GTrust: An Innovated Trust Model for Group Services Selection in Web-based Service-oriented Environments

Xing Su[1], Minjie Zhang[1], Yi Mu[1] and Quan Bai[2]

[1] School of Computer Science & Software Engineering, University of Wollongong, Wollongong, NSW 2522, Australia
Email: xs702@uowmail.edu.au, {minjie,ymu}@uow.edu.au
[2] School of Computing & Mathematical Sciences, Auckland University of Technology, Auckland, New Zealand
Email:quan.bai@aut.ac.nz

Abstract. In past twenty years, the multi-agent technology has been widely employed for the development of web-based systems. Currently, agent-based service-oriented applications have been widely applied in many complex domains such as web-based e-markets, web-based grid computing, e-government and service-oriented software systems, cross Internet and organizations. In this kind of service-oriented systems, service provider (agents) and service consumer (agents) are autonomous entities and can enter and leave the environment freely. How to select the most suitable service providers according to the requested services from consumers in such an open environment is a very challenge issue. In this paper, we propose an innovated trust model-the GTrust model for group services selection in a general service-oriented environment. In our model, the trust evaluation for a group service is based on (1) the coverage rate of the requested functionalities from a group service, (2) the dependency relationships among individual services in a group, (3) reference reports from third parties for each provider of individual services in a group and (4) the similarity measurement about to what extent the reference reports can reflect the new service request in terms of priority distributions on attributes of the service. The experimental results demonstrate the good performance of the GTrust model in terms of satisfaction degree in group service selections.

1 Introduction

Multi-Agent Systems (MASs) have attracted a lot of attention from researchers in computer science, information technology, engineering, as well as other disciplines due to their abilities of autonomous decision making, collaborative problem solving, learning and adaptation abilities under open and distributed environments. In past decade, agent and multi-agent technologies have been widely employed for developing web-based service-oriented systems such as Internet-based grid systems [7], e-market [2], as well as pervasive computing systems. The web-based service-oriented environment is an open environment where most service consumer (agents) and service provider (agents) have only local views about their partners and the environment and can also join and leave the environment freely at any time. In such a dynamic environment, how to select a trustworthy service provider to fulfill a requested service from a consumer is a very challenging problem for most service-oriented applications.

'Agent Trust' is one of important research issues and many researchers in MASs had made significant effects on trust and reputations models such as probabilistic theory-based models [4], the certified reputation model [1] and evidential trust models [5]. In past decade, some trust models have been developed in service-oriented domains to help consumers evaluate the trust values of potential service providers based on different considerations. In 2000, Zacharia et. al. proposed a reputation-based trust evaluation model based on the historical performance of a provider, called the SPORAS [6], for service provider selection. In 2006, Huynh et. al. introduced a famous trust model, called the Certified Reputation (CR), to evaluate provider's trust through the third party references [1]. In 2010, Su et. al. did further work based on the CR model and developed a priority-based trust model to evaluate a trust value of a potential service provider based on the third party references, its historical performance and the priority distribution on the attributes of the requested service [3]. The most current models evaluate the trust values for individual providers. However, in recent years, many complex service requests requests from consumers cannot be handled by a single service and a group of services from different providers are needed to satisfy these service requests. Therefore, trust models focusing on the trust evaluations for single service providers cannot be directly used for the group trust evaluation and how to choose a group of services for a consumer has become a new challenge issue.

The trust evaluation for a group of service providers is different from that of for a single service provider, because there are more factors may affect the trust values of group services. The main factors include (1) the coverage rate of the requested functionalities from a service group, which determines whether the service group can satisfy all of the attribute of a service requested by a consumer, (2) the relationships among individual services in a group, (3) the performance of the individual services and reputations of individual providers in a service group, and (4) the suitability measure for the group fitting the new service based on the priority distribution on service attributes requested from a consumer.

The GTrust model has the following merits: (1) we use the 'functionality coverage' value to represent the functionalities which a potential service group can provide corresponding to the request from the consumer; (2) we introduce the concept of the 'dependency degree' to represent relationships among services in a service group; (3) we use the concept of the 'third party reference' from the PBTrust model [3] to represent the performance of individual services in the same group; and (4) we use the concept of 'similarity' to measure the similarity in terms of priority distributions on attributes between historical services of group members and requested services.

