0.0925	0.0957	0.2	0.6	0.4	0.8(C)
$\lambda = 1$	0.0224	0.1587 (C)	0.0528	0.0786	0.2	0.6	0.4	0.8(C)

Therefore, in Scenario 1, the REGRET model always selects G_4, since G_4 has the highest average rating among 4 service groups. The GTrust model considers 4 main factors: the functionality coverage, dependency relationships, the similarity and ratings, which can affect the performance of a service group from different perspectives. After the trust value calculation, the GTrust model selects G_2 for the service request because G_2 has the second highest value on each main factor but the best trust value by considering 4 main factors at the same time.

Now, we analyse the selection results from two models to see which service group is more suitable for the service request from 4 perspectives:

(1) Workflow and dependency relationships. From Figure 4a, we can see that all of 4 service groups have a sequential workflow in this scenario, i.e., the latter individual service depends on its former individual service. In other words, the performance of the former individual service affect the performance of the latter individual service. From Figure 5, we can clearly see that the degree of dependency has an impact on group trust value calculation in the GTrust model. The experimental result of the GTrust model in Scenario 1 shows that when the degree of dependency (λ) among individual services in a group increases, the group trust values decreases. For example, we can see from Table 3 that when the degree of dependency increases from 0 to

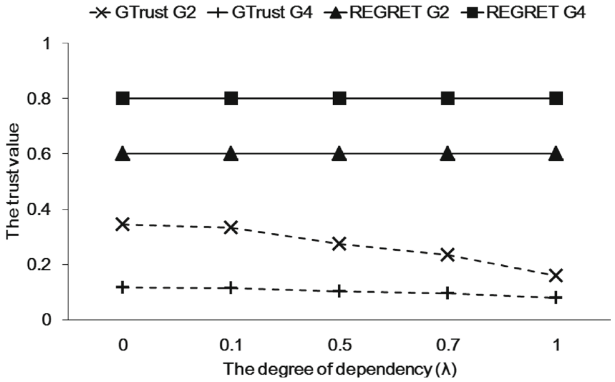


Figure 5 The trust values of service groups selected by the two trust models

1, the trust value of G_2 decreases from 0.34 to 0.16 and the trust value of G_4 decreases from 0.12 to 0.08. Figure 5 also indicates that the trust values of G_2 and G_4 calculated by the REGRET model always remain constant i.e., 0.6 and 0.8, respectively, in all 5 degrees of dependency. In the real-life applications, a service group always has the workflow and dependency relationships among individual services. With this consideration, the trust calculation process of the GTrust model is closer to realistic situations than that of the REGRET model.

(2) Similarity of priority distributions. The reference reports can partially reflect the potential performance of individual services on the service request. The more similar between the priority distribution of the service request and reference reports of a service group, the better the reference reports reflect the potential performance of the service group on the service request. From the priority distribution of 5 attributes of the service request (see Subsection 5.1), we can see that if a service group wants to provide similar reference reports with the service request, the reference reports must have the highest priority values on the 'colour' and 'warranty' attributes, the second highest priority value on the 'quality' attribute, then the 'speed' and the 'cost' attributes. Table 1 shows the priority distributions in reference reports provided by individual services of 4 service groups. In G_4, only reference reports 51, 53 and 57 have the highest priority values on the 'colour' attribute and reference reports 49, 55 and 59 have the highest priority values on the 'warranty' attribute. Different with G_4, in G_2, reference reports 19, 20, 21 and 25 have the highest priority values on the 'colour' attribute and reference reports 22, 23, 24, 26, 27, 29 and 30 have the highest priority values on the 'warranty' attribute. After comparing the priority distribution of reference reports of G_2 and G_4, we can see that reference reports of G_2 can better reflect the potential performance of the service group on the service request than that of G_4.

