RESEARCH ONLINE

University of Wollongong
Research Online

University of Wollongong Thesis Collection

University of Wollongong Thesis Collections

2011

Agent-based trust models for service provider selection in service-oriented environments

Xing Su
University of Wollongong, xs702@uowmail.edu.au

Recommended Citation

Su, Xing, Agent-based trust models for service provider selection in service-oriented environments, Master of Computer Science - Research thesis, School of Computer Science and Software Engineering, University of Wollongong, 2011. http://ro.uow.edu.au/theses/3568

Research Online is the open access institutional repository for the University of Wollongong. For further information contact Manager Repository Services: morgan@uow.edu.au.

RESEARCH ONLINE

UNIVERSITY OF WOLLONGONG

COPYRIGHT WARNING

You may print or download ONE copy of this document for the purpose of your own research or study. The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site. You are reminded of the following:

Copyright owners are entitled to take legal action against persons who infringe their copyright. A reproduction of material that is protected by copyright may be a copyright infringement. A court may impose penalties and award damages in relation to offences and infringements relating to copyright material. Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.

Agent-based Trust Models for Service Provider Selection in Service-oriented Environments

A thesis submitted in fulfillment of the
requirements for the award of the degree

Master by Research

from

UNIVERSITY OF WOLLONGONG

by

Xing Su

School of Computer Science and Software Engineering
December 2011

© Copyright 2011

by

Xing Su

All Rights Reserved

Dedicated to
Tieguang Su and Yang Liu

Declaration

This is to certify that the work reported in this thesis was done by the author, unless specified otherwise, and that no part of it has been submitted in a thesis to any other university or similar institution.

Xing Su
December 12, 2011

Abstract

In the past twenty years, the multi-agent technology has been widely employed for developing agent-based systems. Currently, agent-based service-oriented applications have been widely applied in many complex domains such as e-markets, grid computing, e-governments and service-oriented software systems, across Internet and organizations. In this kind of service-oriented multi-agent systems, service providers (agents) and service consumers (agents) are autonomous entities and can enter and leave environments freely. How to select the most suitable service providers according to the requested services from consumers in such an open and dynamic environment is a very challenging issue.

The objectives of this thesis include (1) studying the challenging issues of trust-based service provider selection, (2) investigating the current approaches of trust models for service provider selection in general service-oriented multi-agent systems, and (3) developing new solutions for service provider selection to overcome several limitations in current approaches.

In this thesis, two trust models are proposed and developed. One is a Priority-based Trust (PBTrust) model for single service provider selection. The other is a Group Service Trust (GTrust) model for group service providers selection when a complex service requests multiple service providers.

The designing purpose of the PBTrust model is to help service consumers in multi-agent systems to select the most suitable single service providers. To deal with the provider selection problem, firstly the PBTrust model uses a rich context service description to represent service requests by confederating different attributes of a service and uses priority values to distinguish the importance of these attributes. This feature allows more objective evaluations on both required services and providers' reputations. Moreover, the PBTrust model uses a relatively easy way to describe the different attributes of a service. Finally, the PBTrust model introduces the concept of experience weight which can avoid subjective and cheating references.

Being different with the PBTrust model, the GTrust model is designed for group service providers selection in service-oriented environments. Currently, many complex services are hard for single providers to fulfill the requests. Therefore, several service providers need to form groups to conduct the services. Developing trust model for group service providers selection is a hard topic, due to the structure of services composition and dependency relationships among services owned by different providers, the reputations of individual services and impacts of individual services on group performance in terms of their trust values towards to the group trust evaluation. The GTrust model offers several innovated mechanisms to help a consumer accurately evaluating the trust value for a group of services by taking the above features into account during group service providers selection. The GTrust model evaluates the trust value for a group of services by considering (1) the coverage rate of the requested functionalities from a service group, (2) the dependency relationships among individual services in a group, (3) the 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 requested service.

Acknowledgements

Studying abroad is a tedious and interesting experience. It would be hard for me to finish my research without the help from many people.

I am indebted to appreciate my supervisors, Associate Professor Minjie Zhang and Yi Mu. Their constant commitment and guidance was instrumental in the completion of this thesis, and in making my study a fulfilling experience. I am grateful to Associate Professor Minjie Zhang for her kind help, encouragement and patient proofreading my thesis and research papers. I am also delightful for Associate Professor Yi Mu's support for my studies. Furthermore, I thank the School of Computer Science and Software Engineering and the University of Wollongong for the financial support to my conference attendance and CSIRO for the Summer Scholarship for my research training in CSIRO Tasmania ICT Centre.

I am delightful for Dr. Quan Bai's support and help during my studies. My thanks are extended to Mr. Dayong Ye, who often discusses with me in the research laboratory and enriches my knowledge.

I would like to express my deepest gratitude to my parents, Tieguang Su and Yang Liu, who always make their financial support, encouragement, understanding and love to me. Without their help, this thesis would not be finished.

Finally, thanks to all the anonymous reviewers of my research papers, and all my other dear friends and relatives who have supported me.

Publications and Submissions

Publications:

- Xing Su, Minjie Zhang, Yi Mu and Quan Bai, GTrust: An Innovated Trust Model for Group Services Selection in Web-based Service-oriented Environments. In *Proceedings of the 12th International Conference on Web Information System Engineering (WISE 2011)*, Sydney, Australia, (Ranked 'A' in ERA), accepted in July 2011.

- Xing Su, Minjie Zhang, Yi Mu and KwangMong Sim, PBTrust: A Priority-Based Trust Model for Service Selection in General Service-Oriented Environments. In *Proceedings of the 2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (TrustCom 2010)*, Hong Kong, China, (Ranked 'A' in ERA), pp. 841-848, 2010.

Submissions:

- Xing Su, Minjie Zhang and Yi Mu, Trust-Based Group Services Selection in Web-based Service-Oriented Environments. *Journal of Concurrency and Computation: Practice and Experience* (Ranked 'A' in ERA), submitted in August 2011.

- Xing Su, Minjie Zhang, Yi Mu and Quan Bai. A Priority-Based Trust Model for Service Selection. *the Journal of Intelligent Systems* (Ranked 'B' in ERA), submitted in April 2011.

- Quan Bai, Xing Su, Qing Liu, Andrew Terhorst, Minjie Zhang and Yi Mu, Case-based Trust Evaluation from Provenance Information. *The 10th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom 2011)* (Ranked 'A' in ERA), submitted in July 2011.

Contents

Abstract v

Acknowledgements vii

Publications and Submissions viii

1 Introduction 1

1.1 Background Knowledge 2

1.1.1 Service-oriented environments 3

1.1.2 Trust in multi-agent systems 4

1.2 Research Issues in Service Provider Selection 6

1.2.1 Research issues in single service provider selection 6

1.2.2 Research issues in group service providers selection 7

1.3 Motivation of This Research 7

1.4 Contributions of This Thesis 8

1.5 Structure of This Thesis 10

2 Literature Review 11

2.1 Classifications of Trust Models 11

2.1.1 A classification method by Esfandiari and Chandrasekharan . . 11

2.1.2 A classification method by Ramchurn, Huynh, and Jennings . . 12

2.1.3 A classification method by Sabater and Sierra 15

2.1.4 A new classification method proposed by this study 17

2.2 Representative Trust Models 18

2.2.1 Centralised trust models 18

2.2.2 Decentralised trust models 21

2.3 Summary 24

3 A Priority-Based Trust Model **25**
3.1 Problem Description and Definitions 25
3.2 The Basic Principle of the PBTrust Model 27
3.2.1 The request module 27
3.2.2 The reply module 27
3.2.3 The priority-based trust calculation module 28
3.2.4 The evaluation module 32
3.3 Experiments 33
3.3.1 Experiment 1: evaluating the impacts of *Sim* and *SInd* 34
3.3.2 Experiment 2: evaluating the impact of *EW* 38
3.4 Summary 39

4 A Group Service Trust Model **41**
4.1 Problem Description and Definitions 41
4.2 The Principle of the GTrust Model 44
4.2.1 The request module 44
4.2.2 The reply module 44
4.2.3 The priority-based group trust calculation module 45
4.2.4 The evaluation module 49
4.3 Experiment 49
4.3.1 Experiment setting 49
4.3.2 Experimental results and analysis 51
4.4 Summary 59

5 Conclusion **61**
5.1 Major Contributions of This Thesis 61
5.1.1 The contributions of the PBTrust model 62
5.1.2 The contributions of the GTrust model 62
5.2 Remaining Problems and Future Work 63

Bibliography **65**

List of Tables

3.1 Average Similarity Degrees of Each Provider Group 35
3.3 Differences of Service and Trust Description in the CR Model and the PBTrust Model 35
3.2 Service Providers' References 36

4.1 The References Reports of Twenty Service Providers 50
4.2 The Similarity, Functionality Coverage and Ratings of Four Service Groups in Five Dependency Degrees in Scenario 1 52
4.3 The Trust Values of Four Service Groups in Scenario 1 52
4.4 The Similarity, Functionality Coverage and Ratings of Four Service Groups in Five Dependency Degrees in Scenario 2 55
4.5 The Trust Values of Four Service Groups in Scenario 2 56
4.6 The Similarity, Functionality Coverage and Ratings of Four Service Groups in Five Dependency Degrees in Scenario 3 57
4.7 The Trust Values of Four Service Groups in Scenario 3 57

List of Figures

2.1 The Classification of Current Trust Models Proposed by Ramchurn *et al.* 13
2.2 The New Classification of Current Trust Models 17

3.1 The Dot Product of Two Vectors 31
3.2 The Satisfaction Degrees of the Services Selected by the CR Model and the PBTrust Model in Experiment 1 38
3.3 The Satisfaction Degrees of the Services Selected by the CR Model and the PBTrust Model in Experiment 2 39

4.1 Workflows and Dependency Relationships of Services in Two Groups . 42
4.2 The Process of the GTrust Model 43
4.3 Services Workflow of the SG 45
4.4 Workflows of Individual Services in A Group in Three Scenarios 51
4.5 Trust Value of Service Groups Selected by Two Trust Models in Scenario 1 53
4.6 Trust Value of Service Groups Selected by Two Trust Models in Scenario 2 56
4.7 Trust Value of Service Groups Selected by Two Trust Models in Scenario 3 58
4.8 Trust Value of Service Groups Selected by the GTrust Model and the Benchmark Model in Three Scenarios 59

Chapter 1

Introduction

In the past twenty years, Multi-Agent Systems (MASs) have attracted much attention from researchers in computer science, information technology, engineering, as well as other disciplines, due to their abilities of autonomous learning [60, 8, 19], decision making [7, 65, 24], collaborative problem solving [31, 32, 2, 5], and adaptation abilities under open and distributed environments [36, 23, 44]. Currently, agent and multi-agent technologies have been also widely employed for developing service-oriented systems such as Internet-based grid systems [21, 56, 14, 66], e-markets [16, 30, 44], pervasive computing systems [48, 41, 53], as well as e-governments [25, 61, 33].

A Multi-Agent System (MAS) is an intelligent system composed by multiple interacting agents. In a MAS, an agent can perform tasks individually or collaboratively with other agents when dealing with complex problems. In a service-oriented MAS, an agent can play two roles, a service provider or a service consumer. For simplicity, we use terms 'consumer' or 'provider' instead of the term 'agent' in some part of this thesis. Currently, most of service-oriented MASs are situated in web-based environments [38, 4, 10]. These types of environments are open and dynamic and agents can leave or enter environments freely. For a consumer, how to choose a trustable provider or a group of providers to fulfil its requested service is a very important research topic in both MAS research and agent-based service-oriented applications.

Ramchurn *et al.* published the paper 'Trust in multi-agent systems' in 2004 [43]. In this paper, Ramchurn *et al.* summarised the 'Trust problem' in MASs as a 'How, Who, When' problem which can be explained as follows.

- 'How' means that an agent chooses what protocols or mechanisms offered by a MAS to interact with other agents.
- 'Who' means that an agent chooses which service provider/s to interact with.
- 'When' means that an agent chooses what time to interact with the chosen service

provider/s.

In summary, the 'How, Who, When' problem can be used to describe the core of the trust problem for service provider selection in service-oriented MASs. However, the characteristics of service-oriented environments and MASs make some difficulties for service provider selection.

The **objectives** of this thesis are

1 To study the challenging issues of trust-based service provider selection in general service-oriented MASs.

2 To investigate the current approaches of trust models for service provider selection in general service-oriented MASs.

3 To develop new solutions for service provider selection to overcome several limitations in current existing approaches in both single provider selection and group providers selection.

This chapter provides a brief overview of the service provider selection in service-oriented MASs, introduces the challenging issues of trust-based service provider selection, highlights the contributions of this thesis and gives the structure of this thesis. Section 1.1 introduces some background knowledge and concepts related to this research, which include the characteristics of service-oriented environments and the trust evaluation in MASs. Section 1.2 addresses the challenges in service provider selection in service-oriented MASs. In Section 1.3, the motivation of this research is given. Section 1.4 highlights the contributions of this thesis. Section 1.5 gives the structure of the thesis as well as contents of each chapter.

1.1 Background Knowledge

In this section, some important knowledge and concepts about service provider selection in service-oriented MASs are introduced. Section 1.1.1 focuses on introducing the features of general service-oriented environments. Section 1.1.2 introduces trust evaluation in service-oriented MASs.

1.1.1 Service-oriented environments

Generally, a service-oriented environment is an open and dynamic environment. A MAS based on this environment is composed by a number of heterogeneous and distributed service providers and consumers (i.e. agents). These agents use their services as source to interact with other agents in the system. In order to clearly introduce the characteristics of general service-oriented environments, we borrow the classification proposed by Ramchurn *et al.* in the paper 'Trust in multi-agent systems' in 2004 [43] to divide the features of MASs in service-oriented environments into two categories, the individual-level features and the system-level features. The detail introduction of this classification will be given in Chapter 2.

The individual-level features indicate the characteristics of the individual agents in a system, which include:

- **Autonomy**

  An agent in the system has the ability or partial ability to control itself and make decisions by itself. Since agents in MASs are intelligent agents, the autonomous ability is a sign for an intelligent agent.

- **Self-interest**

  An agent in service-oriented systems normally pays attention to its own benefit. When an agent interacts with other agents, the purpose of the agent is trying to maximise its own benefit.

- **Local views**

  It is hard for an agent in an open system having full information of other agents or a global view about the whole system. Since the scale of a service-oriented MAS is big and the system is highly dynamic, it is hard or impossible for an agent to have all of the newest local and global information of the system.

The system-level features demonstrate the characteristics of a service-oriented MAS, which include:

- **Dynamic environments**

  An agent can freely join and leave the system at any time. The number of agents in the system is changing all the time.

- **Decentralised control**

  There is no a centralised controller to control the decision process of all agents in the system. This feature makes difficult for an agent to dynamically get the newest global information about the whole system situations. Therefore, designing a centralised controller for the system is nearly impossible.

- **Complex relationships**

  The relationships among agents in MASs are complicated. In a system, an agent may have multiple roles such as a service consumer, a service provider or a third party, which means that an agent can offer a service, request a service, and evaluate a service. Because of the multiple roles, the agent can have different relationships with other agents. If two agents offer the same service, they may have a completion relationship. If an agent offers a service to another agent, they can have a collaboration relationship.

- **Distributed sources**

  There is no central database designed for this kind of systems to store the information. The information are separatively stored in individual agent systems.

- **Different service requests**

  The service requirements can be different from case to case. Even if two service consumers request for the same service, they often pay attention to different aspects of the service.

In summary, the individual-level and system-level characteristics of service-oriented MASs increase the uncertainty for service provider selection. Therefore, how to reduce the uncertainty by considering these characteristics is the main design purpose of most trust models.

1.1.2 Trust in multi-agent systems

'Agent Trust' is one of important research issues in MASs [13, 15, 34, 9]. The definition of trust proposed by Ramchurn *et al.* in paper [43] is that *'Trust is a belief an agent has that the other party will do what it says it will (being honest and reliable) or reciprocate (being reciprocative for the common good of both), given an opportunity to defect to get higher payoffs'.* The core of the trust problem is to solve the 'How, Who, When' problem. However, since the characteristics of service-oriented MASs

make some difficulties for service provider selection, we need to use the limited and distributed information to reduce the uncertainty caused by these characteristics [6].

Many researchers in MASs have made significant effects on trust and reputation models such as probabilistic theory-based models [57], the certified reputation models [28] and evidential trust models [59]. In past decade, some trust models have been developed in service-oriented domains to help consumers evaluating the trust values of potential service providers based on different considerations [62, 29, 26]. In 2000, Zacharia *et al.* proposed a reputation-based trust evaluation model based on the historical performance of a provider, called the SPORAS [64], 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 [28]. However, there are still some problems in current trust models.

Firstly, most of trust models evaluate the trust of a potential service provider for a service request from the reputations offered by the former service consumers to the same service. This evaluation method may neglect the difference between the current and former service requests in terms of the context of the service. For example, 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, it is hard or impossible to use a single value to express complex contexts related to a service [52]. In contrast, a service provider's performance can be evaluated from different aspects such as speed, cost, quality, reliability etc. In addition, the evaluation results may also depend on constrains of a particular service, as well as the preferences of service consumers.

