

2013

# A comprehensive approach to QoS-based single and composite service selection

Hao Gao

*University of Wollongong*

---

## Recommended Citation

Gao, Hao, A comprehensive approach to QoS-based single and composite service selection, Doctor of Philosophy thesis, School of Information Systems and Technology, University of Wollongong, 2013. <http://ro.uow.edu.au/theses/3986>

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: [research-pubs@uow.edu.au](mailto:research-pubs@uow.edu.au)

## **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.



# A comprehensive approach to QoS-based single and composite service selection

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

**Doctor of Philosophy**

from

UNIVERSITY OF WOLLONGONG

by

**Hao Gao**

School of Information System and Technology  
November 2013

© Copyright 2013

by

Hao Gao

All Rights Reserved

*Dedicated to*  
*My family and friends*

# 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.

---

Hao Gao  
November 14, 2013

# Abstract

---

Web services are implementation of Service-Oriented Architecture (SOA) which has been recognized as the next generation framework for the alliance of distributed applications over the Internet. During the last a few years, Service-Oriented Computing (SOC) has emerged to be a significantly important research area that has attracted increasing attention from both the research and industry communities. With the ever-increasing number of services published on the Internet, how to discover and invoke a desired service effectively and efficiently has become a challenging issue. More specifically, the service consumers have become more sophisticated and concern about Quality of Service (QoS) in addition to the functionalities of the services. Therefore, selecting web service with high quality that best satisfies consumer's requirements within a large pool of functionally equivalent service providers is the key factor for web service selection. After the consumers present their requirements, system support is needed to assist the consumers in selecting the desired services automatically or semi-automatically. Furthermore, as individual services with complementary functionalities could be composed together to form a value-added new service, i.e., a composite service, it is a challenging issue to select individual services for the composite service based on the QoS requirements.

The first aspect of the work presented in this thesis is QoS based single service selection. For the success of QoS based single service selection, some critical issues need to be addressed, such as how to express and represent the consumer's QoS requirements more accurately, how to compare and measure the difference between the consumer's requirements and the service's published information, and how to find the best service which not only meets the functional requirements but also fulfils the QoS requirements of the service consumer. The fundamental idea of this work is to measure the similarity between the service consumer's QoS requirements and the service's published QoS information. The service provider with the highest similarity score will be selected and recommended to the consumer. In this thesis,

the service request model is defined to represent the consumer's request in the non-functional aspect more accurately. A novel similarity computation approach that is based on the relative distance is proposed for calculating the similarity score. The categorical QoS values can be calculated in this approach, based on the set theory and semantic ontology. In addition, an attribute based service customization and selection model is proposed, which extends the attribute based access control policy language to express the additional effects beyond the accessibility. In this approach, different QoS values can be defined and applied to different user groups distinguished by their attributes.

The second aspect of the work presented in this thesis is QoS based composite service selection. In order to satisfy the consumer's requirements, how to select and compose multiple services together as a concrete service composition plan is still a challenge because finding an optimal solution is NP-hard. The key idea is to adopt the Genetic Algorithm (GA) to find a near-optimal concrete service composition plan. In this thesis, it is pointed out that the subjective opinion, which could be reflected by the trust, needs to be considered in the selection process. The formal service composition architecture for QoS based composite service selection is presented. The definition of the abstract service composition graph and the concrete service composition graph are provided. A novel Trust-oriented Genetic Algorithm (TOGA) is proposed with the trust evaluation method based on the subjective probability theory. The experiment result shows that the TOGA is able to find a near optimal plan effectively and efficiently. Furthermore, TOGA is extended to identify and support various business relationships among individual service providers. Four basic effect types of business rules among concrete service providers are defined, known as dependency, conflict, positive inference, and negative inference. A formal business rule description language is proposed to support the implementation of service composition. The Business Relationship Matrix (BRM) is captured to represent and handle various business relationships in the service composition.

The major contribution of this thesis is a comprehensive approach to QoS-based single and composite service selection. With this approach developed, it can largely facilitate consumers in selecting right services for their business needs.



# Acknowledgement

---

I am most grateful to my supervisor Jun Yan, for his support and guidance of this thesis. He has been providing invaluable suggestions and encouragement from the beginning of my research. This thesis would never been possible without his help and support.

I would like to thank my co-supervisor Professor Yi Mu for his continuous guidance in the process of conducting this research.

During my PhD study from July 2009 to present, I also received lots of advice and support from my colleagues and friends. The non-exhausted list includes: Dr. Jinguang Han, Dr. Fuchun Guo, Nan Li, Wei Wu, Prof. Xinyi Huang, Dr. Yong Yu, Yang Wang, Zhenfei Zhang, Yafu Ji, Jingtai Piao.

I am also grateful to University of Wollongong and all staff in the Faculty of Informatics, for their supports that help me to achieve my goals.

I would like to thank my parents and my wife Miao Zhou, for their patience and love. Without them, this work would never be possible.

# Publications

---

The following is a list of my research papers during my PhD study.

1. **Hao Gao**, Jun Yan, and Yi Mu, Dynamic Trust Model for Federated Identity Management. In *Proceedings of the 4th International Conference on Network and System Security (NSS)*, pp. 55-61, 2010.
2. **Hao Gao**, Jun Yan, and Yi Mu, Web Service Selection Based on Similarity Evaluation. In *Proceedings of IEEE International Conference on Services Computing (SCC)*, pp. 322-329, 2011.
3. **Hao Gao**, Jun Yan, and Yi Mu, Trust-oriented QoS-aware Composite Service Selection based on Genetic Algorithms. *Journal of Concurrency and Computation: Practice & Experience*, (accepted), in press.
4. **Hao Gao**, Jun Yan, and Yi Mu, Business Rule aware Composite Service Selection. *Journal of Service Oriented Computing and Applications*, (under review).
5. **Hao Gao**, Jun Yan, and Yi Mu, Attribute based Service Customization and Selection. *Journal of Systems and Software*, (under review).

# Contents

---

<b>Abstract</b>	<b>v</b>
<b>Acknowledgement</b>	<b>vii</b>
<b>Publications</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Research Motivation . . . . .	2
1.2 Research Problems and Contribution . . . . .	5
1.2.1 Research Problems and Contribution in Single Service Selection	5
1.2.2 Research Problems and Contribution in Composite Service Se-	
lection . . . . .	7
1.3 Organization of the Thesis . . . . .	9
<b>2 Literature Review and Requirements Analysis</b>	<b>11</b>
2.1 The Concept of Web Service and Web Service Composition . . . . .	11
2.1.1 Basic concept of web service . . . . .	11
2.1.2 Basic concept of web service composition . . . . .	15
2.2 Service Selection . . . . .	18
2.3 QoS Based Service Selection . . . . .	20
2.3.1 QoS Attributes . . . . .	20
2.3.2 QoS based single service selection . . . . .	22
2.3.3 QoS based Composite Service Selection . . . . .	26
2.4 Requirements Analysis . . . . .	31
2.4.1 Requirements in single service selection . . . . .	32
2.4.2 Requirements in composite service selection . . . . .	33
2.5 Summary . . . . .	34

<b>3</b>	<b>Overall System Architecture</b>	<b>35</b>
3.1	Overall Architecture . . . . .	35
3.2	Sub-architecture for single service selection . . . . .	37
3.3	Sub-architecture for composite service selection . . . . .	38
3.4	Summary . . . . .	40
<b>4</b>	<b>Similarity Based Service Selection</b>	<b>41</b>
4.1	Introduction . . . . .	41
4.2	Service Selection Model . . . . .	45
4.2.1	QoS Request Model . . . . .	46
4.2.2	Nfp Publication Model . . . . .	47
4.2.3	QoS based Service Selector . . . . .	50
4.3	The Matchmaking Phase . . . . .	50
4.3.1	Matchmaking on Dimensions matching . . . . .	51
4.3.2	Matchmaking on values matching . . . . .	51
4.4	Service Selecting Phase . . . . .	55
4.4.1	Selection Method . . . . .	55
4.4.2	Similarity Measurement for Categorical Values . . . . .	57
4.4.3	Similarity Measurement for Numerical Values . . . . .	58
4.5	Case Study . . . . .	59
4.6	Summary . . . . .	62
<b>5</b>	<b>Attribute Based Service Customization and Selection</b>	<b>63</b>
5.1	Introduction . . . . .	63
5.2	System Model and Data Flow . . . . .	64
5.3	Attribute Based Policy Language . . . . .	68
5.3.1	Attribute Based Access Control Structure . . . . .	68
5.3.2	Policy Language . . . . .	69
5.4	Attribute Based Policy Evaluating . . . . .	71
5.5	Customized Service Selection . . . . .	75
5.6	System Prototype and Demonstration . . . . .	76
5.7	Summary . . . . .	79
<b>6</b>	<b>Trust-oriented QoS-aware Composite Service Selection Based on Genetic Algorithm</b>	<b>80</b>
6.1	Introduction . . . . .	80

6.2	Service Composition Architecture . . . . .	82
6.2.1	Abstract and Concrete Component Service Invocation . . . . .	84
6.2.2	System Notation . . . . .	87
6.2.3	Problem Modeling . . . . .	88
6.2.4	Fitness Function . . . . .	89
6.3	Trust Evaluation for Concrete Service Composition Plans . . . . .	90
6.3.1	Trust value estimation for a concrete service . . . . .	91
6.3.2	Global trust value for a concrete service composition plan . . . . .	91
6.4	Genetic Algorithm for Service Selection . . . . .	92
6.4.1	Algorithm overview and assumptions . . . . .	92
6.4.2	Genetic algorithm for sequential invocation flow . . . . .	95
6.4.3	Genetic algorithm for general invocation flow . . . . .	97
6.5	Experiments . . . . .	101
6.5.1	Experiment settings . . . . .	101
6.5.2	Effectiveness of TOGA . . . . .	101
6.5.3	Efficiency of TOGA . . . . .	106
6.6	Summary . . . . .	107
<b>7</b>	<b>Business Rule aware Composite Service Selection</b>	<b>108</b>
7.1	Introduction . . . . .	108
7.2	A Motivating Case . . . . .	109
7.3	System Model . . . . .	111
7.4	Business Rules . . . . .	112
7.4.1	Business Rules Description . . . . .	112
7.4.2	Business Relationship Matrix . . . . .	115
7.5	Business Rule Based Composite Service Selection . . . . .	117
7.6	Case Study and Experiment . . . . .	119
7.7	Summary . . . . .	125
<b>8</b>	<b>Conclusion and Future Works</b>	<b>127</b>
8.1	Summary of Major Contribution . . . . .	127
8.2	Limitations . . . . .	130
8.3	Future work . . . . .	131
	<b>Bibliography</b>	<b>133</b>

# List of Tables

---

2.1	QoS attributes models . . . . .	21
4.1	Motivating case: the movie download services . . . . .	44
4.2	The non-functional property model for service A . . . . .	60
4.3	The matchmaking values in the case . . . . .	61
4.4	The result of similarity measurement in the case . . . . .	61
6.1	Notations . . . . .	88
6.2	QoS Aggregation Methods . . . . .	90
6.3	Experiment records . . . . .	104
6.4	Average $O(f_i)$ on different number of composition plans . . . . .	104
6.5	Average $O(f_i)$ on different mutation probability . . . . .	104
6.6	Average $O(f_i)$ on different population size . . . . .	105
7.1	The syntax of the business rule description language . . . . .	113
7.2	QoS information of each concrete service in the RA case . . . . .	120
7.3	Examples for Business Rules Description . . . . .	120
7.4	An Example of BRM . . . . .	121
7.5	Experiment records . . . . .	124

# List of Figures

---

2.1	Service-Oriented Architecture Model . . . . .	12
2.2	QoS-broker based web service architecture . . . . .	23
2.3	Approaches for QoS based Composite Service Selection . . . . .	27
3.1	Overall System Architecture . . . . .	36
3.2	Sub-Architecture for Single Service Selection . . . . .	37
3.3	Sub-Architecture for Composite Service Selection . . . . .	38
4.1	Similarity based Service Selection Model . . . . .	45
4.2	The Category Hierarchy . . . . .	54
5.1	The Service Customization and Selection Model . . . . .	65
5.2	The Data Flow Diagram . . . . .	66
5.3	Attribute Based Access Control Structure . . . . .	68
5.4	An example . . . . .	72
5.5	Different evaluating strategies . . . . .	73
5.6	Service Policy Input . . . . .	77
5.7	Service Customization View . . . . .	78
6.1	The System Model for Trust-Oriented Composite Service Selection . .	83
6.2	Abstract Service Composition Graph . . . . .	84
6.3	General Relations . . . . .	85
6.4	Two Concrete Service Composition Graphs . . . . .	86
6.5	Overview of Genetic Algorithm . . . . .	94
6.6	An example of chromosome . . . . .	97
6.7	An example of crossover and mutation operations . . . . .	98
6.8	A example of encoded chromosomes . . . . .	99
6.9	Histogram of $O(f_i)$ for each composition plan . . . . .	102

6.10	TOGA utility with different generations . . . . .	103
6.11	CPU time with different generations . . . . .	106
7.1	The Roadside Assistant Service . . . . .	110
7.2	Business Rule Driven Service Composition Model . . . . .	111
7.3	Crossover and mutation operations for RA case . . . . .	121
7.4	Genetic algorithm running without the consideration of the business rules . . . . .	123
7.5	Genetic algorithm running with the consideration of the business rules	124
7.6	Optimality of GAs . . . . .	125



# Chapter 1

---

## Introduction

Web services are implementation of Service-Oriented Architecture (SOA) which has been recognized as the next generation framework for the alliance of distributed applications over the Internet. During the last a few years, with the rapid development of the distributed systems and information technologies, Service-Oriented Computing (SOC) has emerged to be an significantly important research area that has attracted increasingly attentions from both the research and industry communities [Pap03]. Web service discovery process is to locate desired services, which are published in a service registry like a UDDI registry, to match consumers' requirements. Conceptually, the service-oriented applications are developed in a loosely-coupled way, while the consumers could discover and invoke their preferred services from the rich service pool [MPSF08]. With the ever-increasing number of services published on the Internet, finding desired web services is similar to looking for a needle in a haystack [GPST04]. In the SOC environment, how to discover a desired service effectively and efficiently has become a challenging issue. In particular, selecting web service with high quality while satisfying consumer's requirements within a large pool of functionally relevant service providers is the key factor for web service selection.