(3) Functionality coverage. The functionality coverage value can reflect whether a service group can satisfy the functionalities of the service request. In other words, whether a service group can cover all attributes of the service request. Table 1 also shows the functionality coverage of individual services in 4 service groups. In G_4, the reference reports (reference reports 46 to 60) miss too many priority values and ratings on the attributes of the service request and the functionality coverage of G_4 on the service request is only 20 %. In G_2, even if reference reports (reference reports 16 to 30) also miss some priority values and ratings on the attributes of the service request, the functionality coverage of G_2 on the service request is 60 %, which is much higher than the functionality coverage of G_4. Therefore, the potential performance of individual services in G_2 is more predictable than that of G_4 on the service request.

(4) Final decision making. The REGRET model selects G_4 based on its average rating (0.8). The GTrust model selects G_2 not only based on its average rating (0.6), but also based on its workflow, the similarity of priority distributions as well as its functionality coverage on the service request.

Therefore, the service group selected by the GTrust model is more reasonable than that of selected by the REGRET model in Scenario 1.

5.2.2 Experimental result and analysis in scenario 2

Table 4 shows the results of the functionality coverage, group similarities and ratings of 4 service groups in the GTrust model with 5 degrees of dependency in Scenario 2 (Figure 4b).

Table 4 The functionality coverage, similarities and ratings of 4 service groups

	G_1	G_2	G_3	G_4
$ACov$	0.8	0.6	0.4	0.2
$GSim$	0.559	0.956	1.000	0.731
$Ratings$				
$\lambda = 0$	0.2	0.6	0.4	0.8
$\lambda = 0.1$	0.194	0.590	0.390	0.793
$\lambda = 0.5$	0.167	0.548	0.35	0.765
$\lambda = 0.7$	0.154	0.526	0.328	0.749
$\lambda = 1$	0.133	0.490	0.294	0.723

Table 5 shows the trust values of 4 service groups calculated by the GTrust model and the REGRET model in 5 degrees of dependency of Scenario 2. The trust values with (C) indicate the service groups selected by the GTrust model and the REGRET model, respectively.

Figure 6 demonstrates the trust values of G_2 and G_4 evaluated by the GTrust model and the REGRET models in 5 degrees of dependency in Scenario 2. In Figure 6, the X-axis represents the degree of dependency, while the Y-axis represents the trust values of service groups.

In this scenario, the GTrust model also selects G_2 while the REGRET model also selects G_4. Using the same analysis as in Scenario 1, we can see that the service group selected by the GTrust model is more reasonable than that of selected by the REGRET model in Scenario 2.

5.2.3 *Experimental result and analysis in scenario 3*

Table 6 shows the results of the functionality coverage, group similarities and ratings of 4 service groups in the GTrust model with 5 degrees of dependency in Scenario 3 (Figure 4c).

Table 7 shows the trust values of 4 service groups calculated by the GTrust model and the REGRET model in 5 degrees of dependency of Scenario 3. The trust values with (C) indicate the service groups selected by the GTrust model and the REGRET model, respectively.

Figure 7 demonstrates the trust values of G_2 and G_4 evaluated by the GTrust model and the REGRET models in 5 degrees of dependency in Scenario 3. In Figure 7, the X-axis represents the degree of dependency, while the Y-axis represents the trust values of service groups.

Table 5 The trust values of 4 service groups

	GTrust				REGRET			
	G_1	G_2	G_3	G_4	G_1	G_2	G_3	G_4
$\lambda = 0$	0.0894	0.3442 (C)	0.1600	0.1170	0.2	0.6	0.4	0.8(C)
$\lambda = 0.1$	0.0866	0.3386 (C)	0.1561	0.1160	0.2	0.6	0.4	0.8(C)
$\lambda = 0.5$	0.0748	0.3146 (C)	0.1398	0.1118	0.2	0.6	0.4	0.8(C)
$\lambda = 0.7$	0.0687	0.3016 (C)	0.1312	0.1095	0.2	0.6	0.4	0.8(C)
$\lambda = 1$	0.0594	0.2808 (C)	0.1178	0.1057	0.2	0.6	0.4	0.8(C)