Secondly, most of current trust models are developed to evaluate the trust values of individual service providers. However, in recent years, many complex service requests from service consumers cannot be handled by single services and a group of services from different service providers need to combine together with certain structures and workflows to satisfy these service requests [58, 55]. Therefore, the trust models focusing on the trust evaluations for single service providers cannot deal with the group trust evaluation problem, since the structure and relationships among group members also play important roles on the trust value of the overall service offered by a group. Therefore, how to choose a group of services for a service consumer has become a new challenge for service provider selection.

1.2 Research Issues in Service Provider Selection

Service provider selection can be classified as single service provider selection and group service providers selection.

1.2.1 Research issues in single service provider selection

For the single service provider selection, the major research issues are listed below.

1 Trust information retrieval

If a service consumer wants to find the trust information of a potential service provider, the service consumer often has two choices which are the direct experience with the provider or the reputations of the provider evaluated by third parties. If the service consumer has direct interaction with the potential service provider before, it is very lucky for the service consumer to use the former experience. However, it is not very often for a service consumer to have such experience. Therefore, most of time, the service consumer needs the reputations from third parties. However, searching for reputations of a potential service provider also leads to new problems which are:

- How to effectively search for the useful trust information in the system, since the information is stored in individual agents.
- Whether the third parties want to share the trust information with the service consumer, since most of agents are self-interested in most of service-oriented systems.
- Whether the trust information offered by the third parties can realistically reflect the behaviours of the potential service provider, since the third parties may have different relationships with the potential service provider.

2 Trust information aggregation

If a service consumer collects a number of trust information for a potential service provider, how to summarise all of the collected trust information to generate the trust value for a potential service provider is also a challenging task, since different third parties may have different views on the same potential service provider.

3 Trust information description

If an agent has the trust information of another agent, how to quantify this trust information and make the information can be exchanged with other agents and understood by other agents is a challenging issue.

4 **Full context representation**

In most of current trust models, a service is represented by a single item and the trust value of the service given by a referee is represented by a single value. In the real world, it is hard or impossible to use a single value to represent complex contexts related to a service [52]. In contrast, a service provider's trust value should be evaluated from different aspects associated with the context information.

1.2.2 Research issues in group service providers selection

Since the group service providers selection problem emerges in recent years, some new challenges come out for the group service providers selection. Beings different from the single service provider selection, the group service providers selection problem needs to consider additional issues which have impact on group trust evaluation. These issues include:

1 **Structure and workflows of a service group**

The group services are formed in certain structures and workflows. For the same services, different structures and workflows can have different trust values.

2 **Dependency relationships**

There exists the situation that one service needs the result of another service as a source. Therefore, the quality of the former service has impact on the quality of later service.

3 **Efficiency**

That means whether a service group has redundant services or whether a service group can offer all of the functionalities to fulfill a service request.

1.3 Motivation of This Research

The ***motivation*** of this thesis is to solve some research issues listed in Section 1.2 by developing two trust models for single service provider selection and group service providers selection, respectively.

The trust model for single service provider selection mainly solves the following issues:

- **Trust information aggregation** – by providing the trust aggregation formula which considers a provider's general experience, similarity between requested service and referenced service, suitability of the service offered by provider, and timestamp of the reference.
- **Full context representation** – by proposing a formal way of full context service description.

The trust model for group service providers selection mainly solves the following issues:

- **Trust information aggregation** – by developing the trust aggregation formula which considers issues of similarity, suitability and functionality coverage rate.
- **Full context representation** – by proposing a formal way for full context service description.
- **Structure and workflows** – by developing a method for workflow analysis and description.
- **Dependency relationships** – by developing a formula for dependency relationship calculation.
- **Efficiency** – by proposing a concept of functionality coverage rate.

1.4 Contributions of This Thesis

This thesis proposes two innovative trust models which are the PBTrust model for single service provider selection and the GTrust model for group trust evaluation.

The PBTrust model calculates the reputation of a potential service provider from four perspectives, which are 1) the provider's experience on the service, 2) the similarity of priorities distributions on attributes between the referenced service and the requested service, 3) the suitability of the potential provider for the requested service, and 4) the time effectiveness of ratings from third party references. The PBTrust model can give a robust and accurate evaluation of the trustworthiness of service providers in open dynamic environments.

The **contributions of the PBTrust model** are that:

1 The PBTrust model considers the attributes of a service and uses priorities to distinguish the importance of different attributes. This feature allows more objective evaluations on both required services and providers' reputations.

2 The PBTrust model uses a relatively easy way to describe different attributes of a service.

3 The PBTrust model introduces the concept of 'experience weight' which can avoid subjective and cheating references.

4 The PBTrust model uses general experience to describe service provider's experience.

The GTrust model offers an innovated way for group trust evaluation by considering three new challenges emerging in group trust evaluation, which are 1) the structure of the service group, 2) the dependency relationships among group members, and 3) the efficiency of the service group.

The **contributions of the GTrust model** are that:

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 'dependency degree' to represent relationships among services in a service group.

3 The GTrust model uses the concept of 'third party reference' from the PBTrust model to represent the former performance of individual services in a service group.

4 The GTrust model uses 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 detail design of the PBTrust model and the GTrust model are introduced in Chapters 3 and 4, respectively.

1.5 Structure of This Thesis

The rest of this thesis is organised as follows.

Chapter 2: This chapter gives the literature review. In this chapter, three existing classifications of current trust models are introduced first. Then, a new classification from this study is proposed and a number of important trust models following the new classification are reviewed in detail.

Chapter 3: This chapter proposes the PBTrust model for single service provider selection. In this chapter, each module of the PBTrust model is introduced first. Then, the trust evaluation mechanism of the PBTrust model is presented in detail and the experiments are demonstrated, which include the design of the experiments, experimental results and analysis.

Chapter 4: This chapter presents the GTrust model for group service providers selection. The trust evaluation mechanism of the GTrust model is introduced in detail, the experimental results are demonstrated and experimental analysis and discussions are provided.

Chapter 5: The thesis is concluded by this chapter. This chapter highlights the contributions of the thesis. Then, limitations of the PBTrust model and the GTrust model are discussed and the directions of future research are pointed out.

Chapter 2

Literature Review

This chapter firstly gives three general classifications of trust models, and then presents a new classification method for current trust models by the consideration of the features of service-oriented applications during this study. Under the new classification, this chapter gives the detail reviews for several important trust models which are related to this research. Section 2.1 introduces three general classifications and a new classification of trust models, which is proposed by this study. Section 2.2 is detail reviews of several important models under the new classification.

2.1 Classifications of Trust Models

In the past twenty years, many trust models have been proposed based on different considerations and perspectives to deal with service provider selection problem in service-oriented MASs. In this section, three classification methods for current trust models are introduced and analysed, and a new classification is proposed.

2.1.1 A classification method by Esfandiari and Chandrasekharan

In 2001, Esfandiari and Chandrasekharan published a paper, 'On How Agents Make Friends: Mechanisms for Trust Acquisition' [18]. In this paper, they classified trust models based on what views on trust that a trust model holds. Form this perspective, trust models are classified into *cognitive trust models* and *mathematical trust models*.

Cognitive trust models

In the view of cognitive trust models, trust is composed of underlying beliefs and a function of the value of these beliefs. This view of trust is similar with the trust existing

among people in human society. The mental states of an agent result in whether an agent trusts another agent and what level an agent trusts another agent. A typical example of this kind of trust models was proposed by Castelfranchi and Falcone in 1998 [12]. In their trust model, the trust is divided into three beliefs, which are

- Competence belief
- Dependence belief
- Disposition belief

The disposition belief is further divided into

- Willingness belief
- Persistence belief

All of these beliefs combine together to form the trust of an agent to another agent. This kind of trust models are not very popular, since it is hard to identify the basic beliefs an agent having on another agent.

Mathematical trust models

Mathematical trust models neglect the effect of basic beliefs on trust and only use trust metric to predict the future behaviour of an agent based on its former behaviours. The trust in this kind of trust models is similar with playing a game by calculating how much utility an agent gains in an interaction. Mathematical trust models are also be called game-theoretical trust models. The ways that calculates the utility and aggregates the reputations varies from model to model. There are a number of trust models belong to this type such as [39, 49, 1, 18, 62].

The classification proposed by Esfandiari and Chandrasekharan is a little bit bias, since there are too many mathematical trust models and too few cognitive trust models

2.1.2 A classification method by Ramchurn, Huynh, and Jennings

Ramchurn *et al.* proposed a classification for trust models in their paper 'Trust in Multi-agent Systems' [28]. This classification is shown in Figure 2.1.

In their paper, Ramchurn *et al.* first divided the trust models in MASs into two big categories which are *individual-level trust models* and *system-level trust models*.

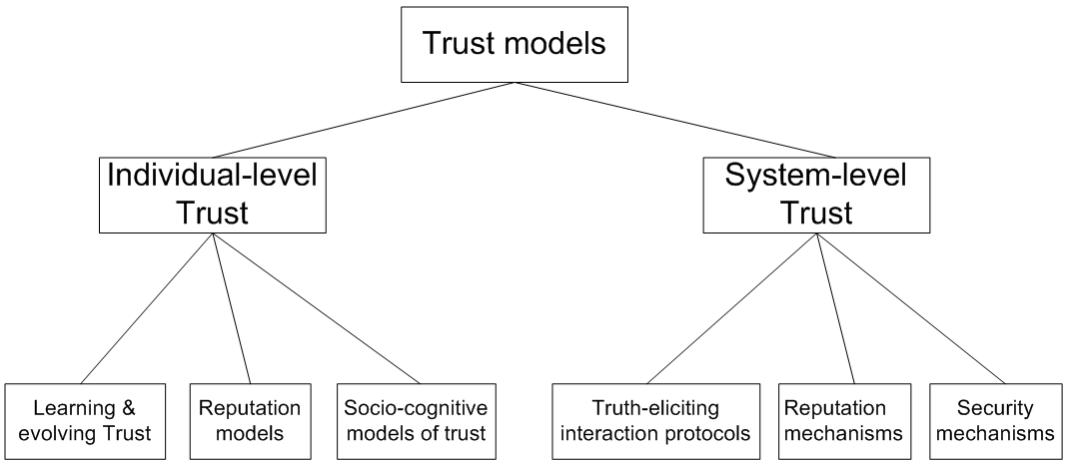


Figure 2.1: The Classification of Current Trust Models Proposed by Ramchurn *et al.*

- **Individual-level trust models**

  The objective of individual-level trust models is to give an individual agent the ability to make decisions on whether a potential communicating partner is trustworthy or not.

- **System-level trust models**

  The aim of the system-level trust models is to force all of agents in the system to conduct the trust behaviours by establishing trust rules.

1. Individual-level trust models

Individual-level trust models can be further classified into three subcategories.

- **Learning and evolving trust models**

  In this kind of trust models, an agent uses the direct experience to learn the behaviours of its partners through many times of interactions and develops corresponding strategies to interact with its partners according to the behaviours of partners.

- **Reputation trust models**

  In this kind of trust models, an agent retrieves the reputation ratings of a potential partner, which was rated by those agents who had interaction with the potential partner before. Then, the retrieved reputation ratings are aggregated to create a reputation trust value for the potential partner.

- **Socio-cognitive trust models**

  In this kind of trust models, an agent gets the trust information of a potential partner by analyzing the subjective perceptions which include assessment of the environment and characteristics of the potential partner.

2. System-level trust models

The system-level trust models can be also further classified into the following three subcategories

- **Truth-eliciting interaction protocols**

  Truth-eliciting interaction protocols establish a series of rules which can reduce the payoffs of the lying agents and force the agents to conduct trusted behaviours in interactions.

- **Reputation mechanisms**

  Similar to reputation trust models in individual-level trust models, the reputation mechanisms also collect the reputations of an agent from the system and aggregate the reputation to create trust values for corresponding agents. However, instead of collecting the reputation for a single agent, the mechanism conducts this task for all of the agents in a MAS.

- **Security mechanisms**

  Security mechanisms transfer the trust concept to the domain of the network security which includes identity recognition, assess permissions, content integrity and content privacy.

The above classification provides a general way and a good inspiration to classify current trust models. However, this classification is too complex and the subcategories in two trust domains have some repetition. For example, the processes in both of the reputation trust models and the reputation mechanisms are very similar, which are retrieving and aggregating the ratings for a potential partner and creating the trust value of the potential partner. The only difference between reputation trust models and reputation mechanisms is about whether an individual agent or the system finishes the task.

2.1.3 A classification method by Sabater and Sierra

Sabater and Sierra proposed three different classification methods for trust models from three different aspects in 2003 [47]. The three aspects are explained and analysed as follows.

Aspect 1: Visibility types

This kind of classification is based on in what form the trust information stored in a system. Based on this perspective, trust models can be classified into *global-type trust models* and *private-type trust models*.

- **Global-type trust models**

  This kind of trust models summarise reputation information for each agent of a system and make the trust information visible to all of the agents in the system. The trust value of an agent is updated dynamically after the agent finishing an interaction. A classical example of this type of trust models is the eBay typed trust models [17, 3, 42] widely used in online transaction or auction systems.

- **Private-type trust models**

  This kind of trust models always save the reputation information in individual agents. Therefore, if a service consumer wants to know the trust information of a potential service provider, the consumer needs to search the information from those agents, who had interaction with the potential service provider before for the trust information of the potential provider. The famous models are the CR model [28] and the trust model proposed by Rubiera *et al.* [45].

Aspect 2: Model's granularity

This kind of classification is based on whether a trust model considers the context of the trust value. Based on this consideration, the trust models can be classified into *single-context trust models* and *full-context trust models*.

- **Single-context trust models**

  This type of trust models do not consider the context for a trust value and hold the opinion that the trust value of an agent can reflect the performance of the agent in all kinds of situations. Therefore, in single-context trust models, an agent only has one trust value to represent its performance in history. The

typical single-context trust models are the trust model proposed by Carter *et al.* [11] and the trust model proposed by Yu and Singh [62].

- **Full-context trust models**

  Full-context trust models maintain trust information of an agent in the format of several trust values and their corresponding contexts. By considering the context of a trust value, the trust prediction for a potential partner is more accurate than that of in a single-context trust model. The trust model proposed by Esfandiary and Chandrasekharan [18] is an example of this type of models.

Aspect 3: Information source

This kind of classification depends on information source that a trust model uses for trust evaluation. Based on different trust information sources using in the trust models, the trust models can be classified into three categories, which are 1) *experience-based trust models*, 2) *reputation-based trust models* and 3) *sociological trust models*.

- **Experience-based trust models**

  In this kind of trust models, an agent uses its direct experience to evaluate the trust value for a potential partner. The direct experience of an agent to judge a potential partner can come from participating in the interaction with the potential partner or observing the interaction that the potential partner participating in. The trust model proposed by Marsh [39] is one of the trust models that only uses direct participating experience as the information source for trust evaluation. The Sen and Sajja proposed a trust model [51] which uses the observation experience as direct experience of an agent having on another agent.

- **Reputation-based trust models**

  This kind of trust models use witness information from the third parties to evaluate the trust for a potential partner. The eBay trust model [17, 3, 42] and the CR model [28] belong to this kind of turst models.

- **Sociological trust models**

  This kind of trust models evaluate the trust value of an agent based on the knowledge of social relations of this agent with other agents and also the role of the agent in the system. Normally, this kind of models are used to simulate human behaviours in a social environment in the real world. If an agent has

good relations with a number of other agents and plays an important role in the system, it is more likely that the agent is trustable and vice versa. There are only a few trust models use this kind of knowledge as the trust information, because it is a hard job to get all of the social and relationship knowledge from open environment in current applications. A famous model using sociological knowledge as trust information is proposed by Scott [50].