Nowadays, the service consumers become more concerned with the Quality of Service (QoS) in addition to the functionalities of the services. After the consumers present their requirements, the mechanism is needed to assist the consumers in selecting the desired services automatically or semi-automatically. Furthermore, as the web service technology has been developed during the last decade, various individual service oriented applications with complex functionalities could be composed together to form web service composition, or business processes. Another critical issue is how to select the individual services for the composition based on the QoS requirements. The selecting mechanisms for the single services and the composite

services are the focus of the work in this thesis. The remainder of this chapter elaborates the research motivation, the research issues and problems, the contribution of the thesis, and the organization of the thesis.

## 1.1 Research Motivation

Web service based technology takes a lot of benefits, such as easy and fast deployment, inter-operability, just-in-time integration, and reduced complexity by encapsulation. In particular, a service in this thesis is a web based software component which can be described by its functional and non-functional attributes. The functional attributes represent the functionality of a service to the consumers and generally described by the inputs, outputs, and the behaviors. This means the functional attributes represent what a service can accomplish. The non-functional attributes represent how well a service accomplishes that functionality, for example, the availability, cost, and response time, etc. Normally the non-functional aspect of a service is referred as QoS attributes. Although in some literatures QoS attributes are described as the subclass of the non-functional attributes, in this thesis the non-functional attributes are directly related to the QoS attributes, and we are focusing on the QoS based service measurement and selection.

Since the service selection is commonly driven by the non-functional attributes in addition to the functional aspect, it is critical to evaluate the QoS accurately and objectively. Considering the typical web service discovery scenario, the consumer is able to provide a definition of the desired service and pose a query to the repository of the advertised services, a comprehensive discovery mechanism is needed to help the consumer to select the “right” and “best” service automatically or semi-automatically.

To select a single service, a lot of work focused on collecting quality ratings while others presented a similarity assessment between a requested and a published service. However, for the success of the service selection process in the real world, there is a need to address some critical issues, such as how to express and represent the consumer’s QoS requirements more accurately; how to compare and measure the difference between the consumer’s requirements and the service’s published information; and how to find the best service which not only meets the functional requirements but also fulfills the QoS requirements of the service consumer.

In addition to the service’s published QoS information, the QoS customization

has become one of the key strategies for a service provider to differentiate itself from the competitors. There is a significant need for the service provider to offer customized QoS to different people or groups. For example, a hotel service may provide different QoS, cheaper cost for instance, to the VIP members. Therefore, the QoS customization mechanism is also needed to collect and provide the correct QoS information for the service selection.

For the composite service, the QoS based composite service selection has been considered and experienced in both industrial and academic fields for many years. To achieve this goal, a wide range of research areas has been covered. Examples include QoS modeling for identifying the QoS attributes of the services, QoS based specification languages for establishing and exchanging the QoS information, QoS capable frameworks for supporting QoS based service composition, and QoS monitoring for checking the compliance of the provided QoS values. According to these fundamental mechanisms, the most important issue is how to select an expected composite service with an optimal QoS that can be delivered to the consumers. Moreover, sometimes the consumer is likely to present the expectation on more than one QoS dimension. Therefore, the QoS based selecting mechanism for composite service is needed to compose the single services together and create the value-added composition with the best QoS results on all the dimensions to satisfy the consumer's overall requirements.

In the service composition process, a number of expected tasks can be composed together to form the abstract service composition. When each task is substituted with a concrete service deployment, a concrete service composition plan could be generated. The QoS based composite service selection process is gathering the concrete services together to fulfill the consumer's QoS requirements. Mathematically, this selection process is commonly modeled as a Multi-dimension Multi-choice Knapsack Problem (MMKP). In MMKP, the items which are classified into several object groups need to be filled into a knapsack with a set of constraints. The objective of the MMKP is to select the items from the object groups to maximize the total profit of the collected groups and fulfill the constraints of the knapsack [MR99]. In the same way, QoS based composite service selection is to pick one concrete service deployment for each task to create the concrete service composition plan to meet the defined QoS constraints and maximize a total utility value. It is a combinatorial optimization problem by its nature.

From the computation point of view, QoS based composite service selection

may face two main challenges. The first one is that the selection pool can be very large in scale. For example, if a composite service has 20 tasks and each task has 20 candidates, then there will be  $20^{20}$  possible concrete service composition plans. Finding an optimal solution from such a large number of options is computationally intractable. The second one is that many dependencies, conflicts, and QoS related impacts may exist among the concrete services. For example, there might be business to business contracts and relationships between the service candidates. Handling these impacts is necessary but difficult to guarantee the feasibility of the final optimal solution.

As the QoS based composite service selection problem is NP-hard [CPEV05b], the effective and efficient algorithms are required in the selection process. This research adopts Genetic Algorithm (GA) to address the QoS based composite service selection problem because GA is a powerful tool to solve combinatorial optimization problem and has been successfully applied in many other similar research domains, such as electronic circuit design [LC99], wireless sensor networks [JdW04], and computer-automated design [WDWY99], to solve the large scale and complex optimization problems. These successes motivate our choice of GA to deal with the QoS based composite service selection problem which has the similar characteristics in terms of large scale and problem complexity.

Furthermore, it is considered that the QoS information of a service reflects the objective opinion for the service selection as a number of literature were focused on; while the subjective opinion such as trust measurement is rarely taken into account. It is believed that a comprehensive service selection mechanism needs to consider both parts because the trust management of the service composition is an important research area for service selection. Trust is the measurement of belief on the willingness and ability to achieve some goals in a situation from one entity to another. Trust can be also reflected by the subjective probability if the trust value is presented in the range of  $[0, 1]$ . Meanwhile, in the composite service environment, trust management becomes more complex because of the different types of service invocation. Therefore, there are many research challenges in trust-oriented QoS based composite service selection.

## 1.2 Research Problems and Contribution

The research problems and the contribution of this thesis can be described in two main groups: QoS based single service selection and QoS based composite service selection. These two dimensional works are inter related, for example, the consumer's QoS request model is presented in the same way. The difference between two groups is based on the nature of the different service scenarios, for example, the business process is not considered in single service selection. In this research, the research problems are estimated by reviewing the related works. Then the overall system are modeled. After that, the findings are based on the experiments.

### 1.2.1 Research Problems and Contribution in Single Service Selection

One aspect of the work presented in this research dimension is the similarity based service selection (Chapter 4). The fundamental idea of this work is to measure the similarity between the service consumer's QoS requirements and the service's published QoS information. The service provider with the highest similarity score will be selected and recommended to the consumer.

A lot of work have been done for the similarity assessment between a request and a published service because the single similarity value between the published service and the request is efficient and easy to use in the QoS based service selection. Normally, the similarity score is normalized to a value in  $[0, 1]$  based on the metrics of multi dimensions. This similarity value reflects the absolute distance between the published QoS attribute and the request. Unfortunately, this kind of distance cannot depict the similarity very well under certain circumstances. For example, if there are two service provider A and B with their similarity value on a particular QoS dimension  $S_A = 0.8$  and  $S_B = 0.8$  (each of them is in the range of  $[0, 1]$ ), does it mean that both A and B have the same quality on this dimension? It is not true if A performs its most ability to achieve the value, while B has more tolerance. In this case, service B is better than A in terms of predicting th performance during the implementation. In order to take advantage of the tolerance of a service, the robust similarity assessment is required. In addition, a single value presentation of the consumer's QoS requirement without their expectation is incomprehensive for the request expression. Furthermore, a lot of selecting approaches were proposed to

calculate the similarity of numerical values, while the categorical data type is rarely mentioned. For example, we cannot estimate the similarity score between “student” and “professor”.

Focusing on the problems in similarity based service selection approach, our contribution in this thesis is summarized as follows:

- We introduce a novel similarity computation approach which is based on the relative distance instead of the absolute distance for the QoS based service selection. In our approach, the tolerance of a service is considered as an aspect for calculating the similarity score.
- We propose the service request model to represent the consumer’s request more accurately. The attribute of *Willingness* represents the tendencies of the consumer’s preferences on different QoS dimensions which are classified into five categories.
- We introduce the attribute of *Type* to present the service request and the published QoS information. The negotiable and non-negotiable values make the request and the publication more flexible.
- We propose a novel approach to calculate the similarity scores for the categorical values. The main idea is based on the set theory and semantic ontology. It is believed that the relationships between two categorical values could be depicted by a hierarchy from semantic web service ontology.

The other aspect of this research dimension is the attribute based service customization and selection (Chapter 5). As we mentioned that a lot of work focused on analyzing the difference or the similarity between the consumer’s request and the published QoS value, how to publish and locate the particular QoS value is an issue because many service providers have a need to offer customized QoS to different people. This means different groups of consumers, distinguished by their attributes, would be offered different QoS values. Furthermore, from service providers’ perspective, they prefer to expose their QoS values to the consumers who hold the accessibility rather than exposing to every entity. Therefore, the characteristics of the consumers, which are extracted from their attributes, need to be taken into account when selecting a service. We observed that Attribute Based Access Control (ABAC) approaches in which the access decisions are based on attributes of

the consumers and resources rather than the consumers' identifications, are helpful to customizing the QoS information. However, the effect of the traditional ABAC approach is defined as an alternative choice "permit" or "deny" without any QoS information. In order to present the customized QoS information as the additional effects, the access policy language needs to be extended.

Focusing on the problems in attribute based service customization and selection approach, our contribution is:

- We propose an attribute based service customization and selection model to address the need of acquiring the QoS information based on the consumer's unique attributes.
- We propose an extended attribute based access control policy language to publish the additional effects beyond the accessibility. In our approach, different QoS values are provided to different user groups which are distinguished by their attributes. And the service providers expose their QoS information to the consumers who hold the accessibility rather than every entities.

### 1.2.2 Research Problems and Contribution in Composite Service Selection

One aspect of the work presented in this research dimension is the trust-oriented QoS based composite service selection which is based on genetic algorithm (Chapter 6). As mentioned, QoS based composite service selection is a combinatorial optimization problem. The key idea in this research is to adopt the genetic algorithm to find a near optimal concrete service composition plan because GA is widely adopted in other combinatorial optimization research areas. In addition, we point out that the subjective opinion, which could be reflected by the trust, needs to be considered in the selection process.

In order to satisfy the consumer's requirements, how to select and compose the multiple services together as a concrete service composition plan is still a challenge because finding an optimal solution is NP-hard. There are some existing studies for QoS based composite service selection. However, the trust evaluation in the complex structure of composite services is rarely mentioned. How to measure the trustworthiness is an a problem because the different types of service invocation might be involved in service composition. When adopting the genetic algorithm, the

special coding scheme should be concerned to solve different problems. For example, the coding scheme for QoS based composite service selection should express not only the concrete service composition but also the information of composition strategies. Therefore, an advanced trust-oriented genetic algorithm needs to be designed for the QoS based composite service selection.