The rest of paper is organized as follows. Section 2 is the problem description. The basic components of the GTrust model is briefly introduced in Section 3. The detail descriptions of each components in the GTrust are introduced in detail Section 4. Finally, the paper is concluded and future work is outlined in Section 5.

2 Problem Description and Definitions

In general, a service can be described by a number of attributes such as price, time, quality etc. For different requests, the priority distribution on each attribute for the same service can be different. In order to describe the attributes and their corresponding priority values of a service, we make a service description in a formal way.

Suppose that a requested service includes n attributes and each attribute has a priority value to describe the request for the service. A service can be represented by n attributes and their corresponding priority values as follows.

Definition 1. *A service description $SDes$ is defined by a $2 \times n$ matrix.*

$$SDes = \begin{pmatrix} A_1 \; A_2 \; A_3 \; ... \; A_i \\ P_1 \; P_2 \; P_3 \; ... \; P_i \end{pmatrix} \tag{1}$$

where i indicates the number of attributes in requested service, A_i indicates the i^{th} attribute of the requested service, P_i is priority value of the A_i and $\sum_{i=1}^{n} P_i = 1$.

Definition 2. *A reference report Rf is defined as a 2-tuple, Rf =<SDes, Ratings>, where $SDes$ is the* service description *of the service requested by the pervious consumer (referee) and Ratings is defined as a vector, $Ratings =< R_1, R_2, ..., R_i >$, where R_i represents the performance rating value of the provider on i^{th} attribute of the requested service and R_i is a value between [0, 1], where 0 and 1 represents the worst and best performance of a provider, respectively.*

To deal with a complex request, a number of individual services need to form a group with certain workflows and dependency relationships among individual services in the group. Even if two groups have the same individual services, if the workflows and dependency relationships of the individual services in the two groups are different, the two groups may have different performance on the requested service. For example, two groups have the same individual services $S1$, $S2$, $S3$, $S4$ and $S5$, but different workflows and dependency relationships as follows.

Fig. 1. Workflows and dependency relationships of services in Two Groups

In Fig. 1, Group 1 has a sequential workflow to process from $S1$ to $S5$, i.e. the later service depends on the former service. However, the workflow is different in Group 2, $S1$, $S2$, $S3$ and $S4$ can work at the same time and $S5$ can only be conducted when the former four services is finished. We can see that there are no dependency relationships among S1 to S4 but 4 dependency relationships exist between S5 with other 4 services. In another word, S5 depends on $S1$, $S2$, $S3$ and $S4$, respectively. In order to identify relationships among services in a group, we define the dependency degree as follows.

Definition 3. *A dependency degree λ is defined as a value in-between [0, 1], where 0 represents an independency relationship between two services and 1 denotes the strongest dependency relationship between two services.*

We also use a matrix to describe the workflow of a group by using the following definition.

Definition 4. *A workflow description $WDes$ of a group is represented by a $n \times n$ matrix, where 'n' is the number of individual services in the group. The $WDes$ is defined by Equation 2 as follows.*

$$WDes = \begin{pmatrix} \lambda_{11}, \lambda_{12}, \lambda_{13}, ..., \lambda_{1n} \\ \lambda_{21}, \lambda_{22}, \lambda_{23}, ..., \lambda_{2n} \\ ..., ..., ..., ..., ... \\ \lambda_{n1}, \lambda_{n2}, \lambda_{n3}, ..., \lambda_{nn} \end{pmatrix} \tag{2}$$

where λ_{ij} represents the value of dependency degree between service i and service j. $\lambda_{ij} = 0$ represents there is no dependency relationship between service i and the service j. If $\lambda > 0$, there exists a dependency relationship between service i and service j and service j depends on service i.

Definition 5. *A service reply SR is defined as a 2-tuple, SR =<$WDes$, $RfSet$>, where $WDes$ is the workflow description of a group and $RfSet$ is the set of reference reports of each services in the group.*

3 Basic Modules of the GTrust Model

The GTrust model consists of three modules which are the Request Module, the Reply Module and the Priority-based Group Trust Calculation Module. The working procedure of the GTrust model can be described as follows. When a consumer requests a complex service (1) the request module will generate the service requirements and broadcast it to potential providers; (2) potential service groups with requested services will reply the service request by using the reply module; (3) the consumer will evaluate the trust value for each potential service group using the priority-based group trust calculation module and choose the best service group based on the trust value of the group; With the reference report, the members of the service group can dynamically update their reference report.