2.1.4 A new classification method proposed by this study

A service-oriented environment is an open and dynamic environment. Many new trust models have been developed in order to face new challenges. Many of these models use mixed techniques and the above three classifications have limitations to classify current trust models. Control mechanism is an important factor to distinguish current trust models. In this subsection, a new classification for current trust models is proposed based on the view of control mechanism used in trust models as shown in Figure 2.2

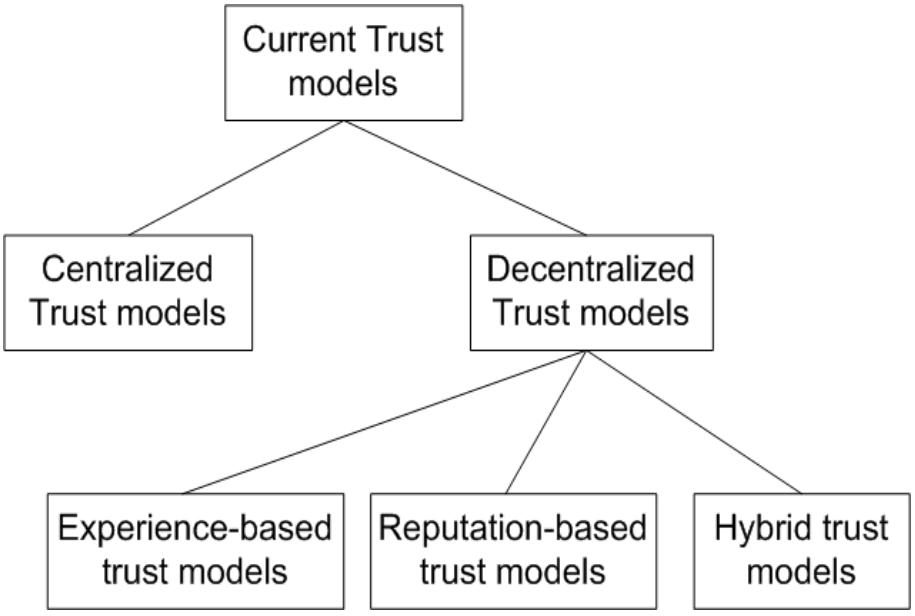


Figure 2.2: The New Classification of Current Trust Models

Based on control mechanisms, trust models can be classified into centralised trust models and decentralised trust models. In current literatures, there are more decentralised trust models than centralised models. We use the classification method from Sabater and Sierra (refer to subsection 2.1.3) to further classify the decentralised models into three subcategories, which are experience-based trust models, reputation-based trust models and hybrid trust models.

The advantages of the proposed classification can be outlined as follows.

- **Balance**

  Although there are less centralised trust models than decentralised models, the centralised trust models still play important roles in real world applications. For example, one of famous centralised trust models is eBay trust model which is widely used in many online transition and auction systems.

- **Clarity**

  The control mechanism in a trust model is a clear mark for classifying current trust models, since we can easily identify what kind of the system control mechanism a trust model uses.

- **Relevance to this research topic**

  In this thesis, we mainly focus on trust models for service provider selection in service-oriented MASs. Because of the characteristics of the MASs and service-oriented environments, it is hard for a centralised controller to be employed in service-oriented MASs. Based on this classification, we pay much attention on the deep investigation in decentralised trust models and focus on the detail reviews of important models in next section, which are close to this research.

2.2 Representative Trust Models

In this section, several important trust models are reviewed in detail based on the new classification proposed in Subsection 2.1.4.

2.2.1 Centralised trust models

A centralised trust models generally has a centralised controller to control interactions among agents and to store the trust information of the system. Since service-oriented MASs are decentralised in nature, the centralised control mechanism cannot fit the characteristics of MASs and service-oriented environments in most current applications. Most of the centralised trust models [20, 64, 63, 37, 22, 17] were proposed in the early stage of the trust model development, which ever played or still play an important role in some real world applications or provide basic foundations for the development of new trust models.

The eBay trust model is an example of these models, which has been widely used in online electronic commerce systems including eBay [17], Amazon [3], OnSale [42] and so on. The major features of this type of trust models are simple and easy to use. The eBay trust model only uses historical experience from interaction partners to deduce the trust value of a user. In eBay trust model, after a transition, all of the users (consumers and providers) participating in the transition need to report their feedbacks about partner users in the format of a single value. Then, a centralised unit can dynamically update the trust values for the corresponding users based on the feedbacks. Next time, a new user can make decision about whether a partner can be trusted to do business with based on the updated rating retrieved from the system for this partner. For example, after a transition, the users participating the transition need to rate trust values for each other within the range [-1,1], where -1 indicates a fully negative trust while 1 represents a totally positive trust on a participant, respectively. Then, the feedbacks are sent to the central trust management unit. These feedbacks are summed up with the historical trust values of the corresponding users in a time period (mostly six months). After that, the newest trust values for corresponding users can be obtained and stored by the centralised management unit. Thus, the trust value of a user in eBay trust model can accurately reflect the average performance of a user in a historical period. However, the limitations of eBay trust model can be analysed as follows.

- The trust value of a user in eBay trust model is represented by a single value, which can only indicate the trustworthiness of a user. From this value, a user cannot discover any other useful information (i.e. context, situations). Therefore, it is relatively hard for a user to accurately predict the future behaviours of the host of the trust value.

- The newest trust value of a user is obtained by averagely summing up the trust values of the user in a time period. Therefore, the updated trust value can only reflect the general performance of a user in a time period instead of the newest or recent performance. This mechanism can cause some kinds of malicious behaviours. For example, a user may offer good products for a period of time. However, in recent transitions, the quality of products offered by the user becomes bad. Although the user gets bad ratings for its bad quality products, with the accumulation of its historical good behaviours, another user cannot find great changes in trust value of the user until it offers bad products for a long period of time.

- The eBay trust model does not consider noisy ratings for users. For example, although a user can get a very good product from its transition partner, the user can deliberately give its transition partner a low rating without any punishment on the user's malicious behaviours.

In summary, the eBay trust model gives us a basic and simple idea on how to evaluate the trust for a user. However, it is hard for eBay trust model to be widely employed in service-oriented MASs.

Another important centralised trust model is the SPORAS trust model proposed by Zacharia and Maes [64]. Being different with the eBay trust model, SPORAS introduced several new mechanisms to overcome the limitations of eBay trust model. For example, SPORAS employed a learning function for updating trust values of agents. Therefore, the trust value of an agent can realistically reflect the recent performance of the agent. SPORAS also introduced the following mechanisms to ensure the accuracy of trust value of agents.

1. New agents in SPORAS can only start with a minimum trust value.

2. The trust value of a user who already had transitions with other agents never falls below the trust value of a new user.

3. After a transaction, the trust values of the involved agents need to be updated according to the feedbacks offered by their partners.

4. Agents with very high trust values can only have very small rating changes after updating.

5. Trust values in former periods need to be discounted according to time, by which the system can ensure that the trust value can reflect the recent performance of the corresponding agent.

From above mechanisms, we can see that the first and second mechanisms can avoid an agent with a bad reputation leaving the system to refresh its bad reputation with a new reputation and identity. The fifth mechanism considers the recency factor of the trust value of an agent. However, although the above mechanisms can overcome some limitations of eBay trust model, the SPORAS has its own problems. For, example, it does not consider the relationships between agents, which may lead to inaccurate ratings. For example, if the agents involved in a transition have collaboration relationships, they may give higher ratings than real values for each other and if the agents

involved in a transition have competition relationships, they may give lower ratings than real values for their competitors.

2.2.2 Decentralised trust models

Being different with centralised trust models, a decentralised trust model does not have a centralised controller to control all of agents behaviours and to manage trust information [39, 27, 35, 54]. From this consideration, decentralised trust models are more suitable and encouraged to be applied in service-oriented MASs than centralised trust models.

Experience-based trust models

In experience-based trust models, an agent evaluates the trust value for a potential partner based on its former direct interactions with the partner or its observation experience of other agents interaction with the potential partner. The advantages of experience-based trust models are that the trust information is reliable and easy to be obtained, since the experience can directly come from the agent itself. Mostly, the reliable trust information from direct experience needs a number of interactions between two agents. However, the scale of most service-oriented MASs is big and the members of these systems are dynamic. Therefore, it is hard for an agent to have direct interaction or observation experience with most of agents in a system. Moreover, even if an agent wants to use a service offered by a familiar agent, it is possible that the familiar agent might be not in the system at that time. Another important problem in experience-based trust models is that if a system allows the interaction between two agents to be observed by other agents, the system should offer some security mechanisms to protect the privacy of interacting agents.

Currently, there are a few trust models that only use direct experience as the information source of the trust. In 1994, an important experience-based trust model was proposed by Marsh [39]. In this model, Marsh classified the trust of an agent into three aspects which are basic trust, general trust and situational trust. Although the trust model proposed by Marsh is an early trust model, it made a significant contribution to the field in terms of its consideration of the effect of other agents' opinion, utility, situation and environment on the trust value of an agent. However, the trust models which only use the direct interaction experience as the trust information source can limit its application in service-oriented MASs. Due to the scale of most

MASs and the dynamic nature of service-oriented environments, an agent in a service-oriented MAS cannot have direct interaction experience with all of other agents. In such a kind of systems, the Marsh trust model cannot work very well.

Sen and Sajja proposed a trust model [51] based on probabilistic calculations of trust values given by a number of agents including both providers and consumers [51]. In their model, surrounding agents of an interaction pair can observe the interaction between the service consumers and providers. Then, the observed service provider's trust information from both the participants and the observing agents is updated using a reinforcement learning rules. When a new consumer needs the reputation of the corresponding service provider, the surrounding agents and the former interaction participants can give the latest reputation of the potential service provider. Their model introduced another example for using direct experience, i.e. the observation experience. The observation mechanism can greatly increase the trust knowledge of an agent on other agents. In Sen and Sajja's model, the interacting agents can also be observed by surrounding agents, which can lead to some security problems in interaction.

Currently, few trust models that only use direct experience as the trust information source. But the direct experience still plays an important role in trust evaluation, since the direct experience is the most reliable trust information source and is also easy to be gained. Many trust models use both the direct experience and the witness information to evaluate the trust values for potential partners.

Reputation-based trust models

In reputation-based trust models, an agent evaluates the trust value for a potential partner based on the witness information of other agents (referees), which may directly or indirectly have interaction with the potential partner before. In some situations, reputation is not a very reliable information, since we need to consider the relationships between the potential partner and referees. If the relationship between a referee and the potential partner is collaboration, the referee may give higher reputation value for the potential partner than the real trust value. In contrast to collaboration, if the relationship between the referee and potential partner is competition, the referee may give a relatively lower reputation value for the potential partner. By this consideration, the reputation trust is more complex than the direct experience.

The most famous reputation-based model for trust calculation in recent years is the Certified Reputation (CR) model proposed by Huynh *et al.* [28]. In the CR model, an agent's reputation is derived from the references of the third parties, which

had previous interaction experiences with the agent (provider) before. A provider can collect and present such references to service consumers in order to be trusted by them. Since the CR model allows consumers to evaluate trust values of providers themselves without using a central controller, it can be adapted in a wide range of open and dynamic environments such as service-oriented environments. However, there are still some limitations in the CR model. Firstly, 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, it is hard or even impossible to use a single value to represent complex contexts related to a service [52]. 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 based on multiple providers.

The trust model proposed by Sen and Sajja [51] can also be called as a reputation based trust model, since the main information source of the Sen and Sajja trust model is the reputation. When an interaction between two agents happens, neighborhoods of interacting agents can observe the interaction. After that, the neighborhoods of the target agent can offer the witness trust information for a potential provider to a service consumer. The service consumer summarises all of the reputation from other agents to select the best potential service provider. This is a typical way of trust model using witness reputation as the trust information source and the observation mechanism can only increase the number of witnesses for potential service provider.

Hybrid trust models

Hybrid trust models use both direct experience and reputation as the trust information source. Currently, most of trust models use both of direct experience and witness information as the information source.

J. Sabater and C. Sierra proposed a famous model, called REGRET, in 2001 [46]. In principle, the REGRET evaluates the trust value of a potential provider from three dimensions which are the individual dimension, the social dimension and the ontological dimension. The individual dimension is the direct experinece of the service provider offered by a service consumer who had an interaction with the provider before. The social dimension is the reputation of a group which a service provider belongs to. The ontological dimension represents the reputations of different aspects of the services

offered by the provider. Based on above comprehensive considerations, REGRET trust model can have a accurate trust evaluation for a potential provider.

Another important hybrid trust model, called FIRE, was proposed by Huynh *et al.* [27]. In the FIRE trust model, a trust and reputation of an agent is composed of four parts which are interaction trust, role-based trust, witness reputation, and certified reputation. Each component was introduced with an example that an agent a wants to evaluate the trust value for the potential partner b.

- The interaction trust of agent b is from the former direct interaction between agent a and agent b.
- The role-based trust of agent b is gained from the agents that have relationship with agent b. The relationship can includes all kinds of relationships such as collaboration, competition and so on.
- The witness reputation of agent b is built on the opinion of other agents (witnesses) that have direct interaction with agent b.
- The certified reputation of agent b is provided by agent b itself which is ranked by former interaction partners of agent b and stored by agent b itself.

2.3 Summary

In this chapter, three existing classification methods for trust models were introduced first. Then, a new classification method was proposed to classify current trust models into centralised trust models and decentralised trust models based on control mechanisms and then a number of important trust models were reviewed in detail.

Chapter 3

A Priority-Based Trust Model

In this chapter, a Priority-Based Trust (PBTrust) model is presented for single service provider selection in order to overcome several limitations of the CR model and to solve several research issues which were identified in Section 1.2 of Chapter 1. This chapter is arranged as follows. Section 3.1 is the problem description and definitions. In Section 3.2, the basic principle of the PBTrust model is introduced and the design of each module is described in detail. Section 3.3 demonstrates two experiments and provides the experimental analysis and discussions. Section 3.4 is the summary of the chapter.

3.1 Problem Description and Definitions

In general, a service can be described by a number of attributes such as price, time, quality, etc. The priority is a value between 0 and 1, which represents how much attention a service consumer paid on an attribute. For different requests, the priorities on different attributes of the same service can be different. In order to deal with the relationships between attributes and their corresponding priorities, we make a service description in a formal way.

Suppose there are n attributes used to describe a requested service and each attribute is in a requested priority as the condition to complete the service. The service can be represented by n attributes and their corresponding priorities, respectively.

Definition 3.1: A *service description* (*SDes*) is the formal description of a service. *SDes* is defined in the following matrix format.

$$SDes = \begin{pmatrix} A_1 & A_2 & A_3 & ... & A_n \\ W_1 & W_2 & W_3 & ... & W_n \end{pmatrix} \tag{3.1}$$

where A_i indicates the i^{th} attribute; W_i is the priority value of A_i and $\sum_{i=1}^{n} W_i = 1$.

Definition 3.2: (*Ratings*) is the rating scores of a service provider's performance on a service given by a referee. *Ratings* is defined as a n-tuple, $Ratings = <R_1, R_2, ..., R_n>$, where R_i indicates the rating value of the i^{th} attribute of the service (recall Definition 3.1). Here the range of R_i is [0, 100], where 0 and 100 represent the worst and best performance for i_{th} attribute.

In the CR model, the references of a provider can only reflect its good performances. So it is hard for a consumer to have a general view about whether the provider has a consistent performance on the requested service. In order to solve this problem, the concept of *service experience* of a provider on a certain service is introduced in this model and defined below.

Definition 3.3: The *Service Experience* (*Exp*) of a provider on a service is defined as a 2-tuple, $Exp = <SRate, SNum>$, where $SRate$ indicates the success rate of the provider on this service and $SNum$ indicates the total number of success times on the same service.

Definition 3.4: A *Service Request* (*SReq*) is defined as a 4-tuple, $SReq = <CID, SDes', RN, SThreshold>$, where CID is the service consumer's ID, $SDes'$ indicates the service, which is a 2 by n matrix representing the requested attributes and their priorities (recall Definition 3.1), RN $(0 < RN)$ is the number of references that CID requests, and $SThreshold$ is the threshold of the success rate for a provider to qualify for providing the service.

Definition 3.5: A $Reference$ (*Ref*) is defined as a 4-tuple, $Ref = <RefID, SDes, Ratings, T>$, where $RefID$ is the ID of the referee, $SDes$ (recall Definition 3.1) is the service description conducted by the provider for the referee, *Ratings* indicates the performance for the service, given by $RefID$ for each attribute of the service (recall Definition 3.2), and T is the time in the completion of the service.

Definition 3.6: A *Service Reply* (*SRep*) is defined as a 3-tuple, $SRep = <SPID, RefSet, Exp>$, where $SPID$ is the ID of the service provider, $RefSet$ is the set of references including several previous best references provided by different referees (the size of $RefSet$ can be determined by consumers), and Exp is the experience indicator (recall Definition 3.3) indicating the provider's general performance on this service.

3.2 The Basic Principle of the PBTrust Model

This section gives the detail introduction of four modules in the PBTrust model.

3.2.1 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-marketplace requests a service described by 3 attributes, i.e. cost, speed, and quality with corresponding priorities for each attribute as (0.3, 0.5, 0.2), respectively. C requests 2 references and the requested success rate for a potential provider on the service in history should be at lest 70%. Based on this service request, the Request Module will generate a service description by using the format of Definition 3.1.

$$SDes' = \begin{pmatrix} Cost & Speed & Quality \\ 0.3 & 0.5 & 0.2 \end{pmatrix}$$

Then, a Service Request $SReq$ will be produced based on the service description and requirements of the Consumer C in the format defined by Definition 3.4.

$$SReq{=}< C,\ SDes',\ 2,\ 0.7>$$

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

3.2.2 The reply module

When a potential provider P can offer the service based on the requirement from Consumer C, P will provide the following information: the provider ID, two reference reports, as well as service experience on the service before including success rate and total success times.

Suppose that P received 3 reference reports for its previous performances on the same service from different consumers representing by a reference set $\{Ref_1, Ref_2, Ref_3\}$, and each element in the set is in the format defined by Definition 3.5. P will pick up two best reference reports to represent its previous performance on the service, say Ref_2 and Ref_3.