Focusing on the problems in the trust-oriented QoS based composite service selection, our contribution is illustrated as follows:

- We present a formal service composition architecture for QoS based composite service selection. We provide the definitions of the abstract service composition graph and the concrete service composition graph.
- Based on the subjective probability theory, we propose the global trust evaluation method as well as the local trust evaluation method for the concrete service composition plans.
- We introduce the Trust-oriented Genetic Algorithm (TOGA) to find a near optimal concrete service composition plan with QoS constraints. The experiments have been conducted on a composite service that includes a large number of concrete service composition plans, to compare the proposed TOGA with the exhaustive search method.

The other aspect of this research dimension is the business rule driven composite service selection (Chapter 7). In the service composition, the individual services from different providers can be used, composed, and coordinated in a loosely coupled manner. Due to the ever-increasing number of services published on the Internet, the relationships between the concrete services are becoming more complicated. Unfortunately, the QoS of a composite service is aggregated in a forthright manner because the concrete services in the composition are seen as atomic and independent. As more and more different business relationships are established between service providers, the dependencies and conflicts may exist, thus affecting some QoS dimensions, cost for instance, of the service composition.

We argue that the traditional work-flow based approaches for the service composition selection and binding exhibit an important shortcoming, as they hardly concern about the integration of business to business rules. Depending on different kinds of business rules, the service providers may not keep their QoS values consistently. For example, the hotel service could offer a special price to the customers



who are served by the co-operated travel agent. Therefore, QoS influences which are affected by business rules need to be investigated, especially for QoS based composite service selection.

Focusing on the problems in the business rule driven composite service selection, our contributions are illustrated as follows:

- We define four basic effect types of business rules among concrete service providers in the composition, such as dependency, conflict, positive inference, and negative inference on the particular QoS dimension.
- We propose a formal business rule description language to support the implementation of service composition. Once the business rules were captured, the Business Relationship Matrix (BRM) will be generated for the selection phase.
- We adopt the GA to find the near optimal solution. And the crossover and mutation policies are based on the different business relationships, which are generated from the business rules, among the concrete services.

## 1.3 Organization of the Thesis

This thesis addressed some critical problems in QoS based service selection in the service-oriented environments. The structure of this thesis is illustrated as follows.

Chapter 2 presents a comprehensive literature review of the QoS based service selection, based on which detailed requirements, analysis is given. Firstly, the basic concepts of web service and web service composition are presented as the background information. Secondly, some research dimensions of the service selection are illustrated. After that, the related works of the QoS based service selection, on which this thesis is focusing, is provided. Finally, the research requirements on both single service selection and composite service selection are analyzed.

Chapter 3 introduces the overall system architecture to perform the QoS based service selection. Differentiating single service to the composite service, the overall architecture is distinguished into two sub-architectures.

Chapter 4 introduces the similarity based service selection model to overcome the limitations of the existing QoS based service matchmaking and selection approaches. Firstly, the limitations and problems of the existing selection approaches are presented by a motivating case. Major properties which need to be considered for

the service selection are illustrated. Secondly, the service selection model, including the service request model, the service publication model, and the service mediator are introduced. After that, the similarity based service matchmaking phase and selection phase are described. Finally, a case study is analyzed.

Chapter 5 introduces the attribute based service customization and selection approach which takes the service consumer's attributes into account. In this chapter, the problems of the needs for service customization are presented. The policy language and the policy evaluation method are defined based on the attribute based access control structure. Furthermore, the system prototype is demonstrated.

Chapter 6 introduces a novel trust-oriented QoS based composite service selection approach based on genetic algorithm. In this chapter, the formal definition of the service composition architecture is described. For the consumer's subjective opinion, the trust evaluation of the concrete service composition plan is discussed. Then, the Trust-Oriented Genetic Algorithm (TOGA) is proposed to find a near-optimal service composition with QoS constraints. The effectiveness and efficiency of TOGA is discussed in the experiments.

Chapter 7 introduces the business rule aware composite service selection approach which deals with the dependencies and conflicts between the concrete services. In order to understand the needs to considering the business rules in the service composition, a motivating case of roadside assistant service center is presented. Then, a formal description language of the business rules is introduced. After that, the business relationship matrix is proposed to present four different types of impacts in the service composition. The extended genetic algorithm is discussed with applying the business relationships. To better understand the business rules between the concrete services, a case study is applied, and the experiment shows the utility of the business rules aware genetic algorithm.

Chapter 8 concludes our contribution, discusses the limitation of this research, and points out some directions for the future works.

# Chapter 2

---

## Literature Review and Requirements Analysis

### 2.1 The Concept of Web Service and Web Service Composition

#### 2.1.1 Basic concept of web service

In the SOC environment, the concept “web service” has been defined in many different ways. The most accepted definition was presented by the World Wide Web Consortium (W3C) which is the main international standards organization for the World Wide Web. They specified the web service as follows:

*“A web service is a software application identified by a URI, whose interfaces and bindings are capable of being defined, described, and discovered as XML artifacts. A web service supports direct interactions with other software agents using XML-based messages exchanged via Internet-based protocols”* [Alo04].

In this definition, they described how a web service works: it should be defined, described, and discovered. And the interaction environment should be XML based machine-to-machine inter-operable network. More specifically, a web service could be considered as a set of operations, where each operation acquires input values called parameters, and outputs values called the results of the invocation. The W3C also classified the web services into two major groups: REST(Representational State Transfer)-compliant web services which manipulate XML representations of the resources by a uniform set of “stateless” operations; and arbitrary web services which expose the arbitrary set of operations.

Web services can be demonstrated more clearly in the SOA model (see Figure 2.1). In the SOA model, there are three main entities: (i) service provider, (ii)

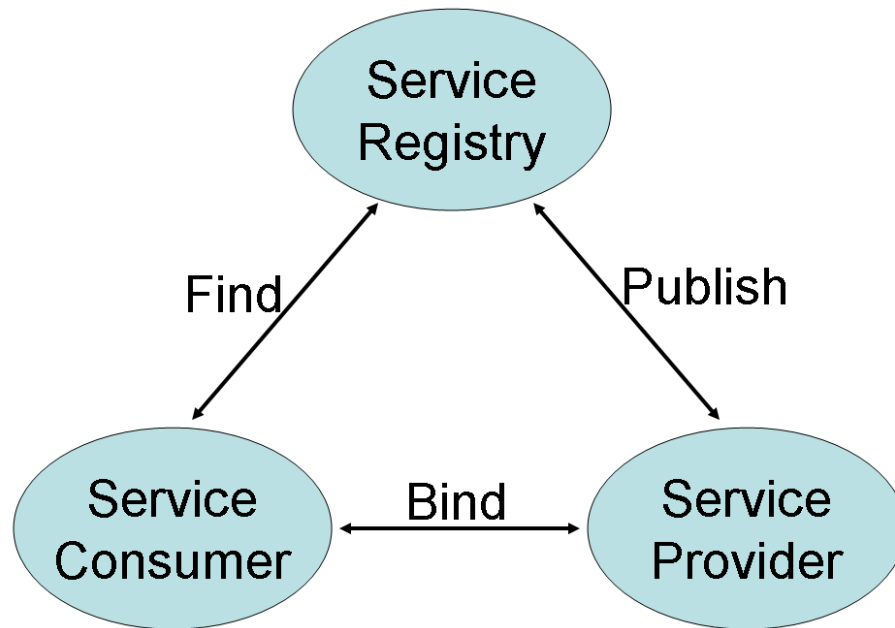


Figure 2.1: Service-Oriented Architecture Model

service consumer, and (iii) service registry. These entities are specified as follows:

- **Service Provider:** The service provider is the entity that provides the particular service to the consumers. A service provider could be a business entity, a public service agency, an academic institution, or a government agency, etc. A service provider may provide one or more services. In the real world, the situation is much more complicated. The identities of the providers may be known by consumers publicly or not. Furthermore, a service provider may or may not actually manage that service. For example, a service provider may outsource the task of actually operating the service to a third party. Obviously, the consumers may or may not be able to distinguish all the parts involved in delivering a given service. The difference between the service provider and the provided web service is out of the scope of this thesis. Thus, when we discuss a service provider, the web service is actually provided and managed by itself. The terms service provider, and provider are synonymous in this thesis.
- **Service Consumer:** A service consumer is any entity that invokes the web service. The service consumer could be a human user, a web application, or another web service.

- **Service Registry:** A service registry is an accessible and searchable engine that contains a collection of published web services with descriptions. A service registry has two basic specifications: (i) a definition of the information to provide about each service, and how to encode it; and (ii) a registry engine that answers the requests sent to the registry by service providers and service consumers. The implementation could be classified into two scopes: public or private. The public service registries are available through the Internet; thus any provider could advertise its capabilities by publishing; while the private registries are only accessible to a limited audience, for example, users of a company intranet.

These major entities in the SOA are considered as self-contained, self-describing, and platform-independent [Pap08]. To support the activities of service description, service discovery, and service invocation in the SOA model, there are a number of XML-based standards such as Web Service Description Language (WSDL) [CCMW04], the Universal Description Discovery and Integration (UDDI) standard [BCCC04], and the Simple Object Access Protocol (SOAP) [GHM<sup>+</sup>07].

- **WSDL:** It is an XML based description language for specifying: what the web service does, where the web service located, and how the web service invoked. In the specification, a WSDL document defines the web service by seven basic elements such as *Type*, *Message*, *Operation*, *Port Type*, *Binding*, *Port*, and *Service*.
- **UDDI:** It defines and contains the common published information about the web services. It is used to enable the worldwide businesses to publish services, discover each other, and define how the services interact over the Internet. The UDDI specifications consist of two main parts: (i) an XML schema for SOAP messages which defines *Business Entities*, *Business Services*, *Binding Templates* and *tModels*; and (ii) a description of the UDDI APIs which contain messages for interacting with the service registries.
- **SOAP:** It is a standard for sending messages, receiving messages, and making remote procedure calls by Request and Response types. It is encoded by XML and transported by HTTP, FTP, and SMTP protocols.

In addition to these three basic standards, a number of complementary standards

are proposed to support the web service implementations, such as WS-Policy, WS-Security, and WS-Trust [Alo04]. All of these basic and complementary standards work together to support the infrastructure of web service by paradigms which are called the web services technology stack [Alo04].

Based on the SOA infrastructure, the *Reference Architecture* (SOA-RA) extends the fundamental principles to provide the conceptual framework for describing and exhibiting how the core components are related to each other [Fie00]. There are a number of SOA-RAs that have been proposed for enterprise-level implementation by the standardized organizations, such as OpenGroup and OASIS [Ope09, ELMT09].

Currently, the web service technologies have been used by a mass of companies and organizations for implementing their applications. Compare to the traditional software applications, there are many advantages and benefits by using the web service-based applications [TP02]:

- **Easy and fast deployment:** the new web services can be developed and provided by the enterprises without investing and delaying the previous products. This means the new applications could be combined with the existing ones.
- **Inter-operability:** As web service standards are XML-based interface definition languages, any web service can interact with other web services. The web service based applications could be deployed and inter-operated with each other. Web services are platform and language independent, which means the developing environments do not need to be changed to produce or consume web services. In addition, since the legacy applications could be exposed as web services, the web service architecture enables inter-operability between the legacy applications and the web services.
- **Just-in-time integration:** The traditional systems may break down the bounded collaborations by any changes in the input/output of the subsystems or any new implementation of the subsystems because they are sensitive in coupling. Based on the notion of building applications by discovering and orchestrating available services, the web service systems promote significant decoupling and just-in-time integration of new implementations and applications.
- **Reduced complexity by encapsulation:** The system complexity has been

reduced for both the consumers and the designers because they only need to care about the provided behavior of the service instead of worrying about the implementation details.

### 2.1.2 Basic concept of web service composition

The service composition is an attractive concept of web services in a SOA environment. It implies that the individual web services can be combined and integrated together as a composition to create added value for the users. This creating process is called web service composition [ZBN<sup>+</sup>04]. For example, the travel planning web service can be created by composing a flight booking service, a car rental service, a travel insurance service, and an accommodation service.

There are many different approaches to web service composition. Nikola et al. [MM04] classified the approaches into five main groups: the algebraic process composition approach, the model checking-based approach [BFHS03], the web component-based approach [YP06], the semantic web-based approach [ABH<sup>+</sup>02], and the workflow-based approach [SJM06]. Among these five groups, the workflow-based approach is widely adopted in e-commerce, e-business, and e-science because it has been well standardized. In this thesis, all the research problems are investigated with the assumptions based on workflow-based approach.