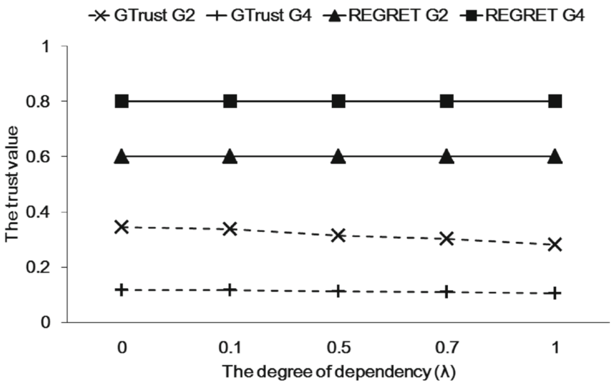


Figure 6 The trust values of service groups selected by two trust models

In this scenario, the GTrust model still selects G_2, while the REGRET model still selects G_4. Using the same analysis as in Scenario 1, we can see that the service group selected by the GTrust model is more reasonable than that of selected by the REGRET model in Scenario 3.

5.2.4 The impact of workflows and dependency relationships on the service group selection

In the experiment, we used three scenarios with three different workflows and 5 different dependency relationships. Now, we analyse the potential impact of different workflows and degrees of dependency on the group trust values. Figure 8 demonstrates the changes of trust values of G_2 and G_4 evaluated by the GTrust model and the REGRET model with 5 degrees of dependency in three scenarios, where the X-axis represents the degree of dependency and the Y-axis represents the trust values of service groups.

From Figure 8, we can clearly see that although the degree of dependency changes in three scenarios, the trust values of G_4 evaluated by the REGRET model remain a constant, while the trust values of G_2 evaluated by the GTrust model change based on different scenarios. In Scenario 1, the 5 individual services of G_2 are in a sequential workflow (see Figure 4a) and there are 4 dependency relationships among individual services. The trust values of a service group in Scenario 1 highly depends on the 4 dependency relationships so that from Figure 8, we can see that the trust values of G_2 decreases significantly as the

Table 6 The functionality coverage, similarities and ratings of 4 service groups

	G_1	G_2	G_3	G_4
$ACov$	0.8	0.6	0.4	0.2
$GSim$	0.559	0.956	1.000	0.731
$Ratings$				
$\lambda = 0$	0.2	0.6	0.4	0.8
$\lambda = 0.1$	0.197	0.595	0.395	0.797
$\lambda = 0.5$	0.184	0.576	0.376	0.784
$\lambda = 0.7$	0.178	0.566	0.366	0.778
$\lambda = 1$	0.168	0.552	0.352	0.768

Table 7 The trust values of 4 service groups

	GTrust				REGRET			
	G_1	G_2	G_3	G_4	G_1	G_2	G_3	G_4
$\lambda = 0$	0.0894	0.3442 (C)	0.2294	0.1600	0.2	0.6	0.4	0.8(C)
$\lambda = 0.1$	0.0880	0.3414 (C)	0.2267	0.1594	0.2	0.6	0.4	0.8(C)
$\lambda = 0.5$	0.0823	0.3304 (C)	0.2157	0.1568	0.2	0.6	0.4	0.8(C)
$\lambda = 0.7$	0.0794	0.3249 (C)	0.2102	0.1555	0.2	0.6	0.4	0.8(C)
$\lambda = 1$	0.0751	0.3166 (C)	0.2019	0.1536	0.2	0.6	0.4	0.8(C)

degree of dependency (λ) increases. In Scenario 2, there are also 4 dependency relationships among individual services in G_2. However, the dependency relationships among individual service are different from that of in Scenario 1. For example, S_3 depends on S_1 and S_2. This means either S_1 or S_2 has a partial influence on S_3. The same relationships exist between S_5 with S_3 and S_4. Scenario 2 has weaker dependency relationships among individual services so that when the degree of dependency (λ) increases, the trust value of G_2 decreases but the decrease rate is slower than that of in Scenario 1. Scenario 3 has the slowest decrease rate on trust values when the degree of dependency increases. This is because each individual service (S_1 to S_4) only has a partial influence on S_5. From the above analysis the trust value calculated based on the GTrust model is more reasonable than that of the REGRET model, because we take the workflow and dependency relationships of a service group into account.