4 The Principle of the GTrust model

In this section, four major modules of the GTrust model are introduced in detail in the following four subsections, recpectively.

4.1 The request module

The objective of the Request Module is to create a *service request* based on the request from a consumer. For example, consumer C in an e-market environment requests a complex service described by 5 attributes, i.e. cost, speed, quality, color and warranty with corresponding priority values for each attribute as 0.1, 0.4, 0.2, 0.1 and 0.2, respectively. Based on the service request, the Request Module will generate a service request in the format of *service description*, (recall Definition 1) as follows:

$$SDes = \begin{pmatrix} Cost & Speed & Quality & Color & Warranty \\ 0.1 & 0.4 & 0.2 & 0.1 & 0.2 \end{pmatrix}$$

Then, the service request will be broadcasted to the system to discover potential service providers.

The above example will be used for the explanation in rest modules.

4.2 The reply module

The function of the reply module is to generate a service reply to describe a service group and the individual services in the group. For example, if a Service Group (SG) intends to offer the requested service, the reply module of SG will collect the following information: the group description of SG, and reference reports of each individual services in SG. Each individual service in SG will present its best reference report. The reply module will create a service reply, (recall Definition 3) for SG in the following format including the *workflow description* of SG and a set of reference reports for five members, respectively, $SR = < GDes, RfSet >$

4.3 The priority-based group trust calculation module

The main purpose of this module is to evaluate the trust value for each potential service group based on the *service reply* SR and service request.

Because a service group is composed of different individual services owned by different providers, the group ability to handle a new service depends on the abilities of individual members. We use a *group service description* to formally describe a group ability by extracting useful information from reference reports provided by group members.

Definition 6. *A group service description* $GSDes$ *is represented by a* $m \times n$ *matrix.* m *is the number of the individual services in a group and n is the number of attributes in service request.* $GSDes$ *is defined by the following matrix.*

$$GSDes = \begin{pmatrix} A_1 & A_2 & ... & A_n \\ P_{11} & P_{12} & ... & P_{1n} \\ P_{21} & P_{22} & ... & P_{2n} \\ ... & ... & ... & ... \\ P_{m1} & P_{m2} & ... & P_{mn} \end{pmatrix} \tag{3}$$

where A_i *indicates the* i^{th} *attribute of the requested service. The* i^{th} *row (excluding the first row) in the matrix represents the priority distribution on a pervious service completed by the corresponding group member and* P_{ij} *represents the priority value on the* j^{th} *attribute of the requested service on that service, where* $P^{ij} = m$*, if the pervious service dose not contain the* j^{th} *attribute; otherwise* P_{ij} *is in-between [0,1], where 0 and 1 represent the highest and lowest priority values, respectively. By using Equation 3, the comprehensive ability of a service group can be described.*

Functionality coverage calculation The purpose of functionality coverage calculation is to measure whether the functionalities offered by a potential service group can cover all the attributes in the service request. A functionality coverage is defined by the following definition.

Definition 7. *A functionality coverage* $FCov$ *is defined as a vector,* $FCov =< ACov_1, ACov_2,\ ACov_3, ...ACov_i >$*, where* $ACov_i$ *is a value in-between [0, 1], which represents the functionality coverage value of a service group on* i^{th} *attribute in the service request.*

$ACov_i$ can be calculated based on the information in $GSDes$ (recall Definition 6) as follows.

$$ACov_i = \frac{m - MS_i}{m} \tag{4}$$

where $ACov_i$ represents the functionality coverage value of a service group on i^{th} attribute of the requested service, m represents the number of the individual services in a group and MS_i represent the number of ‘m’ (i.e. how many members cannot cover the i^{th} attributes) in the i^{th} column of the matrix $GSDes$. If the functionality coverage on i^{th} attribute is ‘0’, we can say that this service group is not suitable to conduct the requested service.

Group similarity calculation The objective of the group similarity calculation is to measure the similarity of the priority distribution between a group service and the requested service. In the GTrust model, the priority distribution of a service is represented by a vector. To compare the similarity between two vectors, we can use the concept 'dot product' of the two vectors. To calculate the similarity of priority distribution between a group service and the requested, we can use a vector $GPV =< GP_1, GP_2, GP_3, ...GP_n >$ to represent the priority distribution in a group of services extracted from reference reports, where GR_i is the priority value on the $i^(th)$ attribute in a group service. GP_i is calculated by the following formula.

$$GP_i = \sum_{i=1}^{m} P_{ij} \tag{5}$$

where, P_{ij} is the priority value of the i^{th} individual service in the group on j^{th} attribute of the requested service.