A workflow is a sequence of tasks or steps to be executed. The notion of sequence does not necessarily mean one after the other, but the tasks could be organized in patterns such as parallel split, exclusive choice, and arbitrary cycles [vdAtHKB03]. Specifically, the abstract specification of the service composition is referred to the abstract workflow as the business process, while the result of a service selection is referred to the concrete workflow. In this thesis, our focus is the service selection process which is after creating the abstract workflow, the interfaces among the concrete services as well as the data flows and control flows are well defined. Thus, once the concrete service are selected to form a workflow, it is guaranteed that this workflow can be executed.

Depending on human involvement in the workflow-based composition process, there are three types of composition methods: manual composition, semi-automatic composition, and automatic composition.

- **Manual composition:** The business processes or workflows are programmed manually. There are a number of frameworks and tools to support this manual

implementation, such as WS-CAF and BPEL.

- **Semi-automatic composition:** The business processes require human user's decisions in the middle steps of the service composition. For example, when the user creates a business process, the available service candidates are discovered automatically and presented to the user [SHP03].
- **Automatic composition:** The business processes are produced without human activities. The composition process could be regarded as an AI planning problem, which is solved by situation calculus [MS02b] or hierarchical task network [WSH<sup>+</sup>04] planners.

In this thesis, we focus on the semi-automatic and automatic composition for the composite service selection. The problems that might occur in manual composition method will not be discussed in the following chapters.

There are two types of service compositions: Abstract service composition and Concrete service composition. The former contains the interaction protocols between the tasks without any details for execution, while the latter could be executed directly. The concrete service composition can be recognized as an instance of an abstract service composition. And they lead to two different types of binding approaches:

- **Static binding:** The service composition is with concrete services to provide a concrete service process directly.
- **Dynamic binding:** The service composition is with abstract services first, and then bind concrete services based on additional requirements.

Dynamic binding has been realized as the flexible and adaptable approach for selecting the concrete services based on the users' requests, especially in the context of QoS requirements [ZYZ10, AP07, YZL07].

For implementing the workflow-based service composition, a number of composition languages and standards have been proposed, such as WSFL [Ley01], WS-BPEL [SJM06], XLANG [Tha01], and WSCI [AAF<sup>+</sup>02]. Among these languages and standards, WS-BPEL has emerged as the principle standard which was also called BPEL4WS or BPEL. BPEL was proposed by IBM, Microsoft and BEA jointly as a specification for modeling the behavior of web services in the business process interaction. BPEL is an industry standard language and it has comprehensive semantics



to the complex requirements. The BPEL specification provides an XML-based grammar for describing the control logic. In addition, this grammar could be interpreted and executed by an orchestration engine which is controlled by one of the participating parties [Pel03]. Each BPEL program can be deployed as a composite service which can be invoked by internal or external clients.

The BPEL specification includes the following key concepts:

- **Variables:** BPEL variables are declared as XML elements within processes that store messages and hold state information during runtime. The Name of a variable has to be unique, and the Messages in variables could be sent and received from the partners.
- **Activities:** The implementation of a composite service contains two kinds of activities: basic activities and structured activities. Basic activity is an instruction for interacting with other web services, such as Invoke, Send, and Receive. Structured activity specifies the execution structures for the invoked web services, such as Sequence, If, While, Pick, Flow, etc. to support the sequential flow, the conditional flow, and the iterative flow.
- **Message correlation:** The message correlation mechanism is used to make sure that each process instance always sends and receives messages with the right instance of a service. Key variables of message exchanged, such as OrderID and UserID, are marked as correlation variables in the service interface to identify the process instance.
- **Fault handling:** A fault handler can make a lot of fault handling activities to find and catch specific faults, and the un-handled fault could be thrown to the parent scope.
- **Compensation handling:** The BPEL compensation handler is used to define a number of activities in the executions which have to be rollback. For example, a payment has to be refunded to the consumer if the seller cannot send a receipt.
- **Event handling:** The event handler specifies what need to be done when an event happens. There are two kinds of events: onMessage event, which leads to send and receive messages; and onAlarm event which has a specified time or a time interval.

Nowadays, there are more than 20 existing execution engines to support BPEL specifications, such as Oracle BPEL Process Manager [SJM06], IBM WebSphere Business Integration Server Foundation [Ren09], IBM BPWS4J [CDK02], and Microsoft BizTalk [Woo07].

## 2.2 Service Selection

The service selection process is to discover the right web services that match the service consumers' requests. Basically, service selection involves gathering the requirements from the consumers and the published information from the service providers who are registered, then matching the requirements with the service description. The requirements of the consumers normally include the description of the services and the quality of the service; while the service's published information contains the semantic profiles and QoS parameters. A number of service selection approaches and techniques have been proposed, such as selecting the service by optimization algorithm [BBMP07], integer linear programming [TAB08], service broker based architecture [SLS09], and negotiation based selection approaches. Based on the literature, three classes of selection approaches are identified:

- **Function based service selection approach:** The function based service selection approaches focus on selecting an appropriate service with the functional description from the service registry. These approaches rely on converting web services to semantic Web. Generally, the functional semantics are used to describe the web services in functions. In the semantic Web vision, several approaches such as WSDL-S, OWL-S, and WSMO have been proposed to exploit ontologies to semantically enhance the service description. Semantic service matchmaking and selection are then treated as a logic inference task [LH03]. Zaremski and Wing [ZW97] extended the signature-matchmaking to the semantic specification matchmaking in semantic Web. The semantical objects were specified by using the proposed protocol [CMK98]. LARKS [SSML02] enables heterogeneous service discovery which was supported by DAML-S.
- **Non-function based service selection approach:** In the SOC environment, a number of service providers may provide similar functionalities with

different non-functional property values. The non-function based service selection approaches focus on differentiating the services within the same functional presenting group. The non-functional properties are normally characterized by QoS and context which are taken into account during the selection process. The property of QoS may refer to response time, availability, reliability, cost etc.; and the property of context may refer to location, provider's detail, e-mail address, etc.. Zeng et al. [ZBD<sup>+</sup>03, ZBN<sup>+</sup>04] proposed two service selection methods driven by QoS. Their methods compute the quality ratings from the users of a service and then aggregate them using a simple arithmetic average to derive the QoS without considering the context. Xiao and Boutaba [XB05] proposed an autonomic service provision framework for providing QoS guarantees. Tian et al. [TGRS04] introduced WSQoS for QoS based web service selection and monitoring. They advocated the use of the service broker for the QoS based service selection.

- **Reputation based service selection approach:** The reputation based service selection approaches also called trust based selection approaches, use consumers' feedback or rating to identify good service providers. In these approaches, the service consumers prefer to choose a service that is more trustworthy with a high reputation score. Galizia et al. [GGD07] proposed a trust based methodology for service selection. Vu et al. [hVHA05] introduced a methodology for enabling QoS-based semantic web service discovery and selection with the trust and reputation management method.

This thesis explores non-function based service selection. In particular, we focus on the QoS aspect of the non-functional properties. For the functional aspect, our approaches assume that the service providers are classified into the same functional groups by the semantic presentation with different QoS values. Furthermore, it is considered that the QoS values indicate the objective opinion of the service consumers. The subjective opinion such as trust value is also needed to be taken into account for the comprehensive service selection method. The QoS based service selection approaches are further introduced in details in the following section.

## 2.3 QoS Based Service Selection

Every web service based application normally provides a particular functionality. In addition to the functional aspect, a service provider also exhibits some non-functional properties which are called QoS. For example, a service provider may charge the service consumer a certain amount of money, or a service provider may take a certain amount of time to respond to the service consumer. In this case, the service consumers' satisfaction and decisions might be determined and impacted by the amount of the cost, the probable length of the response time, and some other QoS attributes of the service provider. The service consumers' decisions will directly impact the success and failure of that web service based application and product in e-business and e-market scenarios. Therefore, when selecting a service, not only the functionality of the service provider, but also the QoS it provides, need to be considered. To achieve this, appropriate and effective methods are used to select the better quality services to fulfill consumers' expectation and satisfactions [CSM<sup>+</sup>04].

### 2.3.1 QoS Attributes

The first requirement is to identify the proper QoS attributes which need to be gathered and analyzed in the QoS based service selection process. In order to analyze the QoS attributes for web services, it is useful to rely on the classification which guides gathering the attributes. In the past few years, modeling QoS attributes for web services has been investigated by many researchers [Men02, Ran03, OEH02, Gro05].

The ontology based QoS attributes classification models have been proposed by many researchers. Dobson et al. [DLS05] proposed an ontology based classification of QoS attributes which was called *QoSOnt*. They identified two basic classes of QoS attributes, i.e., measurable attributes and unmeasurable attributes. All attributes have a physical quantity of a given unit, and conversion rates are used for converting values between units. They also developed a tool for service discovery and selection to identify the applicability of the *QoSOnt*. Papaioannou et al. [PTR06] proposed a QoS ontology language for web services. They defined *QoSParameter*, *Metric*, *QoSImpact*, *Type*, *Nature*, *Aggregated*, *Node* and *Relationship* for QoS attributes. All types of attributes are modeled as sub-classes of *QoSParameter*. Chaari et al. [Bad08] also proposed an ontology model for the non-functional properties. The

Zeng's Model [ZBN <sup>+</sup> 04]	Ran's Model [Ran03]	UML Profile [Gro05]
Throughput	Throughput	Throughput
Response Time	Response Time	Latency
Cost	Cost	Efficiency
Availability	Availability	Availability
Reliability	Reliability	Reliability
Reputation	Security	Security

Table 2.1: QoS attributes models

non-functional properties in their model are broadly categorized into QoS attributes and context attributes.

Unfortunately, there is no universal standard for QoS attributes modeling. Thus, different QoS attributes classifications for web services were put forward by different researchers and organizations. We illustrate three commonly accepted QoS models in Table 2.1.

Among these three comprehensive QoS attribute models, only Zeng's QoS model identified the calculation methods for the QoS based service composition which are the essential parts in the QoS based service selection. Currently, Zeng's QoS aggregation formula has been widely adopted in the research area of QoS based service composition and selection, especially when discussing QoS related optimization problems for the service composition. The QoS attributes in Zeng's model are described as follow:

- **Throughput:** Throughput represents the rate of successful delivery of requests during a time period over the network, it is usually expressed in bits per second or multiples of it.
- **Response Time:** Response time is also referred to as execution time. It measures the expected time delay in seconds between the request is sent and the result is received.
- **Cost:** Cost is also referred to as price. It presents the amount of money that the requester has to pay to the service provider for a certain invoked service.
- **Availability:** Availability represents the probability that the service is available during a period of time.
- **Reliability:** Reliability represents the probability that the request is responded and processed correctly and consistently by the service provider.

- **Reputation:** Reputation is a measure of the trustworthiness. It is normally based on requesters' direct or indirect experiences and rankings.

### 2.3.2 QoS based single service selection

Based on the specified QoS properties of the web services, how to discover these QoS criteria during the selection process is the critical issue. This entails the development of effective QoS based management for web service based systems. A lot of work has been done towards this aim by proposing the web service frameworks to support the QoS based service provision [MRG07, YK05, TGRS04, LKD<sup>+</sup>03, TPP02, Ran03, DSS<sup>+</sup>04, ZLC10].

The most popular QoS based web service frameworks is the QoS broker involved framework as depicted in Figure 2.2. In this framework, the QoS information, together with the general service description, needs to be published to the service registry. The service registry is now known as the QoS broker. When a service consumer makes a request, the service broker is able to provide the registered QoS information to the consumer.

To support this QoS based web service framework, a number of techniques and solutions have been proposed. Firstly, to enable the exchange of QoS information among the service providers, the service broker, and the service consumers, a few communication languages have been specified, such as IBM's Web Service Level Agreement (WSLA) [LKD<sup>+</sup>03], HP's Web Service Management Language (WSML) [MSD02], and the Web Service Offering Language (WSOL) introduced by Tomic et al. [TPP<sup>+</sup>05]. Secondly, to support QoS based service registering and discovering, the QoS brokers were developed to integrate with the role of UDDI repository [Ran03, DSL04]. In Ran's model [Ran03], the QoS broker was defined by extending the UDDI repository of the SOA elements; while Dewekar's model developed the QoS broker which was separated from the UDDI repository [DSL04]. The responsibility of the broker is to publish and discover the QoS information when the interactions occurred. Thirdly, to ensure the published QoS values would be offered properly during the run time, QoS monitoring models have been proposed in Tian's model [TGRS04].