In summary, the trust value of a service group is impacted not only by the ratings in reference reports, but also by the workflow and dependency relationships among individual services of the group.

5.3 Experimental settings of experiment 2

In Experiment 2, 1 consumer and 50 individual services have participated. The 50 individual services form 10 service groups and each service group contains 5 individual services. The service in Experiment 2 also contains 5 attributes (i.e., *cost*, *speed*, *quality*, *colour* and *warranty*).

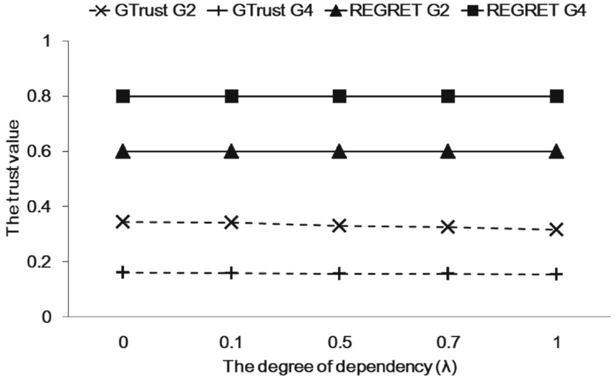


Figure 7 The trust values of service groups selected by two trust models

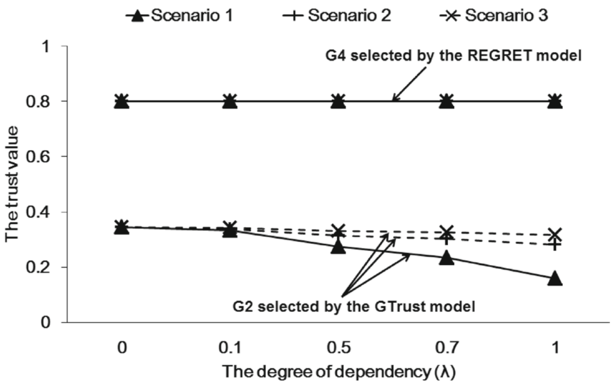


Figure 8 The trust values of service groups evaluated by two trust models in three scenarios

In the experiment, the consumer totally sends 100 service requests, each of which only contains 3 attributes, which are randomly selected from the 5 attributes of the service. The priority values of the 3 attributes are randomly created from values in [0, 1] with one decimal place. In addition, the sum of priority values of the 3 attributes is 1. In the experiment, each service request only requires 1 reference report from each individual service (i.e., $RN = 1$). In order to remove the impact of the workflow and dependency relationships on the service group selection, the workflows and degrees of dependency of individual services in 10 service groups (see, Definition 4) are the same as Figure 4a with 0.5 degree of dependency (i.e., $\lambda = 0.5$).

At the beginning of the experiment, each individual service only has 1 reference report, which also contains 3 random attributes. The creation of priority values of the 3 attributes is the same as those created in service requests. The ratings of 3 random attributes in a reference report are randomly created from values in [0, 1] with one decimal place. For each service request, the consumer selects service groups based on the GTrust model and the REGRET model, respectively. In the REGRET model, the trust values of service groups are calculated from reference reports based on (22). After the service groups completed the service request, the consumer employs the evaluation module of the GTrust model to create new reference reports based on the performance of the service groups, which are recorded by the individual services of the service groups and used to reply the following service requests.

The functionality coverage, group similarities and ratings of the service groups selected by two models are three indicators in this experiment. In addition, the number of reference reports in each service group after 100 service requests is also calculated to know the number of times that each service group is selected by two models in the experiment.