Suppose that the success rate of P on the service is 70% and total success times to complete the service is 35.

The reply information from P responding to the request from C is as follows (recall Definition 3.6).

$$SRep = <\text{P}, \{Ref_2, Ref_3\}, (0.7, 35) >$$

If more than one service providers have the requested service and also have the intention to provide the service, this module will generate more than one replies.

3.2.3 The priority-based trust calculation module

This module is the core of the PBTrust model. The main purpose of this module is to calculate the trust values of potential providers based on reference reports from third parties, service experience of providers, the time weights of references, and the similarities between the description of the requested service and the one from reference reports in terms of different priorities on same attributes. These trust values will help a consumer to select the most trustable provider to complete the service. The final trust value for each potential provider is produced from several calculation results in four perspectives, which are the provider's experience on the service, the similarity of priorities distributions on attributes between the referenced service and the requested service, the suitability of the potential provider for the requested service, and the time effectiveness of ratings from third parties.

The Priority-Based Trust Calculation Module is used to produce the reputation values for potential service providers from four perspectives, which are the provider's experience on the service, the similarity of priorities distributions on attributes between the referenced service and the requested service, the suitability of the potential provider for the requested service and the time effectiveness of ratings from third parties. These perspectives have the contributions to the final reputation value from different views and are defined by separate formulas. This subsection gives the detail introduction of this module.

Design and principle of priority-based trust calculation

In order to produce reliable and robust trust values for potential service providers, we develop a priority-based trust calculation mechanism based on the following considerations:

1. The third party reference is used to derive the reputation of providers.
2. The term 'suitability' is introduced to predict the potential performance of a provider for requested service based on the information from a third party reference about the provider's previous performance and the information of new priority requested by the consumer.

3. The similarity measurement between the priority distribution on attributes of the service from a reference report and the priority distribution on attributes of the service requested from a consumer is also considered.

4. The Timestamp of the reference report is taken into account to reduce the contribution of out-of-date references from third parties.

5. The service experience is also used for the trust calculation.

6. The influence of all ratings from different referees are also considered.

7. Finally, the trust value of a potential provider is calculated based on the above factors.

Based the above considerations, we develop the following formula to calculate the trust values in the PBTrust model.

$$Trust = EW \times \frac{\sum_{k=1}^{RN} Sim_k \times SInd_k \times TStamp_k}{RN} \tag{3.2}$$

In Formula 3.2, EW is the experience weight of the provider, Sim_k refers to the similarity of priority distribution of attributes in the service from the k^{th} reference report of requested service, $SInd_k$ is the suitability indicator based on the information of the k^{th} reference's ratings and the priorities in the requested services, $TStamp_k$ represents timestamp for the k^{th} reference, and RN is the number of references requested by the consumer and $RN > 0$.

The detail design for calculation of items EW, Sim_k, $SInd_k$, and $TStamp_k$ in Formula 3.2 are introduced in this subsection, respectively.

Experience weight calculation

Experience weight EW represents the general performance of a service provider on this service. The higher the experience weight, the more contribution to the trust calculation. EW is constructed by two factors, $SRate$ and Fsn. $SRate$ represents the successful rate (recall Definition 3.3), while Fsn is the contribution to the EW from the total number of successful performance of the provider in history. EW is defined by the following formula.

$$EW = SRate \times Fsn \tag{3.3}$$

The $SRate$ is the successful rate of provider on the service (recall Definition 3.3). The Fsn is defined by Formula 3.4.

$$Fsn = 1 - e^{-\frac{SNum}{\lambda}} \tag{3.4}$$

The reason for calculating Fsn by using an exponential increasing function is that the high success number on the service means the rich experience. When the success number achieves a very large value, the increase of Fsn becomes slowly. Here λ is a coefficient to control the speed changing in the curve which can be adjusted by users based on different application domains.

Similarity calculation

The similarity on priorities between the i^{th} reference and requested service can be calculated by using Formula 3.5.

$$Sim_i = \frac{\sum_{k=1}^{n} CW_k \times RW_k}{\sqrt{(\sum_{k=1}^{n}(CW_k)^2) \times (\sum_{k=1}^{n}(RW_k)^2)}} \tag{3.5}$$

where CW_k and RW_k represent the weight of k^{th} attributes for the requested service by the consumer and reference service by provider, respectively.

To what extent, can the reference reflect the potential performance on the requested service? To answer this question, we should consider the similarity of priorities between the requested service and the referenced service. In the PBTrust model, we use a matrix to describe a service (recall Definition 3.1). Since attributes in both requested service and a referenced service are in the same order, we can omit attributes during similarity calculation. Now, a description matrix becomes a vector which includes priority values for corresponding attributes. We can use dot product of two vectors. If angle between two vectors' direction are named θ, the dot product of two vectors indicates the cosine value of angle θ in mathematics.

$SReq$ in Figure 3.1 represents the priorities vector of the request, and Ref indicates the priorities vector of service reference. θ is the angle between vector $SReq$ and Ref. Since all priorities of attributes are positive numbers and the sum of them is 1, so the range of angle θ is [0°, 90°], the range of $\cos\theta$ is [0,1]. If $\theta = 0°$ and $\cos\theta = 1$ means there is no difference between two vectors' direction and the attributes priorities of requested service and referenced service are the same, so the provider's performance in reference can completely reflect the requested service. Oppositely, if $\theta = 90°$ and $\cos\theta = 0$ means there is the biggest difference between two vectors' direction and the

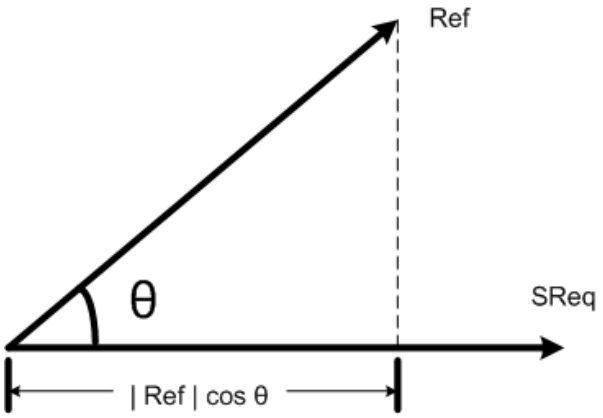


Figure 3.1: The Dot Product of Two Vectors

attributes priorities of requested service and referenced service are totally different, so the provider's performance in reference cannot reflect the requested service.

For example, suppose that there is a Service Request with 3 attributes, i.e., Cost, Speed and Quality; the priority of each attribute is specified in the vector $SReq =< 0.3, 0.5, 0.2 >$. If we have two potential providers, i.e. P1 and P2, for this request and the references of the two providers are $Ref_{P1} =< 0.3, 0.5, 0.2 >$ and $Ref_{P2} =< 0, 0, 1 >$, respectively. By using Formula 3.5, it can be found that $Sim_{P1} = 1$ and $Sim_{P2} = 0.32$. That means that the reference provided by P1 has very similar priority distribution with the request service, and the reference provided by P2 has very different priority distribution with the request service. Therefore, different weights will be assigned to the two references when calculating the trust values of P1 and P2.

Suitability indicator calculation

The purpose of suitability indicator is to predict the potential performance of a provider on the requested service by using two pieces of information, reference ratings and the priorities of attributes in the requested service. The suitability indicator of the i^{th} reference can be calculated by the following formula.

$$SInd_i = \sum_{k=1}^{n} R_k \times CW_k \tag{3.6}$$

where CW_k represents the weight of k^{th} attributes for the requested service by the consumer, and R_k is rating value for the k^{th} attribute given by the i^{th} referee.

Timestamp calculation

The purpose of using the timestamp to evaluate the influence of references on the trust value is to eliminate or reduce the effect of out-of-date ratings depending on the value of T in a reference, (recall Definition 3.5). The method for the timestamp calculation is borrowed from the same concept used in the CR model [28]. Timestamp for the i^{th} reference report is calculated by the following formula:

$$TStamp_i = e^{-\frac{\Delta t(i)}{\lambda}} \tag{3.7}$$

where $\Delta t(i)$ means the time difference between the time when the i^{th} reference was generated and the current time, and λ is an coefficient to control the speed changing in the time curve depending on application domains.

3.2.4 The evaluation module

This module includes two components. One is to generate a reference report from a consumer for a provider based on the performance of a completed service, and the other is to update the record of service experience of a provider when a new reference is available for the provider.

Reference report generation

We use the same example as in Request Module and Reply Module to demonstrate how to generate a reference report in this module. After completing the requested service, Consumer C evaluates the performance of Provider P on the service. The evaluation result is represented in a reference report, (recall Definition 3.5) shown as follows.

$$Ref =< C, SDes, < 60, 40, 90 >, 12/7/2008 >$$

The above reference report shows the evaluation result from Consumer C on the service $SDes$, completed on 12 July 2008. From Consumer C's rating, we can see that C was satisfied with the cost of the service (the first attribute of $SDes$), not satisfied with the speed of the service (i.e., the second attribute of $SDes$), and very satisfied with the quality of the service (the third attribute of $SDes$).

Service experience updating

The service experience updating is based on consumer's judgement on a newly completed service from the provider. A judgement result can be either 'success' or 'fail'.

The service experience Exp includes two elements $SNum$ and $SRate$ (recall Definition 3.3). $SNum$ and $SRate$ can be updated by using Formulas 3.8 and 3.9.

$$SNum = \begin{cases} SNum' + 1 & judgement : success \\ SNum' & judgement : fail \end{cases} \tag{3.8}$$

$$SRate = \begin{cases} \frac{SNum'+1}{SNum'/SRate'+1} & judgement : success \\ \frac{SNum'}{SNum'/SRate'+1} & judgement : fail \end{cases} \tag{3.9}$$

where the $SNum'$ and $SRate'$ represent the total success times and the success rate before updating, respectively.

Suppose that Consumer C is satisfied with the service provided by Provider P, C will give the evaluation result, 'success' for P on this service. In this situation, Formulas 3.8 and 3.9 will be used to update the record of P's experience from (0.7, 35) to (0.706, 36).

Suppose that Consumer C is not satisfied with the service, C will give elevation result, 'fail' for P on this service. In this situation, Formulas 3.8 and 3.9 will be used to update the record of P's experience from (0.7, 35) to (0.686, 35).

By using this updating method, the PBTrust model can not only dynamically update records of service experience for all agents in open environments but also accumulate information to show general performance of each agent without a central control mechanism.

3.3 Experiments

In this research, we conducted two experiments, i.e., Experiment 1 and Experiment 2, to compare the performances of the PBTrust model and the CR model. Through these experiments, we want to claim that the PBTrust model can perform better than the CR model in provider selections. To compare the performance of the two trust models, an evaluation benchmark, i.e., the *Satisfaction Degree* (*SatDegree*), is defined to evaluate how a selected service satisfies the expectation of a consumer.

Definition 3.7: *Satisfaction Degree (SatDegree)* is the difference between a service selected by a trust model and the expected service of a consumer. Satisfaction Degree can be calculated by using Formula 3.10.

$$SatDegree = Sim \times \frac{\sum_{i=1}^{n} R_i}{n \times 100} \tag{3.10}$$

where Sim is the similarity of priority distribution between referenced service and requested service (recall Formula 3.5), R_i is the rating of the i^{th} attribute in the service and n is the number of attributes.

The performance difference between the PBTrust model and the CR model will be mainly impacted by three newly introduced items in the PBTrust model, i.e., the Similarity item (*Sim*), the Suitability Indicator item (*SInd*) the Experience Weight item. As the Timestamp (*TStamp*) item is directly borrowed from the CR model, it will not cause any performance different between the PBTrust model and the CR model. Therefore, we did not include the *TStamp* item in our experiments.

3.3.1 Experiment 1: evaluating the impacts of *Sim* and *SInd*

Experiment setting

In Experiment 1, three attributes, which are *cost*, *speed*, and *quality*, are considered for each service. Seventy service providers (P_1, P_2, ... P_{30}, ... ,P_{70}) with different priority and rating settings were employed. One Service Request with the following description was included.

$$SDes' = \begin{pmatrix} Cost & Speed & Quality \\ 0 & 0.2 & 0.8 \end{pmatrix}$$

The seventy service providers were classified into seven groups according to the similarity degrees of their reference priorities with the requested priorities. Each provider group (G_l) has a different *Average Similarity Degree* ($AveSim_l$) (which indicates the average similarity degree of its group members). $AveSim_l$ can be calculated as follow.

$$AveSim_l = \frac{\sum_{P_i \in G_l} Sim_i}{|G_l|} \tag{3.11}$$

where P_i is a provider in group G_l, Sim_i is the similarity between P_i and the requested service $SDes$, $|G_l|$ is the size of G_l. The average similarity degree of each group is shown as Table 3.3.1, and priority distributions and ratings of providers can be found in Table 3.3.1.

Group ID (l)	Average Similarity Degree ($AveSim_l$)
$Group1$	0.2
$Group2$	0.3
$Group3$	0.4
$Group4$	0.5
$Group5$	0.6
$Group6$	0.7
$Group7$	0.8

Table 3.1: Average Similarity Degrees of Each Provider Group

As Experiment 1 is focused on evaluating the impacts from *Sim* and *SInd*, we exclude *EW* and *TStamp* in this experiment assigning the same values for all providers: EW =<100%, 100>, $TStamp$ = "30/01/2011". In addition, we set the reference number (RN) of the Service Request to 1, which means only one reference report will be collected from each provider.

Trust value transfer function

In order to compare the two trust models, a standard service description format is required. However, there are several differences between service descriptions of the two models which are listed in Table 3.3.1.

	The CR model	**The PBTrust model**
Similarity Item & Priority	Similarity Item is not considered. Similarity factors and priority distributions will not affect trust calculations in the CR model.	Similarity Item is considered. Trust calculations will affected by similarity factors and priority distributions.
Service Description	A service is represented as a single item without consideration of service attributes.	A service is described by a matrix with multiple attributes and their priority distributions.
Description of providers' performances	Providers' performances are represented by single rating values.	Providers' performances are represented as vectors. Each element in a vector represents the rating value of a particular service attribute.

Table 3.3: Differences of Service and Trust Description in the CR Model and the PBTrust Model

To standardise the service descriptions in the two models, we define a transfer function $f(M, V)$ to convert the trust values from the CR model to the PBTrust