Within the QoS based web service framework, various QoS based service selection techniques have been discussed in literature. We classified these QoS based service selection approaches into six groups: Service Adaptation Evaluation (SAE)

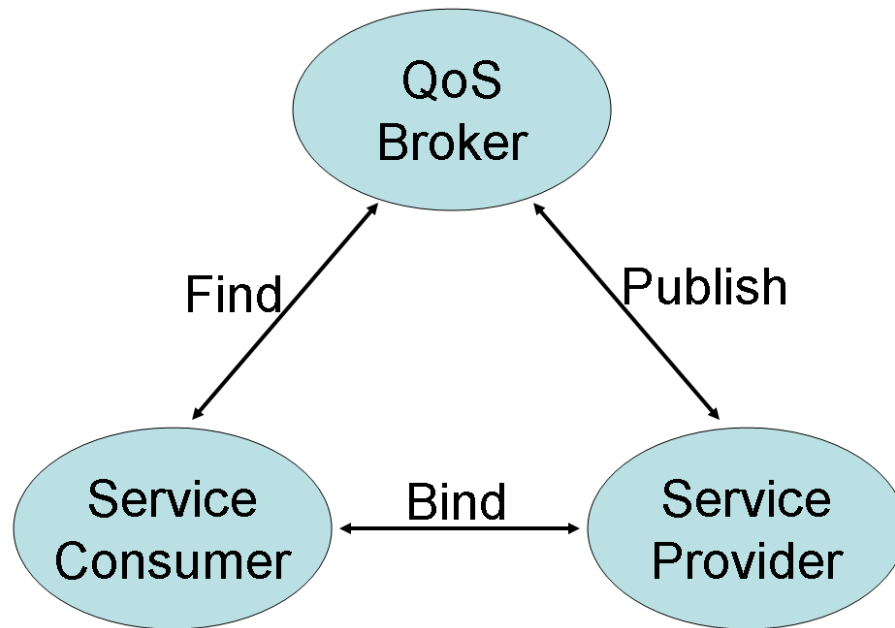


Figure 2.2: QoS-broker based web service architecture

based service selection approach, QoS normalization based service selection approach, Fuzzy logic programming based service selection approach, QoS constraints based service selection approach, Ordered Weighted Averaging (OWA) operator based service selection approach, and similarity based service selection approach. The related work of these different approaches are illustrated as follow.

- Service Adaptation Evaluation (SAE) based service selection approach:** Baopeng et al. [ZSX06] proposed a SAE algorithm to handle the QoS based service selection problem by using the hierarchy policy to capture the users' expectations. Depending on the different layers of the hierarchy, the policy driven service selection method enables the selection mechanism semantic-aware and QoS-aware.
- QoS normalization based service selection approach:** Yutu et al. [LNZ04a] proposed a dynamic framework to evaluate the QoS values of the web services. The extensible QoS model, preference-oriented service ranking, and the fair QoS computation methodology were introduced. They considered three kinds of QoS criteria such as execution time, price and reputation. In order to rank

the web service by the QoS information, the normalization technique was proposed. Li et al. [LRZ11] normalized the multi-dimension QoS parameters and mapped each dimension parameter into the same interval in order to provide references for selecting the suitable service. The normalization method is to allow a uniform measurement of QoS independent of units, provide a uniform index to represent QoS for each service provider, and set a QoS threshold.

- **Fuzzy logic programming based service selection approach:** Li et al. [LHH<sup>+</sup>11] introduced a synthetic service selection method (SSHFAQ) to rank the services. In their approach, QoS metrics are presented by linguistic, interval number and fuzzy number are transformed into precise number. Ping et al. [WCLH06] proposed a QoS based service selection model by Fuzzy logic programming technique to identify the dissimilarity between the service providers and the service consumers. The preferences of the consumers are considered in the selection approach to find the most suitable services. The fuzzy group consensus aware service selection algorithm was proposed based on Linear Programming techniques for multidimensional analysis of preferences to locate the optimal solution. The algorithm was operated on the fuzzy numbers which include the representation for Triangular fuzzy number, fuzzy arithmetic operations for addition, subtraction, multiplication and division.
- **QoS constraints based service selection approach:** Tao et al. [YK05] proposed the QoS based service selection model by the combinatorial model and the graph model. Four efficient algorithms were designed to ensure that the selected services were satisfied to the users' QoS constraints. One of them was designing the QoS constraints for the service selection, and the heuristic algorithms were applied to find the near optimal solutions. In addition, they considered the composite service selection by different invocation structures such as sequential, parallel, conditional ,and loops.
- **Ordered Weighted Averaging (OWA) operator based service selection approach:** Hong et al. [YRM08] proposed a QoS based service selection approach by modifying the Logic Scoring Preference (LSP) with Ordered Weighted Averaging (OWA) operators to automate the service selection process. In order to make the selection process automatically, the ranking problem was transformed into OWA problem to calculate the LSP degree.



- **Similarity based service selection approach:** Similarity measurements have been widely adopted in the information systems, databases management, software engineering and artificial intelligent. In the web service selection research area, Wang et al. [WS03] proposed a similarity measurement between the user's requirement and the available services through a semantic information retrieval method and a structure matching method. Maedche and Staab [MS01, MS02a] provided the multiple-phase cross-evaluation to compute the similarity between two QoS ontologies. Deora et al. [DSGF04, DSGF06a] introduced a range based similarity assessment of expectations. Liu et al. [DZC09] took user's expectation into account when assessing reputation and only the assessment with similar expectation has impact on the reputation rating for services. The matching algorithms in [Car06] and [DAS07] assessed the similarity between requested and offered inputs and outputs by comparing classes in an associated domain ontology.

Most industry standards for the web service description and discovery, such as WSDL, UDDI, provide limited selection capabilities. They are focusing on the structure description, the messages exchanging, the interface definition, and the key-word based searching. In the semantic web vision, several approaches have been proposed exploiting ontologies to semantically enhance the service description. Unfortunately, all the standards mentioned above omit QoS aspects, they do not provide solutions for service discovery and selection to ensure the satisfaction of the user's requirements.

Although a lot of efforts have been devoted to QoS based web service selection as illustrated above, it is critical to evaluate the services accurately and objectively. Especially in the similarity based service selection approach, the similarity evaluation methods in those work are based on the absolute distance between two values. It could be argued that the relative distance could reflect the quality more accurately and tolerantly. In this thesis, we analyse the user's expectancy more specifically, instead of binary(positive/negative) choices. In addition, they hardly considered the similarity measurement for categorical values. The details of our similarity based service selection approach will be presented in Chapter 4. Moreover, most of work above assume that a service provides the same QoS to everybody. They ignored the security issue and assumed all the users in their approaches can access the services. Compared with their work, our similarity score is computed between

the user's requirements and the customized QoS which is based on user's attributes. In Chapter 5, the service providers expose the QoS information to the users who are authenticated to access only.

### 2.3.3 QoS based Composite Service Selection

For the composite service, once the business process is defined, the composite service needs to select the concrete services to perform the tasks. According to the QoS based service configuration and discovery, the QoS based composite service selection is to compose the available satisfied services together and acquire the aggregated QoS values to fulfill the service consumers' QoS requirements. To achieve this goal, the optimization method needs to be designed to produce the best concrete service composition plan. Generally, the QoS based composite service selection problem could be formulated as the combinatorial optimization problem. In the SOC environment, the QoS based composite service selection has the following features:

- The objective is complicated. The objective could be represented to maximize or minimize one QoS dimension for the composite service, or maximize or minimize multiple QoS dimensions simultaneously, or maximize or minimize multiple QoS dimensions for multiple composite services simultaneously.
- The number of possible service composition plan is usually large. For the QoS based composite services, the number of feasible composite solutions normally depends on the size of the business process of the composite service in terms of the number of tasks, as well as the number of concrete service candidates to accomplish each task.
- Various constraints may exist. On one hand, the QoS constraints are presented by the service consumers including the QoS concern on each task and the aggregated QoS constraints for the whole composition plan. On the other hand, the inter-service dependencies and conflicts may be represented as the service to service constraints within a composition.

To address these features, a number of different approaches have been proposed for the QoS based composite service selection in literature. The overview of widely discussed approaches rather than the exhaustive search method is presented in Figure 2.3. The popular optimization approaches for QoS based composite service

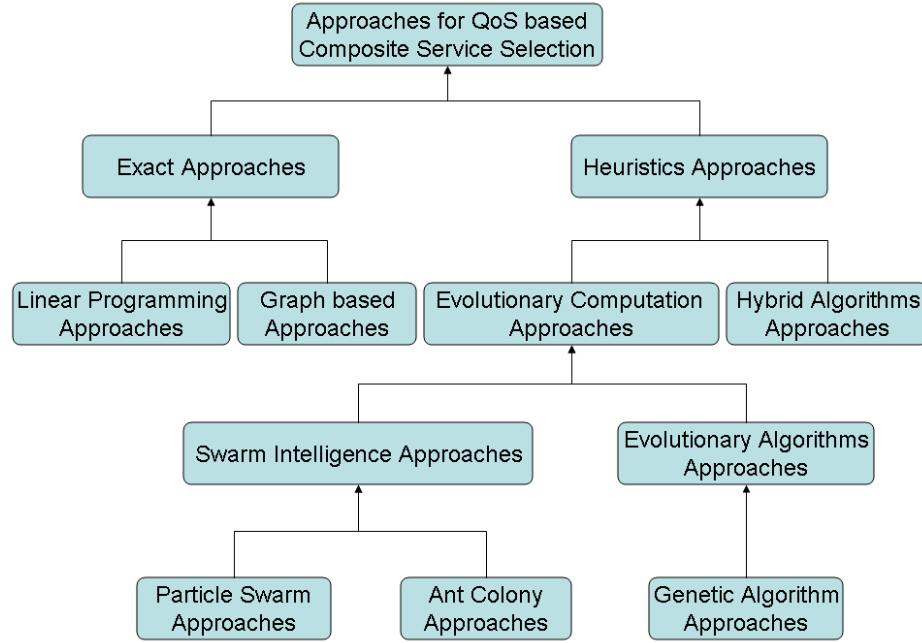


Figure 2.3: Approaches for QoS based Composite Service Selection

selection, such as linear programming approaches, graph based approaches, network topology based approaches, game theory based approaches, swarm intelligence based approaches, and genetic algorithm based approaches, are illustrated as follows.

- Linear Programming Approaches:** The goal of QoS based composite service selection is to maximize or minimize the given utility function while fulfill the QoS requirements and constraints. This problem could be modeled as a Mixed Integer Programming (MIP) problem when the objectives and constraints are linear. In Zeng's model [ZBN<sup>+</sup>04], the composite service selection problem was solved by the Linear Programming method. They presented the concrete execution plan by the Directed Acyclic Graph (DAG), and each concrete service was represented with a quality vector. For each feasible concrete service composition plan, the aggregated QoS value was computed and the Simple Additive Weighting (SAW) was used to capture the optimal plan. However, this approach required that all possible concrete composition plans need to be generated. Therefore, the computation costs are very high when the computation pool is large for the global planning. In addition, their methods compute the quality ratings from the consumers and aggregate them to

derive the global value without considering the context. In [AR09], the authors adopted MIP techniques to combine the global and local service selection algorithms in order to take advantage of the better QoS results globally. In [ZBD<sup>+</sup>03], the authors decomposed the MIP problem into two phases. In phase one, the global QoS constraints are decomposed into local QoS constraints which are sent to the service brokers. Then in phase two, the locally best concrete service which meets the local constraints is selected. The most significant improvement of this two phase solution is the reduction of the computation time, and the results are near optimal. To solve the IP problems, Branch and Bound (BB) algorithms are also usually used. Yu et al. [YL04] proposed BB-based algorithms for composite service selection while considering the sequential, parallel and conditional invocation types. Furthermore, the authors also proposed a heuristic approach in order to reduce the computation time.

- **Graph based Approaches:** The abstract service composition as well as the concrete service composition could be represented as the Directed Acyclic Graph. Yu et al. [YZL07] proposed a converting model between a function graph (abstract service composition) and a service candidate graph (concrete service composition). Based on the DAGs, the QoS based composite service selection problem is mapped to a Multi-Constraint Optimal Path (MCOP) problem because each edge in a DAG associates with a set of QoS attributes and a utility value. Li et al. [LWL10] proposed a service invocation graph and a service execution graph for the service composition. They introduced the trust aggregation method to capture the global trust degree, and adopted the Monte Carlo Method Based Algorithm (MCBA) to find the near optimal composition plan for the trust-oriented composite service selection with QoS constraints.
- **Network Topology based Approaches:** Jin et al. [JN04] focused on the composite service selection problem in large networks. They introduced the QoS service routing which is following the network nomenclature. They defined a service path of a service composition as an end-to-end network path with a given sequence. In addition, they distinguished the service unicast for one-to-one routing scenarios and the service multicast for one-to-many routing scenarios. They put an emphasis on the scalability of service routing by

adopting a distributed routing approach, and they mentioned the current web based network is too big to monitor and plan by the centralized management approaches.