5.3.1 The performance estimation function

In this experiment, how to estimate the performance of the selected service groups on the service request is a problem, which has influenced the reference reports created by the consumer and the following service group selections. In order to estimate the performance, we first assume that all ratings in reference reports are evaluated by the consumer objectively, which means that ratings in reference reports are only related to the performance of individual services. Based on this assumption, we can directly estimate the performance of an individual service on the attributes of the service request from the ratings of its reference report. In addition, if an attribute is not covered by a reference report of an individual service, the performance of the individual service on the attribute is estimated as the average value of existing ratings on the same attribute in other reference reports. The reason for this estimation is to eliminate the influence of the missing ratings on the performance of the service group, which is the same as the setting in the group rating calculation (i.e., see, Section 4.3.3). Based on the above assumption and setting, the performance estimation function is proposed as follows.

$$AP_j = \frac{\sum_{r=1}^{b} R_{rj}}{b}, \tag{23}$$

where AP_j is the estimated performance of a service group on the j^{th} attribute of the service request, R_{rj} is the rating of the r^{th} reference report on the j^{th} attribute of the service request, and b is the number of reference reports of the service group.

5.4 Experimental results and analysis of Experiment 2

Figure 9 shows the average values of the functionality coverage, group similarities and ratings of the service groups selected by the GTrust model and the REGRET model, respectively. In Figure 9, the X-axis represents the three indicators of the service groups, while the Y-axis represents the average values of the three indicators of the service groups.

From Figure 9, it can be seen that the functionality coverage and group similarities of the service groups selected by the GTrust model are more than those selected by the REGRET model. This is because both of the functionality coverage and group similarities are two factors in the trust calculation of the GTrust model, while these factors are not taken into account by the REGRET model during the trust calculation. Therefore, the service groups selected by the GTrust model have higher values on the functionality coverage and group similarities. However, the ratings of the service groups selected by the GTrust model are sightly lower than those selected by the REGRET model. This is because that during the trust calculation, the GTrust model considers not only the ratings of service groups, but also the functionality coverage and group similarities of the service group so that the service groups selected by the GTrust model do not always have the highest values on ratings, but must have the highest values on all of the functionality coverage, group similarities

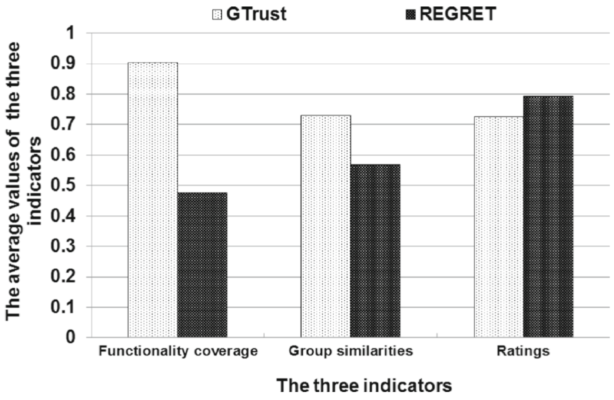


Figure 9 The average values of the three indicators of the service groups selected by two models

and ratings among all service groups. Differing from the GTrust model, the service groups selected by the REGRET model always have the highest value on ratings among all service groups, because the GTrust model only considers the ratings of service groups during the trust calculation. The GTrust model considers three factors rather than one factor during the trust calculation so that the service groups selected by the GTrust model are more reasonable than those selected by the REGRET model.

Figure 10 shows the number of reference reports of each service groups after 100 service requests. In Figure 10, the X-axis represents the ten service groups, while the Y-axis represents the numbers of reference reports in ten service groups after the experiment.