Provider ID (i)	Group ID (l)	Priority distribution on (Cost, Speed, Quality)	Ratings
P_1	1	(0.9, 0.1, 0)	$< 20, 100, 50 >$
P_2	1	(0.8, 0.2, 0)	$< 100, 20, 40 >$
P_3	1	(0.7, 0.3, 0)	$< 100, 50, 0 >$
P_4	1	(0.9, 0, 0.1)	$< 100, 50, 20 >$
P_5	1	(0.8, 0.1, 0.1)	$< 20, 100, 40 >$
P_6	1	(0.5, 0.5, 0)	$< 50, 100, 0 >$
P_7	1	(0.7, 0.2, 0.1)	$< 60, 10, 100 >$
P_8	1	(0.4, 0.6, 0)	$< 60, 60, 60 >$
P_9	1	(0.8, 0, 0.2)	$< 40, 20, 100 >$
P_{10}	1	(0.3, 0.3, 0.4)	$< 0, 50, 100 >$
P_{11}	2	(0.7, 0.3, 0)	$< 20, 100, 50 >$
P_{12}	2	(0.9, 0, 0.1)	$< 100, 20, 40 >$
P_{13}	2	(0.6, 0, 0.4)	$< 100, 50, 0 >$
P_{14}	2	(0.7, 0.2, 0.1)	$< 100, 50, 20 >$
P_{15}	2	(0.4, 0.6, 0)	$< 20, 100, 40 >$
P_{16}	2	(0.7, 0.1, 0.2)	$< 50, 100, 0 >$
P_{17}	2	(0.5, 0.4, 0.1)	$< 60, 60, 60 >$
P_{18}	2	(0.7, 0, 0.3)	$< 60, 10, 100 >$
P_{19}	2	(0.5, 0.3, 0.2)	$< 40, 20, 100 >$
P_{20}	2	(0, 0.5, 0.5)	$< 0, 50, 100 >$
P_{21}	3	(0.6, 0.4, 0)	$< 20, 100, 50 >$
P_{22}	3	(0.8, 0.1, 0.1)	$< 100, 20, 40 >$
P_{23}	3	(0.8, 0, 0.2)	$< 100, 50, 0 >$
P_{24}	3	(0.7, 0.1, 0.2)	$< 100, 50, 20 >$
P_{25}	3	(0.3, 0.6, 0.1)	$< 20, 100, 40 >$
P_{26}	3	(0.7, 0, 0.3)	$< 60, 60, 60 >$
P_{27}	3	(0.6, 0.1, 0.3)	$< 50, 100, 0 >$
P_{28}	3	(0.3, 0.5, 0.2)	$< 60, 10, 100 >$
P_{29}	3	(0.6, 0, 0.4)	$< 40, 20, 100 >$
P_{30}	3	(0.2, 0.3, 0.5)	$< 0, 50, 100 >$
P_{31}	4	(0.7, 0.3, 0)	$< 20, 100, 50 >$
P_{32}	4	(0.9, 0, 0.1)	$< 100, 20, 40 >$
P_{33}	4	(0.4, 0.6, 0)	$< 100, 50, 0 >$
P_{34}	4	(0.1, 0.7, 0.2)	$< 60, 60, 60 >$
P_{35}	4	(0.3, 0.5, 0.2)	$< 100, 50, 20 >$
P_{36}	4	(0.5, 0.2, 0.3)	$< 20, 100, 40 >$
P_{37}	4	(0, 0.7, 0.3)	$< 50, 100, 0 >$
P_{38}	4	(0.3, 0.3, 0.4)	$< 60, 10, 100 >$
P_{39}	4	(0.4, 0, 0.6)	$< 40, 20, 100 >$
P_{40}	4	(0.2, 0.3, 0.5)	$< 0, 50, 100 >$
P_{41}	5	(0.3, 0.7, 0)	$< 20, 100, 50 >$
P_{42}	5	(0.2, 0.8, 0)	$< 100, 20, 40 >$
P_{43}	5	(0.7, 0, 0.3)	$< 100, 50, 0 >$
P_{44}	5	(0.5, 0.2, 0.3)	$< 100, 50, 20 >$
P_{45}	5	(0, 0.7, 0.3)	$< 60, 60, 60 >$
P_{46}	5	(0.5, 0, 0.5)	$< 20, 100, 40 >$
P_{47}	5	(0.4, 0.2, 0.4)	$< 50, 100, 0 >$
P_{48}	5	(0.3, 0.3, 0.4)	$< 60, 10, 100 >$
P_{49}	5	(0.4, 0, 0.6)	$< 40, 20, 100 >$
P_{50}	5	(0, 0.1, 0.9)	$< 0, 50, 100 >$
P_{51}	6	(0.5, 0.3, 0.2)	$< 20, 100, 50 >$
P_{52}	6	(0.1, 0.7, 0.2)	$< 100, 20, 40 >$
P_{53}	6	(0.3, 0.5, 0.2)	$< 100, 50, 0 >$
P_{54}	6	(0.5, 0.2, 0.3)	$< 100, 50, 20 >$
P_{55}	6	(0, 0.7, 0.3)	$< 20, 100, 60 >$
P_{56}	6	(0.5, 0, 0.5)	$< 60, 60, 60 >$
P_{57}	6	(0.1, 0.4, 0.5)	$< 50, 100, 0 >$
P_{58}	6	(0.2, 0.3, 0.5)	$< 60, 10, 100 >$
P_{59}	6	(0.2, 0, 0.8)	$< 40, 20, 100 >$
P_{60}	6	(0, 0.2, 0.8)	$< 0, 50, 100 >$
P_{61}	7	(0.5, 0.4, 0.1)	$< 20, 100, 50 >$
P_{62}	7	(0.5, 0, 0.5)	$< 100, 20, 40 >$
P_{63}	7	(0.4, 0.2, 0.4)	$< 100, 50, 0 >$
P_{64}	7	(0.3, 0.3, 0.4)	$< 100, 50, 20 >$
P_{65}	7	(0.4, 0, 0.6)	$< 60, 60, 60 >$
P_{66}	7	(0.1, 0.4, 0.5)	$< 20, 100, 40 >$
P_{67}	7	(0.2, 0.3, 0.5)	$< 50, 100, 0 >$
P_{68}	7	(0.2, 0.1, 0.7)	$< 60, 10, 100 >$
P_{69}	7	(0.1, 0, 0.9)	$< 40, 20, 100 >$
P_{70}	7	(0, 0.2, 0.8)	$< 0, 50, 100 >$

Table 3.2: Service Providers' References

model. In the PBTrust model, a service S includes a number of attributes, i.e. $S = (A_1, A_2, ..., A_n)$. In order to match the service presentation in the CR model, we can treat attributes of S as sub-services so that S can be represented as $S = (S_1, S_2, ..., S_n)$. Then, the CR model can be used to calculate the trust value of sub-services, i.e., S_1, S_2 ... S_n. The overall trust value of S in the CR model can be obtained by calculating the weighted average values of all sub-services. Since the CR model does not consider priorities, we assume that all attributes have the equal priority values and the sum of these priority values is 1. Based on the above considerations, the trust transfer function can be defined as Formula 3.12.

$$f(M, V) = \frac{\sum_{i=1}^{n} R_i}{n} \tag{3.12}$$

where M is the service description matrix of a referenced service, V is the rating-vector of a service provider from the referee, and R_i is the rating of the i^{th} attribute and n is the number of attributes the service has.

In order to compare the performances of the CR model and the PBTrust model, we need to evaluate which service selected by the two models can satisfy the requirements from the consumer better. The priority distribution in a Service Request indicates the expecting priority distribution of the consumer. Although a Service Request does not indicate the expectation about the rating of each attribute, it can be assumed that a consumer always expects the highest rating (i.e., 100) on each attribute. Therefore, for this experiment, the expected ratings from the consumer on *Cost*, *Speed* and *Quality* are: (100, 100, 100).

Experimental results and analysis of experiment 1

In Experiment 1, services were selected from the seven group by using the two trust models. Then, the Satisfaction Degrees of the selected services are calculated and illustrated in Figure 3.2.

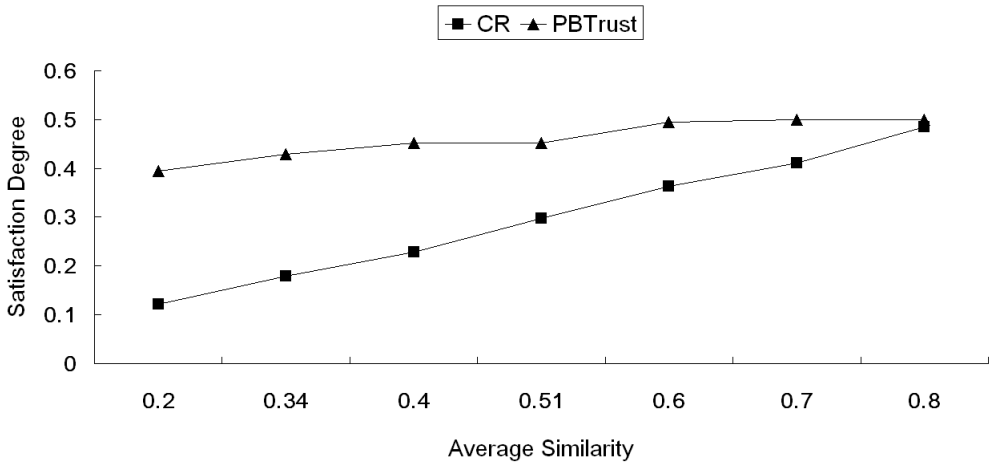


Figure 3.2: The Satisfaction Degrees of the Services Selected by the CR Model and the PBTrust Model in Experiment 1

From Figure 3.2, it can be seen that the Satisfaction Degrees of selected services of the PBTrust model are always higher than the CR model. That means, by including *Sim* and *SInd*, the PBTrust model can select better services than the CR model. It can also be found that the performances of the two models become closer as the Average Similarity Degree increases. This is because there are more possibilities for the CR model to select services with high *Sim* degrees when the average priorities of the service group is closer to the requested service priority. This result indicates that the PBTrust model is especially suitable for selecting service providers when there are not many expected providers available in the environment.

3.3.2 Experiment 2: evaluating the impact of EW

Setting of the service consumer of experiment 2

In Experiment 2, we want to test the impact of EW in the PBTrust Model (Recall Definition 3.3). In this experiment, we also included 70 service providers, with the same reference priority values and rating values with Experiment 1. In addition, we simulated four scenarios in this experiment to evaluate the impacts of EW. For each scenario, a different EW value was assigned to each provider. The average EW values of the seventy providers in the four scenarios are 20%, 40%, 60%, 80%, respectively.

In Experiment 2, we also adopted *SatDegree* as the evaluation criteria, but modify the calculation method by considering success possibility. Formula 3.13 shows the

calculation method of *SatDegree* in Experiment 2. The value of *SatDegree* equals to zero when a service is failed.

$$SatDegree = \begin{cases} Sim \times \frac{\sum_{i=1}^{n} R_i}{n \times 100} & : service\ executed \\ 0 & : service\ failed \end{cases} \tag{3.13}$$

Experimental results and analysis of experiment 2

Figure 3.3 shows the experimental results of Experiment 2.

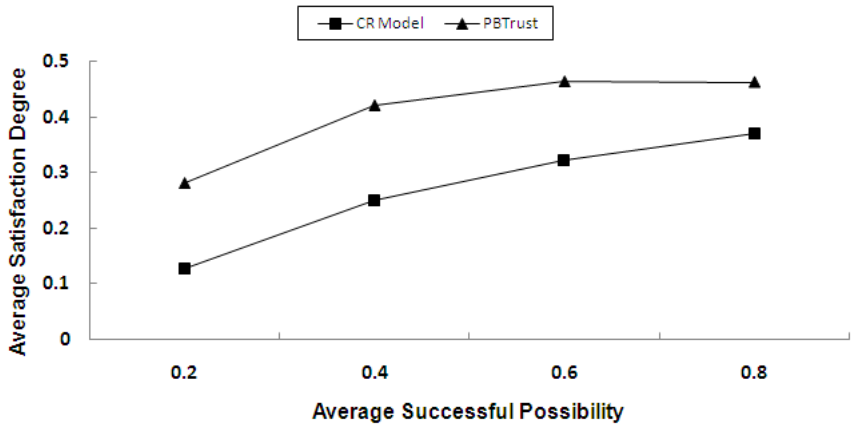


Figure 3.3: The Satisfaction Degrees of the Services Selected by the CR Model and the PBTrust Model in Experiment 2

From Figure 3.3, it can be seen that when the average experience weight is low (the first two scenarios), which means that most service providers have low success rates, the PBTrust model have much better performance than the CR model. As the Average Experience Weight of the providers increase, the performances of the two model will become closer. This is because there are more possibilities for the CR model to select providers with low success rates in the first first two scenarios, and that will cause high risks in service delivery. However, such problems can be avoided in the PBTrust model.

3.4 Summary

In this chapter, the PBTrust model was proposed. The PBTrust model uses the priority-based service description which divides a service into different attributes. By evaluating the priority of each attribute of the service, a service consumer can know

the difference between the requested service and the referenced service and can make the trust prediction on a potential provider more accurate. The experimental results under two different settings demonstrated that the PBTrust model outperformed the CR model in all situations.

Chapter 4

A Group Service Trust Model

Nowadays, many trust models have been proposed for single service provider selection. However, many complex service requests from consumers in recent years cannot be handled by a single service and a group of services from different providers are needed to satisfy these service requests [58, 55]. Therefore, trust models focusing on the trust evaluation 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 challenging issue. This chapter presents a new trust model, called GTrust, targeting to solve research issues identified in Section 1.2 of Chapter 1. The chapter is arranged as follows. Section 4.1 gives several definitions related to this model. The principle of the GTrust model is introduced in Section 4.2. Section 4.3 is the experiments and analysis. The summary of this chapter is given in Section 4.4

4.1 Problem Description and Definitions

In Chapter 3, we defined service description by Definition 3.1. For the notation purpose used in the GTrust model, we redefine the service description by Definition 4.1.

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

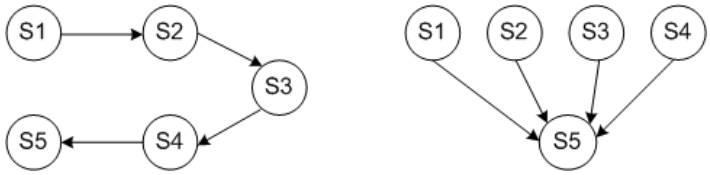
$$SDes = \begin{pmatrix} A_1 & A_2 & A_3 & ... & A_i \\ P_1 & P_2 & P_3 & ... & P_i \end{pmatrix} \tag{4.1}$$

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 4.2: 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 and R_i is a value in-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, suppose that two groups have the same individual services $S1$, $S2$, $S3$, $S4$ and $S5$, but with different workflows and dependency relationships as follows.



(a) Group 1 (b) Group 2

Figure 4.1: Workflows and Dependency Relationships of Services in Two Groups

In Figure 4.1, Group 1 has a sequential workflow to process from $S1$ to $S5$, i.e. a later service depends on its former service. However, the workflow in Group 2 is different from the workflow in Group 1. 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 are finished. We can see that there are no dependency relationships among S1 to S4 but four dependency relationships exist between S5 with other four services. In another word, S5 depends on $S1$, $S2$, $S3$ and $S4$, respectively. The dependency relationship among services can affect the performance of the services. For, example, in Group 1 if $S4$ has a bad performance and offers a bad result to $S5$, even if $S5$ have a very good performance on its service, the input from S4 can affect S5 normal performance. In order to identify relationships among services in a group, we introduce a concept of dependency degree.

Definition 4.3: 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 4.4: 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 4.2 as follows.

$$WDes = \begin{pmatrix} \lambda_{11}, \lambda_{12}, \lambda_{13}, ..., \lambda_{1n} \\ \lambda_{21}, \lambda_{22}, \lambda_{23}, ..., \lambda_{2n} \\ ..., ..., ..., ..., ... \\ \lambda_{n1}, \lambda_{n2}, \lambda_{n3}, ..., \lambda_{nn} \end{pmatrix} \tag{4.2}$$

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 4.5: 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.

The GTrust model consists of four modules which are the Request Module, the Reply Module, the Priority-based Group Trust Calculation Module and the Evaluation Module demonstrated by Figure 4.2.

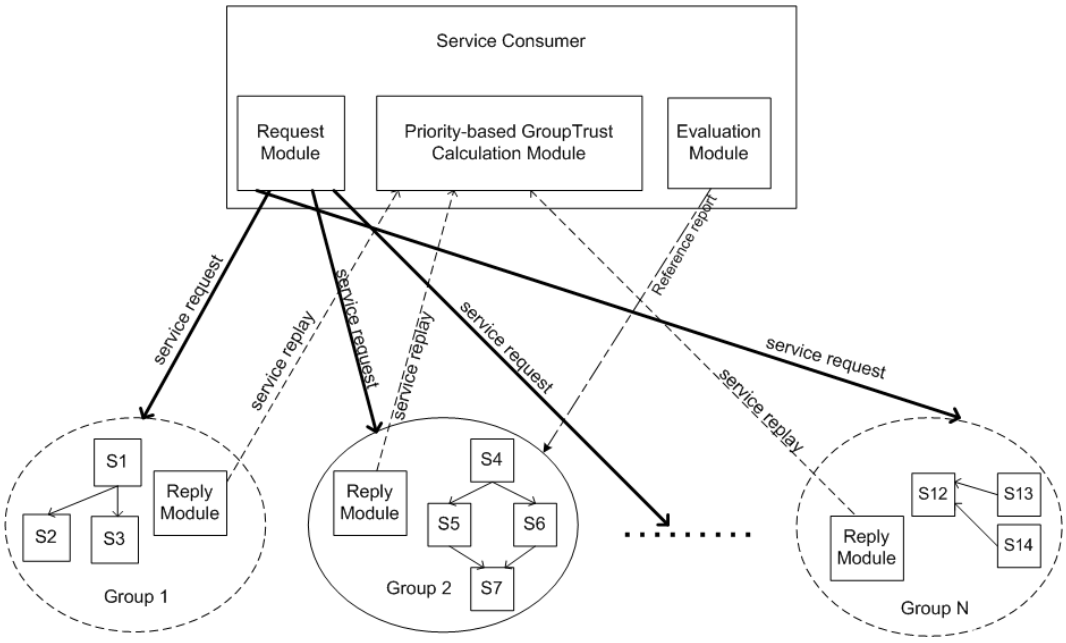


Figure 4.2: The Process of the GTrust Model

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 the 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; (4) After the selected service group finished the requested service, the evaluation module of the consumer will generate the reference report to describe the performance of the selected service group on each attribute of the requested service and send the reference report to the group. With the reference report, the members of the service group can dynamically update their service records.

4.2 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, respectively.

4.2.1 The request module

The objective of the Request Module is to create a *service request* based on a request from a consumer. For example, consumer C in an e-market environment requests a complex service described by five attributes, i.e. cost, speed, quality, colour and warranty with corresponding priority values for individual attributes as 0.1, 0.4, 0.2, 0.1 and 0.2, respectively. Based on the service request, the Request Module generates a service request in the format of *service description*, (recall Definition 4.1) as follows:

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

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

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

4.2.2 The reply module

The purpose 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 contain the following information: the group description of SG, and reference reports of individual services to demonstrate the best performance of individual members in SG.

Supposing SG is combined by 5 individual services named $S1$, $S2$, $S3$, $S4$ and $S5$, with the workflows and dependency relationships as follows.