- **Game Theory based Approaches:** Game theory models and approaches have been proposed in [El05, SF07] to deal with the QoS based composite service selection problem. In [El05], they introduced a game-theoretic setting for the service providers, and formulated the composite service selection problem as a multi-dimensional single-round auction which can be repeated many times. In their work, three different QoS dimensions were considered, such as availability, reliability, and price. The service consumers could select a bid, that the service providers submitted for a same task, to a utility function. So the selection approach was modeled as an auction and the winner determination algorithm was proposed to fulfill the consumers' requirements. In [SF07], a game-theoretic approach has also been proposed to deal with selecting the sensitive services, and the authors pointed out that the response time of a service depends not only on the service provider's performance, but also on the number of users' requests. The authors also proposed a cooperative mixed strategy for service selection. In that selection strategy, the service consumers select the service promising the minimum execution time only with a certain probability. Cheung et al. [CLTW04] proposed a bidding approach for the service selection. In their model, the service broker selects the concrete service provider based on the bidding and the probability distribution.
- **Swarm Intelligence based Approaches:** Swarm Intelligence is a optimization technique generated from Artificial Intelligence (AI). It is inspired by the behavior of animals, such as fish and ants [HBM08]. For the Swarm Intelligence, there are two main kinds of techniques: Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO). A lot of work applied PSO to deal with QoS based composite service selection problem [JXX<sup>+</sup>10, SWY10, XCL<sup>+</sup>09]. In [JXX<sup>+</sup>10], the QoS attributes such as reliability, cost, time, and reputation were considered in the PSO based optimization algorithm. They compared it to a genetic algorithm in the experiments. However, the details of the parameters of the genetic algorithm was not shown. In [SWY10], Sun et al. proposed a PSO based local optimization approach to deal with the fuzzy logic based QoS constraints. Wang et al. [WMC10] proposed an ACO based

approach to address the QoS based composite service selection problem. The workflow of the composition was modeled as the path from the ant nest to the food place. And the optimal composite service was considered as the shortest path.

- Genetic Algorithm based Approaches:** The Genetic Algorithm (GA) is a search heuristic that mimics the ideas of natural selection and evolution. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques basically inspired by *selection*, *crossover* and *mutation*. GA is one of the most suitable method for the QoS based composite service selection which is NP-hard [SP94a]. Canfora et al. [CPEV05a] seem to be the first group to adopt GA for solving QoS based composite service selection problem. They defined the QoS aggregation methods and a single-objective fitness function. The distance based penalty function was introduced to handle the QoS constraints. They estimated that the GA performs better than Integer Programming approaches when the service pool is large. Jaeger et al. [MG07] extended Canfora's work by adopting the same QoS attributes and analyzing the impact of different number of generations, initial populations, mutation settings, and penalty factors. They shown a good experimental results of GA which compares to the exhaustive search algorithm. Li et al. proposed two GAs with modifications for the QoS based composite service selection [LCOZ10, LYO<sup>+</sup>10]. In the former work, they applied the Strength Pareto Evolutionary Algorithm 2 (SPEA2) to solve the optimization problem. In the later work, they applied the Nondominated Sorting Genetic Algorithm (NSGA) to solve the same problem. Earlier et al. [DPJ05] adopted NSGA-2 for the composite service selection and they evaluated the quality of the results as well as the scalability of their approach. Liu et al. [LLJ<sup>+</sup>05] proposed a multi-objective GA, which was named Global Optimization of Dynamic Web Service Selection (GODSS), to deal with the composite service selection problem. They modeled the selection problem as a Multi-Constraint Multi-Objective Optimal Path (MMOP) problem.

In many QoS based composite service selection scenarios in the real world, the service consumers do not have to find the optimum plan. Furthermore, searching an optimum plan is time consuming because it is a multi-objective optimization problem and it is NP-hard. Therefore, among the various approaches for the composite

service selection, we adopt the Genetic Algorithm to find a near-optimal composite service plan efficiently in this thesis, because GAs are usually used for producing acceptable solutions to optimization problems when their search space cannot be traversed efficiently by traditional optimization methods, such as gradient descent methods, or heuristic-based methods. In addition, genetic algorithms have different constraint handling techniques, such as penalty function-based methods, repairing methods, and hybrid methods.

As we analyzed above, a number of QoS-aware composite service selection mechanisms have been proposed, aiming at QoS improvement in composite services. However, to the best of our knowledge, none of them considers the aspect of trust, which is valuable information in many applications. Trust is an important issue in the SOC environment especially in e-commerce. A trust management system can provide significant information to the service consumers and prevent some traditional attacks. A trust evaluation system can also reduce the potential risks to the service consumers and avoid a number of malicious service providers. Besides the objective factors which depend on the conformance between actual QoS and advertised QoS, the subjective factors are very important in a SOC environment. Compared with the existing work, a comprehensive mechanism, which could evaluate both the subjective aspect as trust expression and the objective aspect as QoS detection, will be introduced in Chapter 6.

Indeed, most of the existing work aims at QoS improvement in composite service selection. For example, Zeng et al. presented two service selection approaches by QoS-aware middleware-supporting for the local optimization and global planning; Aggarwal et al. were focusing on the QoS constraints in global view; and Xiao proposed an autonomic service provision framework for providing QoS guarantees over domains. However, the business rules among the service providers are ignored. As different business relationships are established between service providers, the QoS of a concrete service would be changed. Therefore, in Chapter 7, we take the effects of the business rules into account for the QoS changing of the composite service.

## 2.4 Requirements Analysis

According to the literature review of the QoS based service selection, a lot of work has been done for both single service selection and composite service selection. However, some research challenges remain open and need to be further investigated.

### 2.4.1 Requirements in single service selection

For QoS based single service selection, we adopt the similarity based service selection approach because it is easy to apply on computing the distance between a pair of objects. Especially in the data mining and recommendation system research areas, the similarity based measurement has been widely adopt. Generally, the similarity measurements capture the degree of match as the similarity score between a service request and a published service. This process involves two main steps: (i) filtering the matched service providers as the candidates from a large number of potential service providers in the same functional category, (ii) selecting a criterion for assessing the similarity of service parameters and aggregating individual parameter scores to obtain the overall degree. For example, Deora et al. [DSG06, DSGF06b] introduced a range based similarity assessment of expectations. However, the approach would be strict because of the single attribute matching. And other similarity based service selection approaches, such as [WS03, MS01, MS02a, Car06, DAS07], are based on the absolute distance between the requests and the published QoS values. We argue that the absolute distance between two values cannot reflect the results accurately and tolerantly. In addition, they hardly considered the similarity measurement for categorical QoS attributes. Therefore, it is believed that a comprehensive similarity based service selection approach need to consider following requirements:

- *Requirement 1:* The selecting approach should estimate the service consumers' expectation accurately. A simple quality rating without consuser's expectation is not comprehensive.
- *Requirement 2:* The selecting approach should flexibly estimate the similarity between a request and a published service. With the strict definition, sometimes, there is hardly a service provider satisfies the consumer's requirements perfectly. Furthermore, the single criterion may lead to information loss that significantly affects the accuracy of the results. For example, a service that has only one bad matching parameter may be excluded from the result, even though they are potentially good alternatives.
- *Requirement 3:* The selecting approach should be able to estimate the similarity between any data types. A lot of selecting approaches were proposed to calculate the similarity of numerical/continuous values, while the categorical data type is rarely mentioned.



Moreover, when applying the similarity measurement, how to collect the “right” QoS values from the service providers is an issue because many service providers have a need to offer customized QoS to different people or groups. Different groups of consumers, who are distinguished by their attributes, would be offered the different QoS values. Furthermore, the service providers might not wish to expose their QoS values to the consumers who are not able to access. Based on these problems, the further research requirements are presented as follows:

- *Requirement 4*: The selecting approach should support the customized QoS values according to the characteristics of the consumers. This means the user’s attributes need to be taken into account for service selection.
- *Requirement 5*: The selecting approach should estimate the accessibility of the service consumer for the particular services because the service providers may not expose the QoS information to the users who have no rights to access their services.

## 2.4.2 Requirements in composite service selection

For QoS based composite service selection, there are a lot of optimization approaches to finding the optimal or near optimal solutions. In this thesis, we adopt Genetic Algorithm to find a near optimal concrete service composition plan because GA is a powerful tool to solve the combinatorial optimization problem and it has been successfully applied in many domains to address complex and large scale optimization problems. In particular, the efficiency and the global astringency of GA are determined by the coding scheme of chromosomes and the evolution operators. The special coding scheme should be concerned to solve different problems. In addition, previous GA based composite service selection approaches only concerned with the published QoS information of the service providers, which we called objective opinion in the selection. The subjective opinion such as trustworthiness has rarely been considered. Furthermore, among the concrete service providers in the composition, the business relationships, which would affect some QoS dimensions significantly, have not been further investigated. For example, the business relationships would generate the service dependencies and conflicts to affect the QoS values. Therefore, to address these issues, several requirements need to be meet, as presented as follows:

- *Requirement 6:* The composite service selection approach should estimate the subjective opinion of the service consumers, such as trust, rather than the objective QoS values. Particularly, the trust estimating method for different invocation types in the service composition is needed to reflect the consumer's subjective aspect.
- *Requirement 7:* The coding scheme of the GA based composite service selection approach should express not only the services composition but also the information of different service composition strategies.
- *Requirement 8:* The composite service selection approach should identify and work with the business relationships among the concrete service providers. The service publication and discovery of the business relationship need to be defined. How to apply these business relationships in the GA based service selection needs to be further investigated.

## 2.5 Summary

In this chapter, we presented the research background, related work, and the research requirements on QoS based service selection. Firstly, the basic concepts of web service and composite web service were introduced. Then the classification of the service selection was identified. We claimed that the non-function based service selection, especially on the QoS aspect, is the main focus of this thesis. After that, the literatures of QoS based service selection were reviewed into two groups: the single service selection and the composite service selection. Finally, the research challenges and requirements were analyzed. This comprehensive literature review and requirement analysis laid solid foundation for our research.

# Chapter 3

---

## Overall System Architecture

In this chapter, the overall system architecture is introduced to perform the QoS based service selection. As mentioned in Chapter 2, the broker based web service architecture has been widely adopted for service discovery and selection. In our system architecture, we adopt the QoS supported UDDI repository as the service registry where the service providers could publish their QoS information directly, and this kind of service registry is the essential basis of this thesis. Furthermore, the overall architecture needs to be adaptable and extensible because service selection may face various requirements in different environments. Our overall system architecture and sub-architectures are proposed to address these requirements.

### 3.1 Overall Architecture

The overall architecture that is presented in Figure 3.1 extends the standard SOA structure with QoS support for QoS based service selection. The architecture contains three main participating components: the service consumer, the service providers, the extended QoS-supported service broker. The service consumer could present his functional and non-functional requirements to the service broker. Service providers need to register and publish their service description and QoS information in the service registry. The extended modules of the service broker are responsible for collecting the requirement of the consumer, selecting the most suitable service, and calculating the best composite service solution to fulfill the consumer's requirement. For the functional aspect, the service broker clusters the available services by acquiring the service description from the service registry, the function ontology might be applied if the services are semantically described. The QoS aspect is also analyzed by comparing the published QoS information and the consumer's request which are collected from the service registry and the request handler, respectively.