From Figure 10, it can be seen that based on the GTrust model, the number of reference reports of all service groups are more than 1, which means that all of them have been selected for different service requests. Because the GTrust model considers all of the functionality coverage, group similarities and ratings factors during the trust calculation, it can find the service group with the highest values on three factors based on the attributes and priority distribution of the service request. Therefore, the GTrust model could select different service groups for 100 different service requests. However, based on the REGRET model, only G_1 and G_2 have the number of reference reports more than 1, which means that only G_1 and G_2 were selected for 100 different service requests by the REGRET model, in which, G_1 (i.e., having 90 reference reports) was selected for about 90 % service requests. This is because the REGRET model only considers ratings of the service groups during the trust calculation. Even if the priority distributions of 100 service requests are different, the ratings in the reference reports provided by different service groups are the same. Therefore, G_1 and G_2 having the highest ratings among all service groups were always selected by the REGRET model. The GTrust model selects the suitable service group based on the attributes and priority distribution of the service request so that the service groups selected by the GTrust model are more reasonable than those selected by the REGRET model.

6 Related work and comparison

First, J. Sabater and C. Sierra proposed the REGRET trust model in 2001 [12], which includes a simple way to evaluate group trust through calculating the average trust value

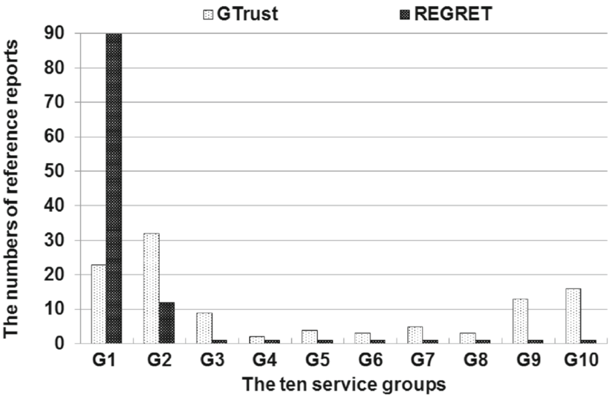


Figure 10 The numbers of reference reports in the ten service groups after the experiment

based on the trust values of individual services in the group. The REGRET model evaluates the trust value of a service provider from three perspectives: the individual dimension, the social dimension and the ontological dimension. The individual dimension is the direct reputation of the service provider offered by service consumers who had a previous interaction with the provider. The social dimension is the reputation of a group to which a service provider belongs. The ontological dimension represents the reputations of different aspects of the services offered by the provider. Even though the REGRET model considers the social dimension to reflect the influence of a group reputation on the evaluation of a trust for an individual provider, it does not consider the relationships among services in the group. In real-life applications, a complex service may consist of a number of services with different relationships and these services can also be owned by different providers. A good trust model for group trust evaluation needs to take these relationships into account. The REGRET model neglects these factors, so it cannot work properly for group service selection when services are provided by different providers. Our model overcomes these limitations through the analysis of the workflow and dependency relationships among individual services of a service group and introduces the workflow description and the degree of dependency to the trust calculation.

The other well known model for trust calculation is the certified reputation (CR) model proposed by Huynh et al. [7]. In the CR model, an agent's reputation is derived from the references of third parties which have had previous interactions with the provider. A provider can collect and present such references to consumers in order to be trusted by them. Since the CR model allows consumers to evaluate trust themselves without using a central controller, it can be adapted in a wide range of open and dynamic environments such as web-based service-oriented environments. But, there are still some limitations in the CR model. First, in the CR model, a service is represented by a single item and the evaluation of the service given by a referee is represented by a single value. In the real-world applications, it is difficult or even impossible to use a single value to represent complex contexts related to a service [14]. A service provider's performance should be evaluated from different aspects such as speed, cost, quality, reliability etc. In addition, the evaluation result may also depend on the service request and the preferences of consumers. Secondly, the CR model only focuses on the trust evaluation for an individual service based on a single provider, so it cannot handle the problem of group trust evaluation for multiple providers. In the GTrust model, we take the concept of third-party references from the CR model to evaluate the performance of a provider on a rich context based services, then evaluate the group trust value for a group of providers not only based on the performance of individual providers but also on the dependency relationships among individual services in a group.