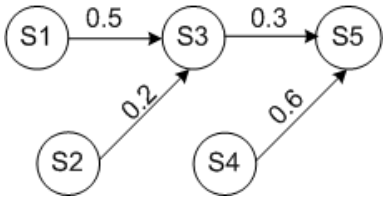


Figure 4.3: Services Workflow of the SG

In Figure 4.3, the arrows between services not only show the dependency relationships by the direction of arrows, but also indicate the dependency degree λ by the data along the arrows.

The *workflow description* of SG can be represented by Equation 4.3 in the format as follows.

$$WDes = \begin{pmatrix} 0,0,0.5,0,0 \\ 0,0,0.2,0,0 \\ 0,0,0,0,0.3 \\ 0,0,0,0,0.6 \\ 0,0,0,0,0 \end{pmatrix} \tag{4.3}$$

Each individual service in SG will present its best reference report. The reply module will create a service reply, (recall Definition 4.3) for SG in the following format including the *workflow description* of SG and a set of reference reports for five members, respectively.

4.2.3 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.

This module produces the trust value for a potential group based on three factors, which are 1) the functionality coverage on each attribute in a service group, 2) the similarity of priorities distributions on attributes between the service completed by members in the group and the requested service, and 3) the group rating on each attribute.

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 4.6: A group service description $GSDes$ is represented by a $m \times n$ matrix, where 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.

$$GSDes = \begin{pmatrix} A_1 & A_2 & ... & A_n \\ P_{11} & P_{12} & ... & P_{1n} \\ P_{21} & P_{22} & ... & P_{2n} \\ ... & ... & ... & ... \\ P_{m1} & P_{m2} & ... & P_{mn} \end{pmatrix} \tag{4.4}$$

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, because different service consumers often pay attention to different attributes for the same service; otherwise P_{ij} is in-between [0,1], where 0 and 1 represent the highest and lowest priority values, respectively. By using Equation 4.4, the comprehensive ability of a service group can be described.

For example if in the former interactions the service provider only pay attention to

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 4.7: 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 4.6) as follows.

$$ACov_i = \frac{m - MS_i}{m} \tag{4.5}$$

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 by the following formula.

$$VSim = \frac{\sum_{i=1}^{n} V1_i \cdot V2_i}{\sqrt{(\sum_{i=1}^{n}(V1_i)^2) \cdot (\sum_{k=1}^{n}(V2_i)^2)}} \tag{4.6}$$

where $VSim$ is the similarity of priority distribution between two vectors $V1$ and $V2$, and $V1_i$ and $V2_i$ represent the priority values of i^{th} elements of the two vectors, respectively. To calculate the similarity of priority distribution between a group service and the requested service, 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.

$$GP_i = \sum_{i=1}^{m} P_{ij} \tag{4.7}$$

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

With the Equation 4.7, we can calculate each element in Vector GPV, then normalise two vectors if necessary before using the dot product. The group similarity calculation can be obtained by the following formula.

$$GSim = \frac{\sum_{i=1}^{n} NGP_i \cdot NP_i}{\sqrt{(\sum_{i=1}^{n}(NGP_i)^2) \cdot (\sum_{k=1}^{n}(NP_i)^2)}} \tag{4.8}$$

where $GSim$ is the similarity between the priority distribution of the requested service and a service completed by the service group, NP_i and NGP_i represent the normalised

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 for the requested service in j^{th} attribute is calculated as follows.

$$GRating_j = \frac{\sum_{i=1}^{m} FRating_{ij}}{m} \tag{4.9}$$

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.

$$FRating_{ij} = Rating_{ij} - \frac{\sum_{k=1}^{n} \lambda_{ki} \cdot (1 - FRating_{kj})}{n} \tag{4.10}$$

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 final performance rating of the k^{th} dependency service on j^{th} attributes, and λ_{ki} is the dependency degree of the i^{th} individual service depending on the k^{th} dependency service.

If the performance rating value of an individual service on an attribute dose not exist, we will use the exist average ratings of other individual services in this group to represent the missing performance of an individual 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.

$$Trust = GSim \cdot \sum_{i=1}^{n} P_i \cdot ACov_i \cdot GRating_i \tag{4.11}$$

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.

4.2.4 The evaluation module

After completing the requested service, the consumer uses the evaluation module to generate a reference report for the service group based on its performance on the requested service.

We use the same example as in Request Module and Reply Module to demonstrate how to generate a reference report in this module. After completing the requested service, Consumer C evaluates the performance of a service group SG on the requested service. The evaluation result is represented in a format of reference report, (recall Definition 4.2) shown as follows.

$$Rf =< SDes, < 0.6, 0.4, 0.9, 0.9, 0.3 >>$$

From the consumer's rating, we can see that the consumer was satisfied with the cost of the service (i.e. the first attribute of $SDes$),was not satisfied with the speed and warranty of the service (i.e. the second and fifth attributes of $SDes$), and was very satisfied with the quality and the 'colour' attribute of the service (i.e. the third and forth attributes of $SDes$). This report will be used to update the record of historical performance of each member in the group.

4.3 Experiment

After researching the related area, we found a suitable model that can be used to compare with the GTrust model. Therefore, we use the average trust value of all individual services in a service group which was introduced in the REGRET model as the Benchmark model [46] without consideration the dependance relationships among individual services.

4.3.1 Experiment setting

In the experiment, one service consumer and twenty services offered by different service providers are employed. The twenty individual services are divided into four service groups and each group contains five individual services owned by five providers. Each

group is tested by three scenarios according to different workflows and dependency relationships among individual services in the group.

The service consumer sends a service request containing five attributes, *cost*, *speed*, *quality*, *colour* and *warranty* with different priorities, respectively. The service request is described as bellow.

$$SDes' = \begin{pmatrix} Cost & Speed & Quality & Colour & Warranty \\ 0.12 & 0.08 & 0.2 & 0.1 & 0.5 \end{pmatrix}$$

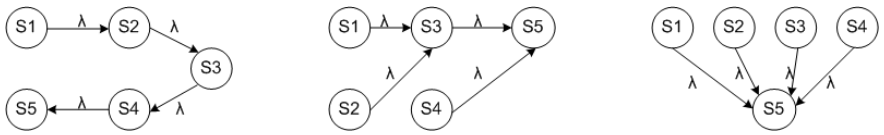
Each provider provides its best reference report to demonstrate its good performance so that each group has five reference reports from five members. Table 4.1 shows our experimental data.

Group ID	Provider ID	Priority Distribution in reference reports	Ratings in reference reports
G1	P_1	(0.7, 0.1, 0.1, 0.1, m)	$< 0.2, 0.2, 0.2, 0.2, 0 >$
G1	P_2	(0.6, m, 0.1, 0.1, 0.2)	$< 0.2, 0, 0.2, 0.2, 0.2 >$
G1	P_3	(m, 0.6, 0, 0.2, 0.2)	$< 0, 0.2, 0.2, 0.2, 0.2 >$
G1	P_4	(0.8, 0.1, m, 0.1, m)	$< 0.2, 0.2, 0, 0.2, 0.2 >$
G1	P_5	(0.6, 0.2, 0.2, m, m)	$< 0.2, 0.2, 0.2, 0, 0.2 >$
G2	P_6	(0.2, m, 0.5, 0.3, m)	$< 0.6, 0, 0.6, 0.6, 0 >$
G2	P_7	(0.1, 0.2, m, 0.7, m)	$< 0.6, 0.6, 0, 0.6, 0 >$
G2	P_8	(m, 0.2, 0.2, m, 0.6)	$< 0, 0.6, 0.6, 0, 0.6 >$
G2	P_9	(m, 0.1, m, 0.3, 0.6)	$< 0, 0.6, 0, 0.6, 0.6 >$
G2	P_{10}	(0.2, m, 0.2, m, 0.6)	$< 0.6, 0, 0.6, 0, 0.6 >$
G3	P_{11}	(0.1, m, m, 0.9, m)	$< 0.4, 0, 0, 0.4, 0 >$
G3	P_{12}	(m, m, 0.1, m, 0.9)	$< 0, 0, 0.4, 0, 0.4 >$
G3	P_{13}	(m, 0.2, m, m, 0.8)	$< 0, 0.4, 0, 0, 0.4 >$
G3	P_{14}	(m, 0.2, m, 0.8, m)	$< 0, 0.4, 0, 0.4, 0 >$
G3	P_{15}	(0.1, m, 0.9, m, m)	$< 0.4, 0, 0.4, 0, 0 >$
G4	P_{16}	(1, m, m, m, m)	$< 0.8, 0, 0, 0, 0 >$
G4	P_{17}	(m, 1, m, m, m)	$< 0, 0.8, 0, 0, 0 >$
G4	P_{18}	(m, m, 1, m, m)	$< 0, 0, 0.8, 0, 0 >$
G4	P_{19}	(m, m, m, 1, m)	$< 0, 0, 0, 0.8, 0 >$
G4	P_{20}	(m, m, m, m, 1)	$< 0, 0, 0, 0, 0.8 >$

Table 4.1: The References Reports of Twenty Service Providers

To comprehensively test the influence of different functionality coverage on the trust calculation of the GTurst model, we set functionality coverage values for four service groups as Group 1: 80%, Group 2: 60%, Group 3: 40%, Group 4: 20%. The three different workflows in three scenarios are shown in Figure 4.4.

In each scenario, we evaluate the trust value for four groups using 5 dependency degree values 0, 0.1, 0.5, 0.7 and 1. The reason for this setting is to comprehensively



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

Figure 4.4: Workflows of Individual Services in A Group in Three Scenarios

test the influence of different dependency degrees on trust calculation of the GTrust model.

Trust value transfer function

Since the GTrust model and the Benchmark model use different presentations on trust values, we define the following trust value transfer function to transfer trust values from the GTurst model to the corresponding trust values in the Benchmark model.

$$f(M,V) = \frac{\sum_{i=1}^{n} R_i}{n - MS} \tag{4.12}$$

where M is the service description matrix of a referenced service in GTrust, V is the rating-vector of a service provider from the referee in GTrust, and R_i is the rating of the i^{th} attribute and n is the number of attributes in the requested service and MS is the number of missed attributes in services provided by individual members of the service group.

4.3.2 Experimental results and analysis

This subsection gives the detail experimental results and analysis in three scenarios, respectively. Further, this subsection also gives the detail discussion about the impact of service structures on group trust values.

Scenario 1:

(1) Experimental results

Table 4.2 shows the calculation results of similarities, functionality coverage and ratings for four groups in the GTrust model under five dependency degrees in Scenario 1 (refer to Figure 4.4(a)).

	G1	G2	G3	G4
$GSim$	0.364953	0.980357	1.000000	0.846364
$ACov$	(0.8, 0.8, 0.8, 0.8, 0.8)	(0.6, 0.6, 0.6, 0.6, 0.6)	(0.4, 0.4, 0.4, 0.4, 0.4)	(0.2, 0.2, 0.2, 0.2, 0.2)
Ratings				
$\lambda = 0$	(0.2, 0.2, 0.2, 0.2, 0.2)	(0.6, 0.6, 0.6, 0.6, 0.6)	(0.4, 0.4, 0.4, 0.4, 0.4)	(0.8, 0.8, 0.8, 0.8, 0.8)
$\lambda = 0.1$	(0.13, 0.13, 0.13, 0.13)	(0.57, 0.57, 0.57, 0.57)	(0.35, 0.35, 0.35, 0.35)	(0.78, 0.78, 0.78, 0.78)
$\lambda = 0.5$	(0.04, 0.04, 0.04, 0.04)	(0.36, 0.36, 0.36, 0.36)	(0.1, 0.1, 0.1, 0.1)	(0.68, 0.68, 0.68, 0.68)
$\lambda = 0.7$	(0.04, 0.04, 0.04, 0.04)	(0.21, 0.21, 0.21, 0.21)	(0.08, 0.08, 0.08, 0.08)	(0.59, 0.59, 0.59, 0.59)
$\lambda = 1$	(0.04, 0.04, 0.04, 0.04)	(0.16, 0.16, 0.16, 0.16)	(0.08, 0.08, 0.08, 0.08)	(0.4, 0.4, 0.4, 0.4)

Table 4.2: The Similarity, Functionality Coverage and Ratings of Four Service Groups in Five Dependency Degrees in Scenario 1

Table 4.3 shows the trust value of four groups calculated by the GTrust model and the Benchmark model, respectively, under five dependency degrees in Scenario 1.

	GTrust				Benchmark			
	G1	G2	G3	G4	G1	G2	G3	G4
$\lambda = 0$	0.110509	0.667927 (C)	0.302804	0.256283	0.2	0.6	0.4	0.8(C)
$\lambda = 0.1$	0.072309	0.629446 (C)	0.263552	0.250746	0.2	0.6	0.4	0.8(C)
$\lambda = 0.5$	0.022102	0.395190 (C)	0.075701	0.217040	0.2	0.6	0.4	0.8(C)
$\lambda = 0.7$	0.022102	0.232439 (C)	0.060561	0.189699	0.2	0.6	0.4	0.8(C)
$\lambda = 1$	0.022102	0.178114 (C)	0.060561	0.128141	0.2	0.6	0.4	0.8(C)

Table 4.3: The Trust Values of Four Service Groups in Scenario 1

Figure 4.5 demonstrates the changes of trust values of Group 2 selected by the GTrust model and Group 4 selected by the Benchmark model, under five dependency degrees in Scenario 1. In Figure 4.5, X-axis represents the dependency degree values while Y-axis represents the trust values.

Form the Table 4.3, we can see that each service group except Group 2 have their own strength on the requested service. Group 1 has the highest functionality coverage rate on the attributes of the requested service, Group 3 has the highest similarity values of priority distribution with the requested service, and Group 4 has the highest performance ratings on the requested service. In principle, the Benchmark model always chooses a potential service group based on the average rating. Therefore, in Scenario 1, the Benchmark model always chooses Group 4 since Group 4 has the highest average rating values among four groups under five dependency degrees. The GTrust model considers four main factors including functionality coverage, dependency relationship, similarities and performance rating, which can affect the performance of a service group

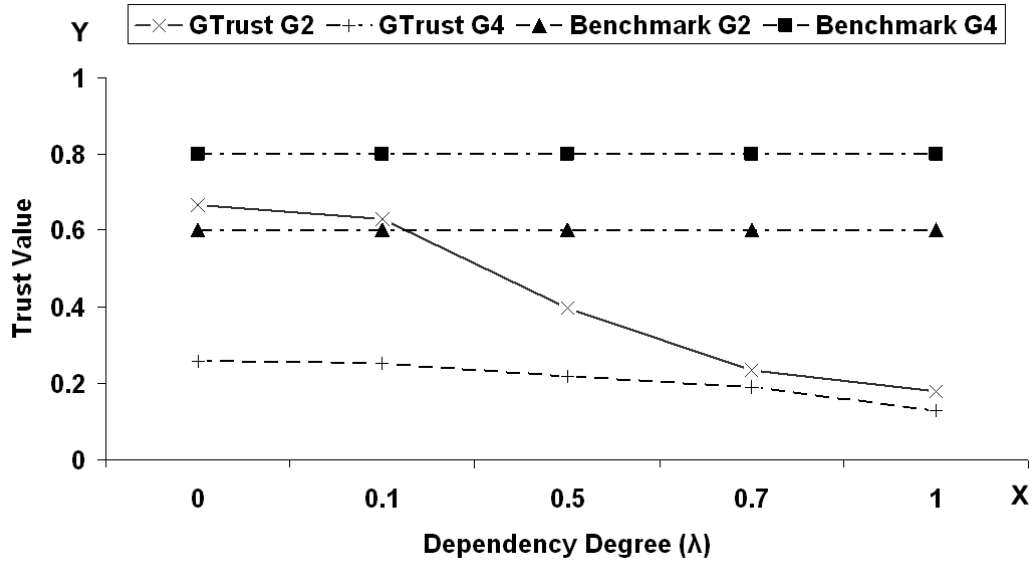


Figure 4.5: Trust Value of Service Groups Selected by Two Trust Models in Scenario 1

from different perspectives. After the calculation the GTrust model chooses Group 2 for the requested service because Group 2 has the second best value on each factors but the best trust value on the synthetical performance after the consideration of the four main factors.