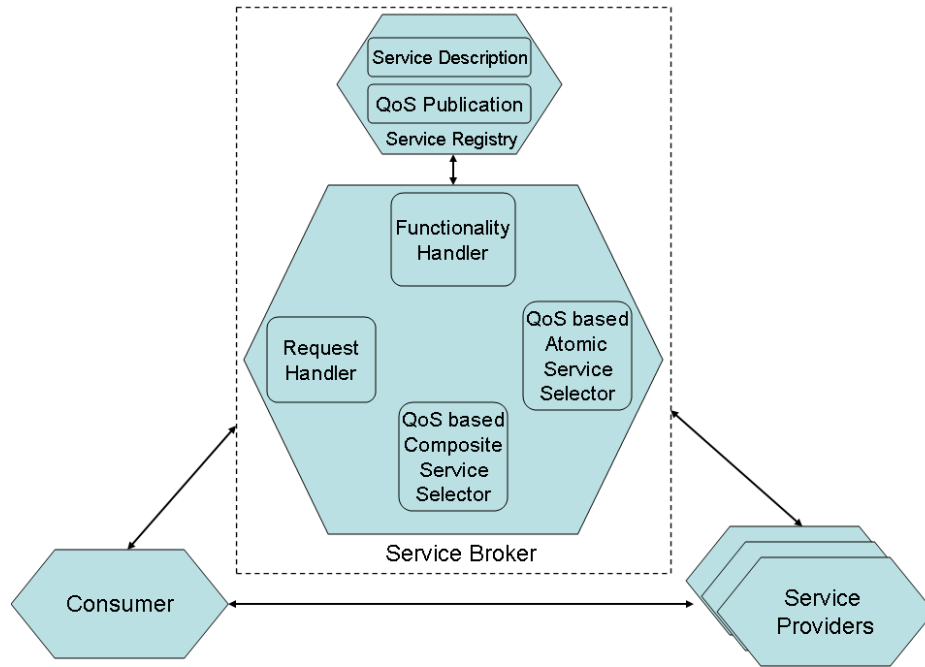


Figure 3.1: Overall System Architecture

The major components of service broker are illustrated as follows.

- **Service Registry:** It is a traditional QoS supported UDDI service registry. Before the services are discovered, the service providers need to publish their functional and QoS information to the service registry.
- **Request Handler:** It is responsible for collecting and managing the consumer's information, including the functional requirements, QoS requirements, and the consumers' attributes. The functional and QoS requirements will be sent to the functionality handler and QoS based service selector, respectively, and the consumers' attributes are used to identify the accessibility and acquire the customized QoS information.
- **Functionality Handler:** Based on the consumer's functional requirements, the functionality handler is responsible for clustering the functionally equivalent services by the service description.
- **QoS based Single Service Selector:** It is responsible for selecting the best single service to fulfill the service consumer's QoS requirements.

- **QoS based Composite Service Selector:** It is responsible for composing the single service candidates together, and applying an optimization method to find a suitable solution.

In addition to the service selection, the service broker could operate other functions, such as QoS negotiation, service monitoring, and service adaptation.

Differentiating the single service to the composite service, the overall architecture could be distinguished into two sub-architectures which are presented in the following two sections, respectively.

## 3.2 Sub-architecture for single service selection

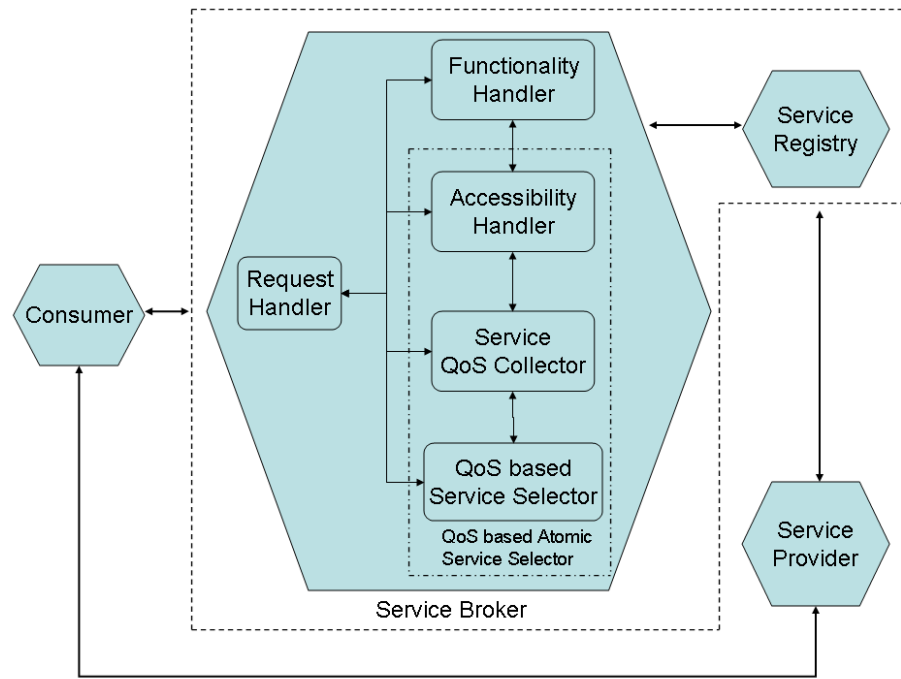


Figure 3.2: Sub-Architecture for Single Service Selection

Figure 3.2 represents the sub-architecture for single service selection. The QoS based single service selector is depicted into the sub modules. We assume that the consumer's functional requirements can be met by one service provider without composing other services. In this case, the function modules within the QoS based single service selector are illustrated as follows.

- **Accessibility Handler:** The accessibility handler identifies whether the consumer could access to the demand services by the consumer's attributes.
- **Service QoS Collector:** The published QoS information as well as the customized QoS information will be collected by the service QoS collector. The details of customizable QoS information will be presented in Chapter 5.
- **QoS based Service Selector:** Based on the consumer's QoS requirements and the service's published QoS information, the QoS based service selector is responsible for selecting the "best" service based on the similarity evaluation method in this thesis.

### 3.3 Sub-architecture for composite service selection

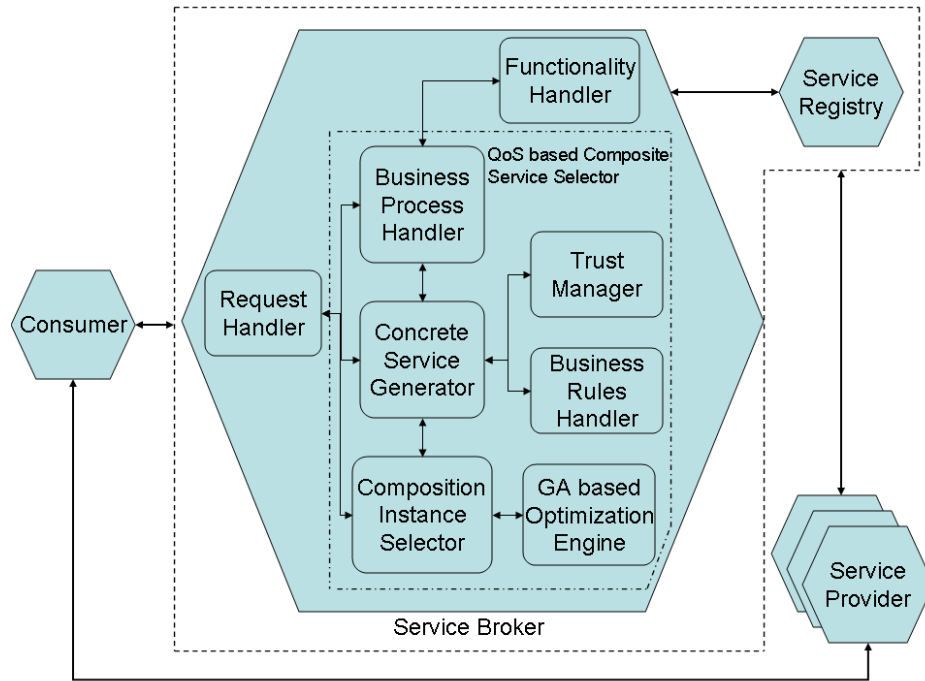


Figure 3.3: Sub-Architecture for Composite Service Selection

Figure 3.3 depicts the sub modules of the QoS based composite service selector as the sub-architecture for composite service selection. The concrete services are considered as functionally independent and they need to be composed together to

satisfy the consumer's requirements. In addition to service registering, requests handling, and functionality filtering, more function modules need to be considered in the service broker for the service composition. In the abstract service composition phase, the functionalities, the inputs, and the outputs of the services need to be analyzed for creating the business process. To generate the concrete service composition plan, the trust manager is responsible for managing the subjective opinions of the service consumers. The business rules are analyzed to detect the dependencies and the conflicts among the concrete services. In this thesis, the genetic algorithm is applied in the optimization engine to find the near-optimal composite service solution. In addition to Service Registry, Functionality Handler, and Request Handler, the sub modules involved in the QoS based composite service selector are illustrated as follows.

- **Business Process Handler:** When the consumer's functional requirements are received, the workflow as well as the invocation types will be determined by the business process handler. The abstract service composition is established by clustering the functionalities of the services.
- **Concrete Service Generator:** After the abstract service composition is created, the available concrete service candidates are composed together by the concrete service generator.
- **Trust Manager:** Trust manager is responsible for managing the subjective opinions of the service consumers. The local trust degree is computed by the direct and indirect experiences. The global trust degree is calculated by the multiple local trust degrees, depending on different invocation types. The details will be shown in Chapter 6.
- **Business Rules Handler:** It is responsible for collecting and analyzing the business to business relationships which will cause dependencies, conflicts, and QoS changing within the service composition.
- **GA based Optimization Engine:** The optimization engine applies the genetic algorithm to find the near-optimal concrete service composition plan. The further details of the settings on GA engine will be described in Chapter 6.

- **Composition Instance Selector:** From the large number of possible concrete service composition plans, the optimized one will be selected by the composition instance selector for the service consumer.

## 3.4 Summary

The overall system architecture has been proposed in this chapter. We have considered that the comprehensive service broker needs to be developed as the core component to find the most suitable single service or composite service based on the consumer's functional and non-functional requirements. To better understand the overall architecture, we have illustrated two sub-architectures to identify different function modules for single service selection and composite service selection, respectively. Focusing on different selection problems, these function modules will be further described in the following chapters.



# Chapter 4

---

## Similarity Based Service Selection

Chapters 4 and 5 focus on research issues on QoS-aware single service provider selection. In Chapter 4, the similarity based service selection model is introduced to overcome the limitations of the existing QoS based service matchmaking and selecting approaches. This chapter is organized into sections as follow. In Section 4.1, we introduce the problems and the limitations of the existing selecting approaches by a motivating case. As a result, three major properties that need to be considered for the service selection are presented. Section 4.2 presents the service selection model, including the QoS request model, the Non-functional property (Nfp) publication model, and the QoS based service selector. The similarity based service and requirement matchmaking phase and selecting phase are described in Section 4.3 and Section 4.4, respectively. A case study is analyzed in Section 4.5. Section 4.6 summarizes this chapter.

### 4.1 Introduction

As briefly mentioned in Chapter 1, efficiently finding web services is a challenging issue in SOC environment. Current web service technologies enable classifying the web services into functional categories. Therefore, selecting web service with high quality while satisfying user's requirements within a large pool of functionally equivalent service providers is the key factor for service selection.

Since service selection is commonly driven by *Non-functional Property*, it is critical to evaluate *QoS* of service accurately and objectively. Considering the typical web service discovery scenario, the user is able to provide a complete definition of the desired service and pose a query to the repository of advertised services. After that, the broker always employs a matchmaking algorithm to identify advertisements relevant to the user's request. A lot of work focused on collecting quality ratings

while others presented a similarity assessment between a request and an advertised service.

Single similarity value between published services and request is efficient and easy to use in service comparison and selection. Normally, the similarity score is normalized to a value in  $[0,1]$  based on the metrics of QoS. This value reflects the absolute distance between the published QoS attribute and request. Unfortunately, this kind of distance cannot depict the similarity very well under certain circumstances. For example, if there are two service providers  $A$  and  $B$  with their similarity value on a particular QoS dimension  $S_A = 0.8$  and  $S_B = 0.8$  (each of them is in the range of  $[0,1]$ ), does it mean that both  $A$  and  $B$  have the same quality on this dimension? It is not true if  $A$  performs its most ability to achieve the value, while  $B$  has more tolerance. In this case, service  $B$  is better than  $A$  in terms of predicting the performance during the implementation. In order to observe the tolerance of a service, the robust similarity assessment is required. Therefore, our approach is based on the relative distance between the published value and the request.

Generally speaking, in web service selection, the similarity measures the degree of match as the similarity score between a request and a published service. Typically, this process involves two steps: (i) filtering the matched service providers as the candidates from a large number of potential service providers in the same functional category, (ii) selecting a criterion for assessing the similarity of service parameters and aggregating individual parameter scores to obtain the overall degree. According to the research requirements that were mentioned in Section 2.4 (Requirement 1, 2 and 3), the desired major properties of a selection approach have to be considered as follows:

- *Property 1*: The consumer's expectation should be estimated accurately. A simple quality rating without consumer's expectation is insufficient. In our approach, we take the consumer's expectations into account and classify the consumer's willingness into five categories to help consumers representing the requirements accurately and specifically.
- *Property 2*: The estimating method should be flexible to calculate the similarity between a request and a published service. With the strict definition, sometimes, there is hardly a service provider satisfies the consumer's requirements perfectly. Furthermore, the single criterion may lead to information loss that significantly affects the results accuracy. Therefore, we advocate an

approach that employs multiple criteria for service selection.