Another related work is the priority-based trust (PBTrust) model proposed by Su et al. in 2010 [18, 19]. In the PBTrust model, the reputation of a service provider is represented by a rich context format, including attributes of the service, the priority distribution of attributes and a rating value for each attribute from a third party. The concept of 'similarity' is also introduced to measure the difference in terms of priority distribution of attributes between requested service and a refereed service in order to precisely predict the performance of a potential provider for the requested service. In addition, the concept of previous performance of a service provider on a service is also introduced to help the trust evaluation of a potential provider. Although the PBTrust model overcomes several limitations of the CR model, it still focuses on the trust evaluation of individual providers. The GTrust model is based on a partial extension of the PBTrust model and focuses on a group trust evaluation based on the four factors introduced in Section 1.

Finally, Su et al. proposed an innovative trust model for service group selection in service-oriented environments in 2011 [17]. In their approach, the reputation of each individual service is obtained from their former service consumers and recorded in the form of reference reports, which is the same as the CR model. In addition, the innovative trust model uses the rich context format and the similarity concept of the PBTrust model to represent the reputation of an individual service rather than a single value. In order to evaluate the trust value of a service group, the innovative trust model introduced the concepts of the workflow description and the degree of dependency to describe the relationships among individual services in a service group. However, in the innovative trust model, each individual service only provides its best historical reference report to the service consumer to indicate their potential performance on the service request. In addition, the innovative trust model does not introduce how to create the individual service reports based on the performance of the service group. In the GTrust model, the request model requires a certain number of reference reports from each individual service in the group and the reply module of a service group should provide the number of reference reports that satisfy the service request so as to enable the consumer to comprehensively evaluate the potential performance of the service group. Moreover, the evaluation module of the GTrust model can generate individual reference reports for individual services based on the group reference report, the workflow and dependency relationships among individual services of the group.

7 Conclusion and future work

In this paper, we proposed the GTrust model for service group selection in web-based service-oriented environments. This model is innovative because: (1) the GTrust model uses the ‘functionality coverage’ value to represent the functionalities which a potential service group can provide corresponding to the request from the consumer; (2) the GTrust model introduces the concept of ‘degree of dependency’ to represent relationships among individual services in a service group; (3) the GTrust model uses the concept of ‘third party reference’ to represent the historical performance of individual services in a service group; and (4) the GTrust model uses the concept of ‘similarity’ to measure the similarity in terms of priority distributions of attributes between the service request and reference reports. The experimental results indicate that the GTrust model has good performance for service group selection in three scenarios. In this study, we used the group performance evaluated by the consumer as the reference report for each individual service of the group without considering the different roles of each individual service. In our future work, we will handle the trust calculation for groups of services with cyclic workflows and employ learning approaches to our trust model to analyse the role of each individual service in a group.

Acknowledgments Dr. Madeleine Cincotta is acknowledged for editing the final manuscript.

References

1. Commerce, B.E., Jøsang, A., Ismail, R.: The beta reputation system. In: Proceedings of the 15th Bled Electronic Commerce Conference (2002)
2. Deitel, H., Deitel, P., Nieto, T.: e-Business & e-Commerce How to Program. Prentice Hall (2001)