We can calculate each element in Vector GPV, then normalize two vectors if necessary before using the dot product. The group similarity calculation can be obtained by the following formula.

$$GSim = \frac{\sum_{i=1}^{n} NGP_i \cdot NP_i}{\sqrt{(\sum_{i=1}^{n}(NGP_i)^2) \cdot (\sum_{k=1}^{n}(NP_i)^2)}} \tag{6}$$

where $GSim$ is the similarity between the priority distribution of the requested service and a service group, NP_i and NGP_i represent the normalized priority values of the i^{th} element of priority distribution vector in the requested service and the priority distribution vector in the service group, respectively.

Group rating calculation The purpose of group rating calculation is to predict the performance of a service group on each attribute of the requested service based on the reference reports. The rating for the group's potential performance in j^{th} attribute is calculated as follows.

$$GRating_j = \frac{\sum_{i=1}^{m} FRating_{ij}}{m} \tag{7}$$

where 'm' is the number of individual services in the service group and $FRating_{ij}$ represents the final rating of the i^{th} individual service, after considering the dependency degrees with other services in the group, on the j^{th} attribute in the group service. $FRating_{ij}$ is calculated by the following formula.

$$FRating_{ij} = Rating_{ij} - \frac{\sum_{k=1}^{n} \lambda_{ki} \cdot (1 - FRating_{kj})}{n} \tag{8}$$

where n represents the number of the individual services which the i^{th} service depends on, $Rating_{ij}$ is the rating of the i^{th} individual service on j^{th} attribute shown in the reference report and $FRating_kj$ is the the final performance rating of the k^{th} dependency service on j^{th} attributes, and $\lambda_k i$ is the dependency degree of the i^{th} individual service depending on the k^{th} dependency service.

Final trust calculation After functionality coverage calculation, similarity calculation and group rating calculation, we can calculate the final trust value *Trust* for a service group by the following formula.

$$Trust = GSim \cdot \sum_{i=1}^{n} P_i \cdot ACov_i \cdot GRating_i \tag{9}$$

where $GSim$ is the similarity value, P_i is the priority value of the i^{th} attribute in the requested service, $ACov_i$ is the functionality coverage value of a service group on the i^{th} attribute of the requested service and $GRating_i$ represents the group rating after considering the dependency relationships and workflows of services in the group.

5 Conclusion and Future Work

In this paper, we proposed the GTrust model for group services selection in web-based service-oriented environments. In current stage, we use the group performance evaluated by the consumer as the reference report for each members of the group during updating their historical records without considering the different roles of each individual services. In the future, we will employe agent learning approach to our trust model to analyze the performance of each member in a group.

References

1. Huynh, T., Jennings, N., Shadbolt, N.: Certified reputation: How an agent can trust a stranger. In: AAMAS'06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems. pp. 1217–1224 (2006)
2. Ren, F., Zhang, M., Sim, K.: Adaptive conceding strategies for automated trading agents in dynamic, open markets. Decision Support Systems 48(2), 331–341 (2009)
3. Su, X., Zhang, M., Mu, Y., Sim, K.: Pbtrust: A priority-based trust model for service selection in general service-oriented environments. In: Embedded and Ubiquitous Computing (EUC), 2010 IEEE/IFIP 8th International Conference on. pp. 841 –848 (dec 2010)
4. Teacy, W., Patel, J., Jennings, N., Luck, M.: Travos: Trust and reputation in the context of inaccurate information sources. Journal of Autofocus Agent and Multiagent Systems (JAAMAS) 12, 183198 (2006)
5. Wang, J., Sun, H.: A new evidential trust model for open communities. Comput. Stand. Interfaces 31, 994–1001 (September 2009), http://portal.acm.org/citation.cfm?id=1555011.1555187
6. Zacharia, G., Maes., P.: Trust management through reputation mechanisms. Applied Artificial Intelligence 14(9), 881–907 (2000)
7. Zhang, M., Tang, J., Fulcher, J.: Advanced Computational Intelligence Paradigms: Theory and Applications, Studies in Computational Intelligence (SCI), vol. 115, chap. Agent Based Grid Computing, pp. 439–483. Springer-Verlag (2008)