(2) Analysis and discussion

Now, we analyse the selection results from two models to see which service group is more suitable for the requested service from four perspectives.

a. Workflow structures From Figure 4.4(a), we can see that the four groups all have a sequential workflow in this scenario, i.e. a later service depends on its former service. In another word, the performance of the former service in term of rating will affect the performance of the later service. From Figure 4.5, we can clearly see that dependency degrees have an impact on group trust value calculation in the GTrust model. The experimental result of the GTrust model in this scenario shows that when the dependency degree (λ) among individual services in a group increases, the group trust values decreases. For example, we can see from Figure 4.5 and Table 4.3 that when the dependency degree increases from 0 to 1, the trust value of Group 2 decreases from 0.67 to 0.18 and the trust value of Group 4 decreases from 0.26 to 0.13. Figure 4.5 also indicates that the trust values of Group 2 and Group 4 calculated by the Benchmark model always keep as constants i.e 0.8 and 0.6, respectively, under all dependency degrees. In

the real world environment, a service group always has its workflows and there are dependency relationships among individual services. To this consideration, the trust calculation process of GTrust model is closer to realistic situations than the Benchmark model.

b. Similarity of priority distributions The reference reports offered by the former service consumers can reflect the ability of individual services on the requested service. The more similar between the priority distribution of the requested service and the reference reports in a service group, the better the reference reports reflects the performance of the potential service group on the requested service. From the priority distribution on five attributes of the service request (refer to Subsection 5.1), we can see that a service group meeting requests should have the best performance on 'Warranty' attribute, the second best performance on 'Quality' attribute, then the 'Cost' attribute, the 'colour' attribute, and the 'Speed' attribute. Table 4.1 shows the priority distributions in reference reports provided by each individual services of four service groups. In Group 4, only P_{20} has the highest priority distribution on 'Warranty' attribute and the priority distribution of other four individual services (P_{16} to P_{19}) focus on other four attributes, respectively. However, in Group 2, P_8 to P_{10} have the highest priority value (0.6) on 'Warranty' attribute and P_8 and P_{10} have the priority value of 'Quality' attribute 0.2, which is the second highest priority value among five attributes. After comparing the priority distributions of Group 2 and Group 4, we can see that the reference reports of Group 2 can better reflect the performance of the service group on the requested service than that of Group 4.

c. Functionality coverage The functionality coverage value can reflect whether a potential service group can offer the functionality of the requested service. In another words, whether a potential service group can cover all the attributes of the requested service. Table 4.1 also shows the functionality coverage values of individual services in four service groups. In Group 4, the individual services (from P_{16} to P_{20}) miss too many priority values on the attribute and each functionality coverage rate on the attributes of the requested service is only 20%. In Group 2, even if the individual services also miss some attributes in reference reports, the functionality coverage rate is 60%, which is higher than that of in Group 4. From this consideration, Group 2 is more suitable than Group 4 for the requested service.

d. Final decision making The Benchmark model chooses Group 4 based on its the potential service group based on performance rating (0.8). The GTrust model chooses Group 2 not only base on its performance rating (0.6), but its workflow structure, similarity of priority distributions, as well as its functionality coverage of the requested service.

Therefore, the GTrust model has the better performance than the Benchmark model on group service selection in Scenario 1.

Scenario 2:

(1) Experimental results

Table 4.4 shows the calculation results of similarity, functionality coverage and ratings for four groups in the GTrust model under five dependency degrees in Scenario 2 (refer to Figure 4.4(b)).

	G1	G2	G3	G4
$GSim$	0.364953	0.980357	1.000000	0.846364
$ACov$	(0.8, 0.8, 0.8, 0.8, 0.8)	(0.6, 0.6, 0.6, 0.6, 0.6)	(0.4, 0.4, 0.4, 0.2, 0.6)	(0.2, 0.2, 0.2, 0.2, 0.2)
Ratings				
$\lambda = 0$	(0.2, 0.2, 0.2, 0.2, 0.2)	(0.6, 0.6, 0.6, 0.6, 0.6)	(0.4, 0.4, 0.4, 0.4, 0.4)	(0.8, 0.8, 0.8, 0.8, 0.8)
$\lambda = 0.1$	(0.17, 0.17, 0.17, 0.17)	(0.58, 0.58, 0.58, 0.58)	(0.38, 0.38, 0.38, 0.38)	(0.79, 0.79, 0.79, 0.79)
$\lambda = 0.5$	(0.12, 0.12, 0.12, 0.12)	(0.51, 0.51, 0.51, 0.51)	(0.27, 0.27, 0.27, 0.27)	(0.76, 0.76, 0.76, 0.76)
$\lambda = 0.7$	(0.12, 0.12, 0.12, 0.12)	(0.47, 0.47, 0.47, 0.47)	(0.24, 0.24, 0.24, 0.24)	(0.73, 0.73, 0.73, 0.73)
$\lambda = 1$	(0.12, 0.12, 0.12, 0.12)	(0.4, 0.4, 0.4, 0.4)	(0.24, 0.24, 0.24, 0.24)	(0.7, 0.7, 0.7, 0.7)

Table 4.4: The Similarity, Functionality Coverage and Ratings of Four Service Groups in Five Dependency Degrees in Scenario 2

Table 4.5 shows the trust values of four groups calculated by the GTrust model and the Benchmark model in Scenario 2 under five different dependency degrees in Scenario 2.

Figure 4.6 indicates the changes of trust values of the service groups selected by the GTrust model and the Benchmark model, respectively, under five dependency degrees in Scenario 2. In Figure 4.6, X-axis represents the dependency degree and Y-axis represents the trust value.

	GTrust				Benchmark			
	G1	G2	G3	G4	G1	G2	G3	G4
$\lambda = 0$	0.110509	0.667927 (C)	0.302804	0.256283	0.2	0.6	0.4	0.8(C)
$\lambda = 0.1$	0.092386	0.649670 (C)	0.284182	0.253656	0.2	0.6	0.4	0.8(C)
$\lambda = 0.5$	0.066306	0.567738 (C)	0.200608	0.241867	0.2	0.6	0.4	0.8(C)
$\lambda = 0.7$	0.066306	0.521428 (C)	0.181683	0.235204	0.2	0.6	0.4	0.8(C)
$\lambda = 1$	0.066306	0.445285 (C)	0.181683	0.224248	0.2	0.6	0.4	0.8(C)

Table 4.5: The Trust Values of Four Service Groups in Scenario 2

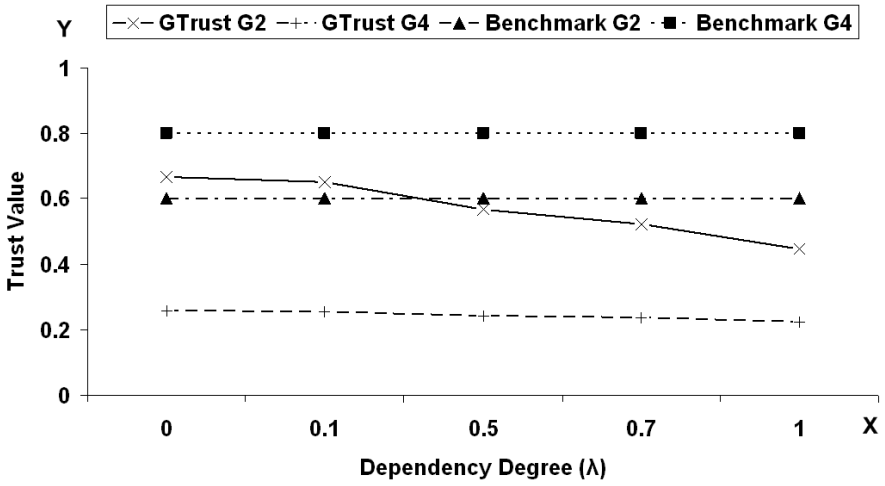


Figure 4.6: Trust Value of Service Groups Selected by Two Trust Models in Scenario 2

(2) Analysis and discussion

In this scenario, the GTrust model also chooses Group 2 while the Benchmark model chooses Group 4. Using the same analysis as Scenario 1, we can see that the potential service group selected by the GTrust model have better performance than that of selected by the Benchmark model after considering workflow structure, similarity values and the functionality coverage rate.

Scenario 3

(1) Experimental results

Table 4.6 shows the calculation results of similarity, functionality coverage and ratings for four groups in the GTrust model under five dependency degrees in Scenario 3 (refer to Figure 4.4(c)).

	G1	G2	G3	G4
$GSim$	0.364953	0.980357	1.000000	0.846364
$ACov$	(0.8, 0.8, 0.8, 0.8, 0.8)	(0.6, 0.6, 0.6, 0.6, 0.6)	(0.4, 0.4, 0.4, 0.2, 0.6)	(0.2, 0.2, 0.2, 0.2, 0.2)
Ratings				
$\lambda = 0$	(0.2, 0.2, 0.2, 0.2, 0.2)	(0.6, 0.6, 0.6, 0.6, 0.6)	(0.4, 0.4, 0.4, 0.4, 0.4)	(0.8, 0.8, 0.8, 0.8, 0.8)
$\lambda = 0.1$	(0.18, 0.18, 0.18, 0.18)	(0.59, 0.59, 0.59, 0.59)	(0.39, 0.39, 0.39, 0.39)	(0.8, 0.8, 0.8, 0.8)
$\lambda = 0.5$	(0.16, 0.16, 0.16, 0.16)	(0.56, 0.56, 0.56, 0.56)	(0.34, 0.34, 0.34, 0.34)	(0.78, 0.78, 0.78, 0.78)
$\lambda = 0.7$	(0.16, 0.16, 0.16, 0.16)	(0.54, 0.54, 0.54, 0.54)	(0.32, 0.32, 0.32, 0.32)	(0.77, 0.77, 0.77, 0.77)
$\lambda = 1$	(0.16, 0.16, 0.16, 0.16)	(0.52, 0.52, 0.52, 0.52)	(0.32, 0.32, 0.32, 0.32)	(0.76, 0.76, 0.76, 0.76)

Table 4.6: The Similarity, Functionality Coverage and Ratings of Four Service Groups in Five Dependency Degrees in Scenario 3

Table 4.7 shows the trust value of four groups calculated by the GTrust model and the Benchmark model, respectively, under five dependency degrees in Scenario 3.

	GTrust				Benchmark			
	G1	G2	G3	G4	G1	G2	G3	G4
$\lambda = 0$	0.110509	0.667927 (C)	0.302804	0.256283	0.2	0.6	0.4	0.8(C)
$\lambda = 0.1$	0.101669	0.659021 (C)	0.293720	0.255001	0.2	0.6	0.4	0.8(C)
$\lambda = 0.5$	0.088408	0.623399 (C)	0.257384	0.249876	0.2	0.6	0.4	0.8(C)
$\lambda = 0.7$	0.088408	0.605587 (C)	0.242244	0.247313	0.2	0.6	0.4	0.8(C)
$\lambda = 1$	0.088408	0.578870 (C)	0.242244	0.243469	0.2	0.6	0.4	0.8(C)

Table 4.7: The Trust Values of Four Service Groups in Scenario 3

Figure 4.7 demonstrates the changes of trust values of the service groups selected by the GTrust model and the Benchmark model, respectively, under five dependency degrees in Scenario 3. In Figure 4.7, X-axis represents the dependency degree and Y-axis represents the trust value.

(2) Analysis and discussion

From Table 4.7 and Figure 4.7, we can see that the GTrust model still choose Group 2 while the Benchmark model still chooses Group 4. Using the same analysis as in Scenario 1, we can convince that the selection result from the GTrust model is more reasonable than that of from the Benchmark model.

The impact of workflows of services on group trust values

In the experiment, we used three scenarios with three different workflow structures of services compositions. Now we analyse the potential impact from different workflows of

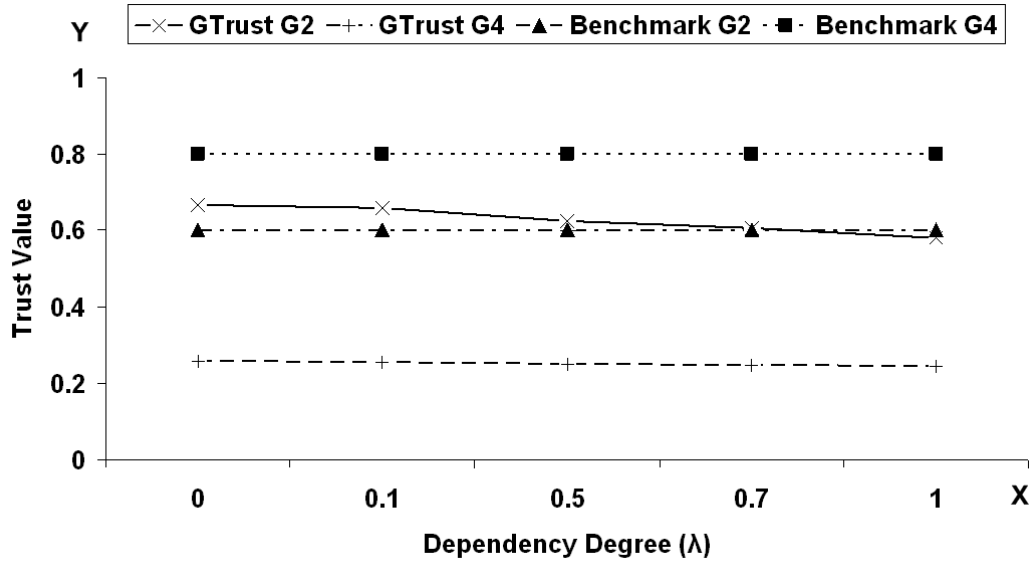


Figure 4.7: Trust Value of Service Groups Selected by Two Trust Models in Scenario 3

services on the group trust values. Figure 4.8 demonstrates the changes of trust values of Group 2 and Group 4 selected by the GTrust model and the Benchmark model, respectively, under five dependency degrees in three scenarios, where X-axis represents dependency degrees and Y-axis represents trust values.

From Figure 4.8, we can clearly see that when the dependency degree changes in three scenario the group trust of Group 4 selected by the Benchmark model always keeps as an constant while the group trust of Group 2 selected by the GTrust model are various. In Scenario 1, the five individual services of Group 2 is a sequential workflow (refer to Figure 4.4 (a)) and there are four dependency relationships among individual services. Therefore, the group trust in Scenario 1 highly depends on four dependency relationships and we can see from Figure 4.8 that the trust value decreases significantly when the dependency degree increases. In Scenario 2, there are also four dependency relationships among individual services in Group 2. However, the dependency structure is different from that of in Scenario 1. For example, $S3$ depends on $S1$ and $S2$. That means either $S1$ or $S2$ has partial influence on $S3$. The same relationships exist between $S5$ with $S3$ and $S4$. $S4$ has no any dependency relationship with $S1$, $S2$ and $S3$. Scenario 2 has weaker dependency relationships among group members so that when dependency degree (λ) increases, the group trust decreases but the decrease rate is smaller than that of in Scenario 1. Scenario 3 has the slowest dropping speed for group trust when the dependency degree increases among three scenarios because each

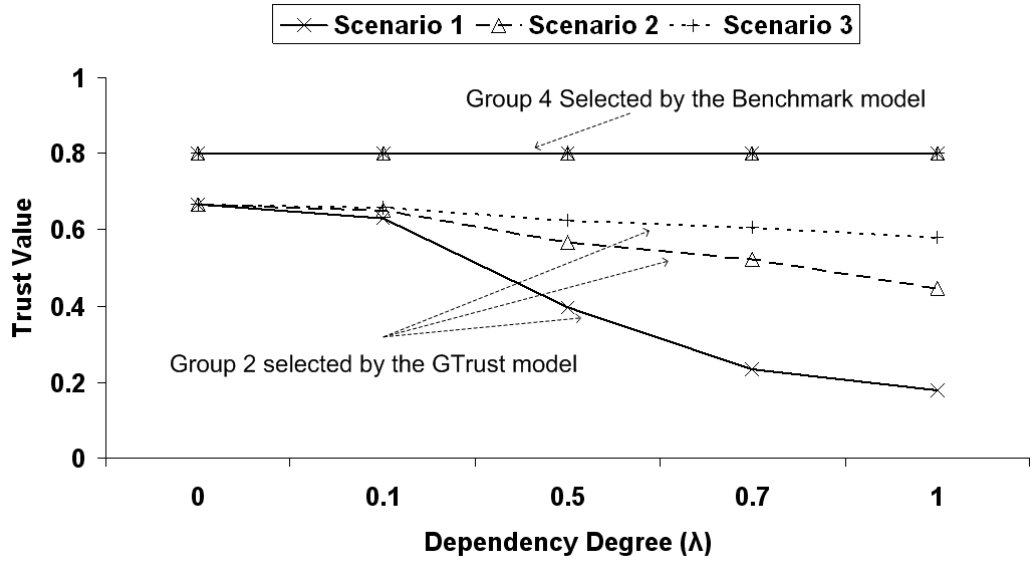


Figure 4.8: Trust Value of Service Groups Selected by the GTrust Model and the Benchmark Model in Three Scenarios

dependency service ($S1$ to $S4$) has partial impact on $S5$. From the above analysis the group trust from the GTrust model is more reasonable than that of from the Benchmark model, because we take consideration of the structure of composition.

In summary, the group trust is not only impacted by dependency degrees among services but also impacted by the structure of the combination of services in a group.

4.4 Summary

In this chapter, the GTrust model for group services selection in service-oriented MASs was proposed. This model is novel, since the GTrust model considers the four main factors that effect the trust value of a service group by using the following mechanisms.

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 a consumer.

2. The GTrust model introduces the concept of 'dependency degree' to represent relationships among 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 can dynamically

update the reference reports of individual services without a central controller.

4. The GTrust model uses the concept of 'similarity' to measure the similarity in terms of priority distributions on attributes between historical services of group members and a requested service.

The experimental results indicated the good performance for group service providers selection in three scenarios.

Chapter 5

Conclusion

Service provider selection is an important and a challenging research topic in service-oriented MASs.