3. Fang, H., Zhang, J., Thalmann, N.M.: A trust model stemmed from the diffusion theory for opinion evaluation. In: AAMAS ’13, pp. 805–812. International Foundation for Autonomous Agents and Multiagent Systems, Richland (2013)
4. Foster, I., Kesselman, C., Tuecke, S.: The anatomy of the grid: Enabling scalable virtual organizations. Int. J. High Perform. Comput. Appl. **15**(3), 200–222 (2001)
5. Heeks, R., Bailur, S.: Analyzing e-government research: Perspectives, philosophies, theories, methods, and practice. Gov. Inf. Q. **24**(2), 243–265 (2007)
6. Hermoso, R., Billhardt, H., Ossowski, S.: Role evolution in open multi-agent systems as an information source for trust. In: Autonomous Agents & Multiagent Systems/Agent Theories, Architectures, and Languages, pp. 217–224 (2010)
7. Huynh, T., Jennings, N., Shadbolt, N.: Certified reputation: How an agent can trust a stranger. In: AAMAS’06, pp. 1217–1224 (2006)
8. Huynh, T., Jennings, N., Shadbolt, N.: An integrated trust and reputation model for open multi-agent systems. J. Auton. Agent. Multi-Agent Syst. **13**(2), 119–154 (2006)
9. Jiang, S., Zhang, J., Ong, Y.: An evolutionary model for constructing robust trust networks. In: AAMAS ’13, pp. 813–820. Richland (2013)
10. Liu, S., Yu, H., Miao, C., Kot, A.C.: A fuzzy logic based reputation model against unfair ratings. In: AAMAS ’13, pp. 821–828. Richland (2013)
11. Ren, F., Zhang, M., Sim, K.: Adaptive conceding strategies for automated trading agents in dynamic, open markets. Decis. Support. Syst. **48**(2), 331–341 (2009)
12. Sabater, J., Sierra, C.: Regret: Reputation in gregarious societies. In: Proceedings of the fifth international conference on Autonomous agents, pp. 194–195 (2001)
13. Saha, D., Mukherjee, A.: Pervasive computing: A paradigm for the 21st century. IEEE Comput. **36**(3), 25–31 (2003)
14. Sensoy, M., Yolum, P.: A context-aware approach for service selection using ontologies. In: AAMAS ’06 (2006)
15. Sharmin, M., Ahmed, S., Ahamed, S.: An adaptive lightweight trust reliant secure resource discovery for pervasive computing environments. In: PERCOM’06, pp. 258–263. IEEE Computer Society (2006)
16. Singh, M.P.: Trust as dependence: a logical approach. In: AAMAS ’11, pp. 863–870. Richland (2011)
17. Su, X., Zhang, M., Mu, Y., Bai, Q.: Gtrust: An innovated trust model for group services selection in web-based service-oriented environments. In: The 12th International Conference on Web Information System Engineering, pp. 306–313. Springer (2011)
18. Su, X., Zhang, M., Mu, Y., Bai, Q.: A robust trust model for service-oriented systems. Journal of Computer and System Sciences **79**(5), 596–608 (2013)
19. Su, X., Zhang, M., Mu, Y., Sim, K.: Pbtrust: A priority-based trust model for service selection in general service-oriented environments. In: 2010 IEEE/IFIP 8th International Conference on Embedded and Ubiquitous Computing (EUC), pp. 841–848 (2010)
20. Tang, J., Seuken, S., Parkes, D.C.: Hybrid transitive trust mechanisms. In: Autonomous Agents & Multiagent Systems/Agent Theories, Architectures, and Languages, pp. 233–240 (2010)
21. Teacy, W., Patel, J., Jennings, N., Luck, M.: Travos: Trust and reputation in the context of inaccurate information sources. J. Autofocus Agent Multiagent Syst. **12**, 18396198 (2006)
22. Vogiatzis, G., MacGillivray, I., Chli, M.: A probabilistic model for trust and reputation. In: AAMAS’ 100, pp. 225–232. International Foundation for Autonomous Agents and Multiagent Systems, Richland (2010)
23. Wang, J., Sun, H.: A new evidential trust model for open communities. Comput. Stand. Interfaces **31**, 994–1001 (2009)
24. Yildiz, M.: E-government research: Reviewing the literature, limitations, and ways forward. Gov. Inf. Q. **24**(3), 646–665 (2007)
25. Yu, B., Singh, M.: An evidential model of distributed reputation management. In: AAMAS’02, pp. 294–301. ACM Press (2002)
26. Zacharia, G., Maes, P.: Trust management through reputation mechanisms. Appl. Artif. Intell. **14**(9), 881–907 (2000)
27. Zhang, M., Tang, J., Fulcher, J.: Advanced Computational Intelligence Paradigms: Theory and Applications, Studies in Computational Intelligence (SCI). Agent Based Grid Computing, vol. 115, pp. 439–483. Springer-Verlag (2008)