The main objectives of this thesis are

1 To study the challenging issues of trust-based service provider selection in general service-oriented MASs.

2 To investigate the current approaches of trust models for service provider selection in general service-oriented MASs.

3 To develop new solutions for service provider selection to overcome several limitations in current existing approaches in both single service provider selection and group service providers selection.

To achieve the above objectives, two trust models were developed during this study for service provider selection.

In Section 5.1, the contributions of this thesis are delightful emphasised. In Section 5.2, the limitations of two proposed models are discussed and the future work is outlined.

5.1 Major Contributions of This Thesis

In Chapters 3 and 4, we proposed two trust models which are the PBTrust model for single service provider selection and the GTrust model for group service providers selection in general service-oriented MASs, respectively. The contributions of each trust model are outlined in the following two subsections.

5.1.1 The contributions of the PBTrust model

The PBTrust model is a full-context priority-based trust model for single service provider selection. In the PBTrust model, a trust value of a service provider is evaluated from four factors, which are 1) the provider's experience on the service, 2) the similarity of priorities distributions on attributes between the referenced service and the requested service, 3) the suitability of the potential provider for the requested service, and 4) the time effects on ratings from third party references.

The experimental results and analysis demonstrated the good performance of the PBTrust model in single service provider selections, since the PBTrust model includes the following advantages:

1 The PBTrust model considers the attributes of a service and uses priorities to distinguish the importance of different attributes. This feature allows more objective evaluations on both required services and providers' reputations.

2 The PBTrust model uses a relatively easy way to describe different attributes of a service.

3 The PBTrust model introduces the concept of experience weight which can avoid un-objective references and cheating references.

4 The PBTrust model uses general experience to describe service provider's experience.

The above advantages are the contributions of the PBTrust model to the research and development of trust models for single service provider selection.

5.1.2 The contributions of the GTrust model

The GTrust model offers an innovated way for group service providers selection by considering three additional challenging issues emerging in group trust evaluation, which are 1) the structure of the service group, 2) the dependency relationships among group members, and 3) the efficiency of the service group.

The GTrust model has the following advantages:

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 a consumer.

2 The GTrust model introduces the concept of 'dependency degree' to represent relationships among services in a service group and provides an innovated method to capture structures and workflows of group services.

3 The GTrust model extents the concept of 'third party reference' of the PBTrust model to represent the performance of individual services in a service group.

4 The GTrust model uses 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 above advantages demonstrate the contributions of the GTrust model to the research by solving challenging issues in group service providers selection.

5.2 Remaining Problems and Future Work

Although a number of challenging issues in service provider selection for both single service provider and group service providers have been solved in this thesis, there are still some remaining problems which need to be dealt with in the future.

1 More complex situations need to be considered.

When designing the proposed trust models, we only focused on how to accurately evaluate the trust value for a potential service provider. Therefore, we just considered the most general situation of service provider selection, i.e. only one service consumer to request a service and there exist sufficient service providers which can offer the requested service. However, in real world application, there are other situations which may be different from the general situation. For example, there may be two service consumers who compete for one service provider. In the future, we will add more mechanisms to deal with other situations.

2 The different priorities of individual services in a service group need to be considered.

In the GTrust model, we calculated the trust value for a service group without considering the different roles of individual services. In a service group, there always exist core services which have greater effects on the trust value of the group than other services. Therefore, in the trust value calculation for a service group, the trust values of these core services should be assigned more impact

factors on the trust value of the group. In the future, we will assign different impact factors for different group members based on their roles in a service group.

3 Better simulation tools need to be involved

In the experiments, we used the general programming and calculation tools, which are $Visual\ C++$ and $MATLAB$ 7.0 [40], to test the proposed trust models. Although the experimental results of the two trust models are good enough for this study to prove the better performance of our trust models than the benchmark models, there might be some errors in our test data sets. In the future, we will find good supporting tools such as AgentBuilder and ABLE, and Q-Learning method to generate and analyse test data for our trust models.

Bibliography

[1] Alfarez Abdul-Rahman and Stephen Hailes. Supporting trust in virtual communities. In *Proceedings of the 33rd Hawaii International Conference on System Sciences*, volume 6 of *HICSS '00*, pages 6007–6015, Maui, Hawaii, 2000. IEEE Computer Society.

[2] James Allen, Nate Blaylock, and George Ferguson. A problem solving model for collaborative agents. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, AAMAS '02, pages 774–781, New York, NY, 2002. ACM.

[3] Amazon. http://www.amazon.com, accessed on 16th. August 2011.

[4] K. Arnold, L. Bordoli, J. Kopp, and T. Schwede. The swiss-model workspace: a web-based environment for protein structure homology modelling. *Bioinformatics*, 22(2):195–201, 2006.

[5] N. Avouris, A. Dimitracopoulou, V. Komis, and C. Fidas. Ocaf: an object-oriented model of analysis of collaborative problem solving. In *Proceedings of the Conference on Computer Support for Collaborative Learning: Foundations for a CSCL Community*, CSCL '02, pages 92–101, Boulder, Colorado, 2002. International Society of the Learning Sciences.

[6] S. Balasubramaniam, G. Lewis, E. Morris, S. Simanta, and D. Smith. Challenges for assuring quality of service in a service-oriented environment. In *Proceedings of the 2009 ICSE Workshop on Principles of Engineering Service Oriented Systems*, PESOS '09, pages 103–106, Vancouver, Canada, 2009. IEEE Computer Society.

[7] R. Bellman and L. Zadeh. Decision-making in a fuzzy environment. *Management Science*, 17(4):pp. B141–B164, 1970.

[8] K. Bennett and E. Parrado-Hernández. The interplay of optimization and machine learning research. *J. Mach. Learn. Res.*, 7:1265–1281, December 2006.

[9] S. Bromuri, V. Urovi, M. Morge, K. Stathis, and F. Toni. A multi-agent system for service discovery, selection and negotiation. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems*, volume 2 of *AAMAS '09*, pages 1395–1396, Budapest, Hungary, 2009. International Foundation for Autonomous Agents and Multiagent Systems.

[10] C. Carrascosa, A. Giret, V. Julian, M. Rebollo, E. Argente, and V. Botti. Service oriented mas: an open architecture. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '09, pages 1291–1292, Budapest, Hungary, 2009. International Foundation for Autonomous Agents and Multiagent Systems.

[11] J. Carter, E. Bitting, and A. Ghorbani. Reputation formalization for an information-sharing multi-agent sytem. *Computational Intelligence*, 18(4):515–534, 2002.

[12] C. Castelfranchi and R. Falcone. Principles of trust for mas: Cognitive anatomy, social importance, and quantification. In *Proceedings of the 3rd International Conference on Multi Agent Systems*, ICMAS '98, pages 72–79, San Francisco, CA, 1998. IEEE Computer Society.

[13] E. Chang, F. Hussain, and T. Dillon. Fuzzy nature of trust and dynamic trust modeling in service oriented environments. In *Proceedings of the 2005 workshop on Secure web services*, SWS '05, pages 75–83, Fairfax, Virginia, 2005. ACM.

[14] P. Chen, J. Chang, T. Liang, C. Shieh, and Y. Zhuang. A multi-layer resource reconfiguration framework for grid computing. In *Proceedings of the 4th international workshop on Middleware for grid computing*, MCG '06, pages 13–18, Melbourne, Australia, 2006. ACM.

[15] W. Conner, A. Iyengar, I. Mikalsen, T.and Rouvellou, and K. Nahrstedt. A trust management framework for service-oriented environments. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 891–900, Madrid, Spain, 2009. ACM.

[16] H. Deitel, P. Deitel, and T. Nieto. *e-Business & e-Commerce How to Program.* Prentice Hall, 2001.

[17] eBay. http://www.ebay.com, accessed on 16th. August 2011.

[18] B. Esfandiari and S. Chandrasekharan. *On how agents make friends: Mechanisms for trust acquisition*, volume 222, pages 27–34. Citeseer, 2001.

[19] Jacques Ferber. *Multi-Agent System: An Introduction to Distributed Artificial Intelligence*. Addison-Wesley, 1999.

[20] L. Foner. Yenta: A multi-agent, referral-based matchmaking system. In *Proceedings of the First International Conference on Autonomous Agents (Agents'97)*, pages 301–307, Marina Del Rey, CA, 1997. ACM Press.

[21] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *The Interntional Journal of High Performance Computing Applications*, 15(3):200–222, 2001.

[22] J. Golbeck. Computer science: Weaving a web of trust. *Science*, 321(5896):1640–1641, September 2008.

[23] Z. Guessoum, M. Ziane, and N. Faci. Monitoring and organizational-level adaptation of multi-agent systems. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '04, pages 514–521, N. Y., Washington, DC, 2004. IEEE Computer Society.

[24] S. Hassard, A. Blandford, and A. Cox. Analogies in design decision-making. In *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology*, BCS-HCI '09, pages 140–148, Swinton, UK,, 2009. British Computer Society.

[25] R. Heeks and S. Bailur. Analyzing e-government research: Perspectives, philosophies, theories, methods, and practice. *Government Information Quarterly*, 24(2):243 – 265, 2007.

[26] R. Hermoso, H. Billhardt, and S. Ossowski. Role evolution in open multi-agent systems as an information source for trust. In *Autonomous Agents & Multiagent Systems/Agent Theories, Architectures, and Languages*, pages 217–224, 2010.

[27] T. Huynh, N. Jennings, and N. Shadbolt. Fire: An integrated trust and reputation model for open multi-agent systems. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI'04)*, pages 18–22, Valencia, Spain, 2004.

[28] T. Huynh, N. Jennings, and N. Shadbolt. Certified reputation: How an agent can trust a stranger. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems (AAMAS'06)*, pages 1217–1224, Hakodate, Japan, 2006.

[29] T. Huynh, N. Jennings, and N. Shadbolt. An integrated trust and reputation model for open multi-agent systems. *Journal of Autonomous Agents and Multi-Agent Systems*, 13(2):119–154, 2006.

[30] K. Islam. An e-market framework to determine the strength of business relationships between intelligent agents. In *Proceedings of the sixth Australasian conference on Data mining and analytics*, volume 70 of *AusDM '07*, pages 215–224, Darlinghurst, Australia, 2007. Australian Computer Society, Inc.

[31] N. Jennings. Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. *Artificial Intelligence*, 75:195–240, June 1995.

[32] N. Jennings and M. Wooldridge. On agent-based software engineering. *Artificial Intelligence*, 117:277–296, 2000.

[33] L. Jiang, P. Zhang, and H. Wang. Community e-government coordination work platform. In *Proceedings of the 2005 national conference on Digital government research*, dg.o '05, pages 249–250, Atlanta, GA, 2005. Digital Government Society of North America.

[34] B. Khosravifar, M. Gomrokchi, J. Bentahar, and P. Thiran. Maintenance-based trust for multi-agent systems. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '09, pages 1017–1024, Budapest, Hungary, 2009. International Foundation for Autonomous Agents and Multiagent Systems.

[35] B. Khosravifar, M. Gomrokchi, J. Bentahar, and P. Thiran. Maintenance-based trust for multi-agent systems. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, volume 2 of *AAMAS '09*, pages 1017–1024, Budapest, Hungary, 2009. International Foundation for Autonomous Agents and Multiagent Systems.

[36] V. Lesser. Cooperative multiagent systems: A personal view of the state of the art. *IEEE Transactions on Knowledge and Data Engineering*, 11:133–142, 1999.

[37] M. Ma and C. Meinel. A proposal for trust model: Independent trust intermediary service (itis). In *Proceedings of the IADIS International Conference WWW/Internet 2002 (ICWI'02)*, pages 785–790, Lisbon, Portugal, 2002.

[38] X. Mao, G. Wu, and H. Wang. Cooperation models for service oriented multi-agent system. In *Proceedings of the 2004 ACM symposium on Applied computing*, SAC '04, pages 510–511, Nicosia, Cyprus, 2004. ACM.

[39] S. Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, University of Stirling, Apr. 1994.

[40] MATLAB. http://www.mathworks.com/products/matlab, accessed on 16th. August 2011.

[41] N. C. Narendra, Umesh Bellur, S. K. Nandy, and K. Kalapriya. Functional and architectural adaptation in pervasive computing environments. In *Proceedings of the 3rd international workshop on Middleware for pervasive and ad-hoc computing (MPAC '05)*, MPAC '05, pages 1–7, Grenoble, France, 2005. ACM.

[42] OnSale. http://www.onsale.com, accessed on 16th. August 2011.

[43] S. Ramchurn, D. Huynh, and N. Jennings. Trust in multi-agent systems. *The Knowledge Engineering Review*, 19:1–25, 2004.

[44] F. Ren, M. Zhang, and K. Sim. Adaptive conceding strategies for automated trading agents in dynamic, open markets. *Decision Support Systems*, 48(2):331–341, 2009.

[45] J. Rubiera, J. Lopez, and J. Muro. A fuzzy model of reputation in multi-agent systems. In *Proceedings of the fifth international conference on Autonomous agents*, AGENTS '01, pages 25–26, Montreal, Canada, 2001. ACM.

[46] J. Sabater and C. Sierra. Regret: reputation in gregarious societies. In *Proceedings of the fifth international conference on Autonomous agents*, pages 194 – 195, Montreal, Canada, 2001.

[47] J. Sabater and C. Sierra. Review on computational trust and reputation models. *Artificial Intelligence Review*, 24:33–60, 2005. 10.1007/s10462-004-0041-5.

[48] D. Saha and A. Mukherjee. Pervasive computing: a paradigm for the 21st century. *IEEE Computer*, 36(3):25–31, 2003.

[49] M. Schillo, P. Funk, and M. Rovatsos. Using trust for detecting deceitful agents in artificial societies. *Applied Artificial Intelligence*, 14(8):825–848, 2000.

[50] J. Scott. *Social Network Analysis: A Handbook*. SAGE Publications, January 2000.

[51] S. Sen and N. Sajja. Robustness of reputation-based trust: Boolean case. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems (AAMAS'02)*, pages 288–293, Bologna, Italy, 2002.

[52] M. Sensoy and P. Yolum. A context-aware approach for service selection using ontologies. In *Proceedings of the 5th international joint conference on Autonomous agents and multiagent systems (AAMAS'06)*, Hakodate, Japan, 2006.

[53] M. Sharmin, S. Ahmed, and S. Ahamed. An adaptive lightweight trust reliant secure resource discovery for pervasive computing environments. In *Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM'06)*, pages 258–263, Pisa, Italy, 2006. IEEE Computer Society.

[54] M. Singh. Trust as dependence: A logical approach. In *Proceedings of the 10th International Conference on Autonomous Agents and Multia-agent Systems (AAMAS'11)*, Taipei, Taiwan, 2011.

[55] J. Tang, S. Seuken, and D. Parkes. Hybrid transitive trust mechanisms. In *Proceedings of the 9th international joint conference on Autonomous agents and multiagent systems (AAMAS'10)*, pages 233–240, Toronto. Canada, 2010.

[56] J. Tang and M. Zhang. An agent-based peer-to-peer grid computing architecture: convergence of grid and peer-to-peer computing. In *Proceedings of the 4th Australasian Symposium on Grid Computing and e-Research (AusGrid 2006)*, volume 54 of *ACSW Frontiers '06*, pages 33–39, Hobart, Australia, 2006. Australian Computer Society, Inc.

[57] W. Teacy, J. Patel, N. Jennings, and M. Luck. Travos: Trust and reputation in the context of inaccurate information sources. *Journal of Autofocus Agent and Multiagent Systems (JAAMAS)*, 12:183198, 2006.

[58] G. Vogiatzis, I. MacGillivray, and M. Chli. A probabilistic model for trust and reputation. In *Proceedings of the 9th International Conference on Autonomous*

Agents and Multiagent Systems (AAMAS'10), volume 1 of *AAMAS '10*, pages 225–232, Toronto Canada, 2010. International Foundation for Autonomous Agents and Multiagent Systems.

[59] J. Wang and H. Sun. A new evidential trust model for open communities. *Comput. Stand. Interfaces*, 31:994–1001, September 2009.

[60] C. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992. 10.1007/BF00992698.

[61] M. Yildiz. E-government research: Reviewing the literature, limitations, and ways forward. *Government Information Quarterly*, 24(3):646 – 665, 2007.

[62] B. Yu and M. Singh. An evidential model of distributed reputation management. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS' 02)*, pages 294–301, Bologna, Italy, 2002. ACM Press.

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

[64] G. Zacharia, A. Moukas, and P. Maes. Collaborative reputation mechanisms for electronic marketplaces. *Decision Support Systems*, 29:371–388, 2000.

[65] M. Zeleny. *Multiple criteria decision making*. McGraw-Hill series in quantitative methods for management. McGraw-Hill, 1982.

[66] M. Zhang, J. Tang, and J. Fulcher. *Advanced Computational Intelligence Paradigms: Theory and Applications, Studies in Computational Intelligence (SCI)*, volume 115, chapter Agent Based Grid Computing, pages 439–483. Springer-Verlag, 2008.