- *Property 3*: The selection approach should be able to calculate the similarity between any data types, especially the categorical values.

To demonstrate the weaknesses of existing selection approaches and the importance of the three properties above, we use an example to explain the challenges in selecting the top matching services. Consider a user searching for a web service who provides movie download for specific locations. For simplicity, we assume the service providers only provide movie download services, and they are in the same functional category. There are four available services *A*, *B*, *C*, *D*, and their published QoS information and the user's requirements are presented in Table 4.1. Obviously, service *A* constitutes a best match with user's request. However, there is no clear winner among the other three services. For service *B*, all the parameters, except "cost", perfectly match the request and are better than services *C* and *D*. Normally, service *B* cannot join the candidate list even if it is a good potential service. Considering service *C* and service *D*, in some selection approaches, choosing a better service based on the weights of "Availability" and "Reputation" is realizable. However, for parameters "Category" and "Covered Country", there is no clear calculation method to detect the similarity distance between them. Although some parameters fully cover the request, how to find a better one is also an issue.

To address the three major properties and the weaknesses of existing selection approaches, our approach exhibits some new features, as summarized in the following:

1. We introduce a novel similarity computation approach which is based on the relative distance instead of the absolute distance for web service selection. In our approach, the tolerance of a service is considered as an aspect for calculating the similarity score.
2. Considering Property 1, we propose the QoS request model to represent the consumer's request accurately. The attributes of *Willingness* represent the tendencies of the consumer's preferences on different QoS dimensions. Specifically, we classify the consumer's willingness into five categories: *Top*, *Bot*, *Equivalent*, *Close* and *Distant*.
3. Considering Property 2, we introduce the attributes of *Negotiability* and *Data Type*

Service\ Parameter	Availability	Cost	Reputation	Category	Covered Country
Request	0.9	\$5	85	Japanese Carton	Australia, China
A	0.98	\$2	98	All	Australia, US, UK, China, Japan
B	0.98	\$6	95	All	Australia, US, UK, China, Japan
C	0.93	\$3	90	General	Australia, China, UK
D	0.95	\$4	90	Carton	Australia, China, UK, Japan

Table 4.1: Motivating case: the movie download services

to present the service request and the publication. The negotiable and non-negotiable values make the request and the publication flexible.

4. Considering Property 3, we propose a novel approach to calculate the similarity scores for the categorical values. The main idea is based on the set theory and semantic ontology. It is believed that the relationships between two categorical values could be depicted by a hierarchy from semantic web service ontology.

## 4.2 Service Selection Model

In this section, the general interaction framework of our single service selection approach is presented. The key entities and components of the approach are enumerated, and the relationships among them are illustrated.

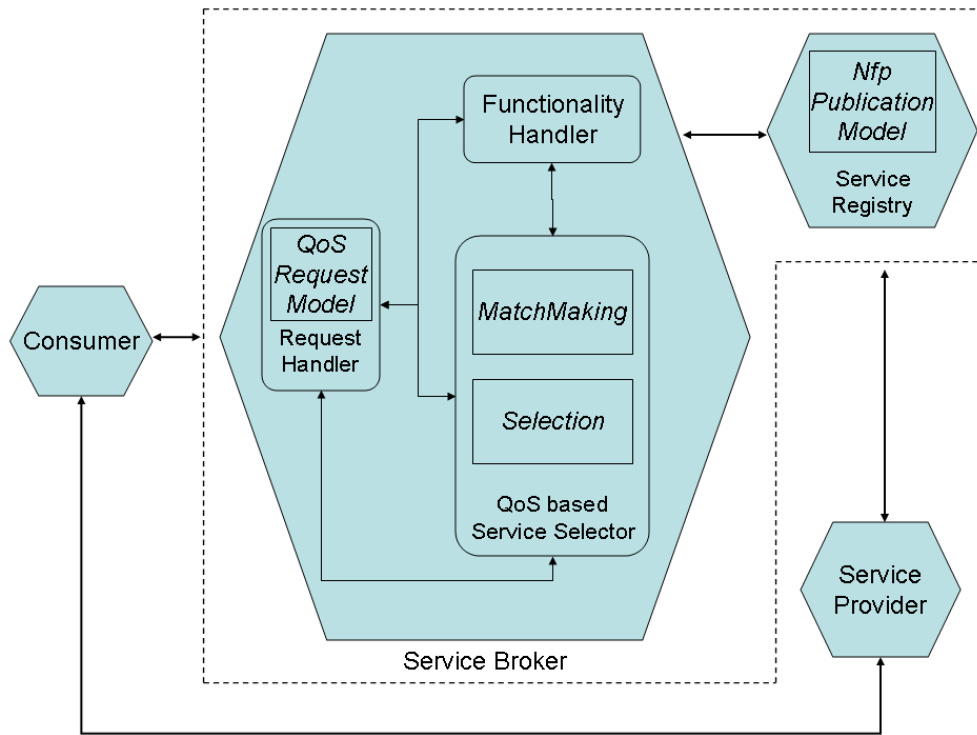


Figure 4.1: Similarity based Service Selection Model

As we introduced in Chapter 3, there are three main entities in the broker based architecture: (i) services provider, (ii) service consumer, and (iii) service broker. Figure 4.1 shows the major components and their interactions in our selection model. We hypothesize that the contracting process between service consumers and service

providers is mediated by the module of a service broker, which is named *QoS based Service Selector* as shown in Section 3.2. The *QoS based Service Selector* is responsible for analyzing the information of consumers' requests and providers' published description. In our service selection model, there are three basic components to support the interactions: the request handler for acquiring consumer's requests, the service registry for publishing the functional and the non-functional attributes, and the QoS based service selector for selecting the best service.

### 4.2.1 QoS Request Model

The service selection process starts when the service broker receives a request from the service consumer. We assume that service consumers could present their requirements including the functional and non-functional requirements. Basically, a request always refers to a specific category of functionally equivalent services, and it is characterized by the non-functional requirements, for instance, a budget. Generally, a service consumer could present a requirement  $R$  with many non-functional requirement dimensions  $I$  ( $I \in N$ ) for a specific category of services. While making a request, each requirement  $R_i$  has four attributes, which could be specified as the following quaternion  $R_i$ , with  $1 \leq i \leq I$ :

$$R_i = \langle Name, Negotiability, Willingness, Value \rangle;$$

The property *Name* is the name of the non-functional requirement dimension  $i$ , for example, availability, response time, or reputation, etc.

The property *Negotiability* indicates the requirement of negotiability on the dimension  $i$ , which is defined as either negotiable or non-negotiable.

- *Negotiable* requirements are those for which, in case the providers could not offer a satisfied value, the consumers may accept a different value from the service providers at runtime.
- *Non-negotiable* requirements are those for which the consumers have already strictly defined the value, and the values could not be modified at runtime.

The property *Willingness* represents the tendency of the service consumers' preferences for the particular requirement dimensions. In our research, we define five basic tendencies of the preferences: (i) *Top*, (ii) *Bot*, (iii) *Equivalent*, (iv) *Close*, and (v) *Distant*.

- *Definition 2.1: Bot ( $\perp$ )* tendency indicates that the value of the requirement ( $v_R$ ) is the bottom element of that dimension. The preferred  $v_{Nfp}$  (value of the published non-functional property) from the service provider must be more than or equal to  $v_R$ .
- *Definition 2.2: Top ( $\top$ )* tendency indicates that the value of the requirement ( $v_R$ ) is the top element of that dimension. The preferred value of non-functional property ( $v_{Nfp}$ ) from the service provider must be less than or equal to  $v_R$ .
- *Definition 2.3: Equivalent ( $\equiv$ )* tendency indicates that the preferred  $v_{Nfp}$  from the service provider must be equivalent to  $v_R$ .
- *Definition 2.4: Close ( $\wedge$ )* tendency indicates the consumers prefer a service provider whose  $v_{Nfp}$  is the most close to  $v_R$ .
- *Definition 2.5: Distant ( $\vee$ )* tendency indicates the consumers prefer a service provider whose  $v_{Nfp}$  is the most distant to  $v_R$ .

We argue that a consumer can only present one kind of willingness for one requirement dimension. The *Willingness* is specified as:

$$Willingness = \{\perp, \top, \equiv, \wedge, \vee\}.$$

The property *Value* represents the specific value of the requirement, i.e.,  $v_R$ , and it can be either numerical or categorical. The numerical value is of integer, real, or date/time data types, while the categorical value belongs to some domain-specific ontology which will be discussed in Section 4.3.

### 4.2.2 Nfp Publication Model

The service providers need to publish their functional and non-functional properties to the service registry.

For the service publication of the functional properties, UDDI is currently specified to support publishing the functional information in the SOA environment. However, it relies on static description of the service interfaces and functional attributes, and forces consumers to find and bind services at design time. A lot of researchers have focused on the semantic web services descriptions for either the

functional attributes or the non-functional aspects. The most relevant models are WSMO, OWL-S, and WSDL-S. The WSMO is a conceptual model for various aspects related to semantic web services. It provides an ontology based framework and specifications, which support the core element of the semantic web services. OWL-S is an ontology, within the OWL-based framework of the semantic web, for describing semantic web services. The OWL-S service ontology captures the service profile as well as the service-process model. WSDL-S is a proposed extension to the WSDL standard developed by IBM and the University of Georgia. WSDL-S extends to semantic elements which could improve the reusability and description of the web services. From the functional point of view, web services which are classified under the same category are equivalent [YL04, LNZ04b]. In our selection approach, we assume that the web services are classified in functional categories based on ontology.

For publication of non-functional properties, the service provides are generally dividing those properties into many dimensions. Within a particular functional category, if there are  $J$  ( $J \in N$ ) web services, each non-functional property  $Nfp_i$  of  $WS_j$  ( $1 \leq j \leq J$ ), has five attributes which could be specified in the following definition by the Relational Model:

$$\begin{aligned}
 Nfp &= (H, B); \\
 H(Nfp) &= \langle Name, Description, DataType, Metric, Value \rangle; \\
 B(Nfp) &= \{Nfp_1, Nfp_2, \dots, Nfp_n\}; \\
 Nfp_i &= t_i \in B(Nfp); \\
 &\text{With Functional Dependency:} \\
 A(DataType) &\rightarrow A(Metric) \wedge A(DataType) \rightarrow A(Value)
 \end{aligned}$$

The attribute *Name* presents the name of the non-functional property dimension  $i$ , such as execution time, price, or reputation, etc.

The attribute *Description* indicates the formal description of non-functional dimension which could be represented in web service description language.

The attribute *DataType* represents the data types and some special attributes that are defined by the domain experts within a functional category. In our research, we identify the data type into a quaternary of the binary string.

The attribute *DataType* can be either numerical or categorical.

- *Numerical* type indicates that the value of the  $Nfp_i$  belong to numbers, such as integer, real, or date/time.



- *Categorical* type indicates the value of the  $Nfp_i$  belong to some domain-specific ontology. For example, the value of identification is “professor”.

The attribute *DataType* can be either single or range.

- *Single* type indicates that the value of the  $Nfp_i$  belongs to a single number or a category. For example, the value of reputation is “98”, and the value of covered area is “New York”.
- *Range* type indicates that the value of the  $Nfp_i$  belongs to a range number or a range categories. For example, the execution time is “1-5(ms)”.

The attribute *DataType* can be either domain-dependent or domain-independent.

- *Domain-dependent* dimensions refer to the non-functional properties that are meaningful only within a specific functional category which is namely a domain. For the weather report service, for instance, the covered area is a domain-dependant dimension.
- *Domain-independent* dimensions refer to the non-functional properties that can be defined for any web services, regardless of their functional category. They may be related to the web service performance in terms of response time and availability, or other technical aspects, such as the encryption of data or the reputation of a service.

The attribute *DataType* can be negotiable or non-negotiable.

- *Negotiable* type indicates that the service provider offers different values, which are reported in an SLA, and can be chosen at runtime during the invocation.
- *Non-negotiable* type indicates that the service provider cannot modify the value which is provisioned.

The attribute *Metric* represents the data relationships within a functional category. Based on semantic ontology and set theory, our approach assumes that a hierarchy can exist for both numerical and categorical value types. The hierarchy is provided by the service provider or the domain experts. From the consequence point of view, we classify the hierarchy into two classes: consecutive hierarchy (*Hierarchy<sub>C</sub>*) which is based on the set theory, and inconsecutive hierarchy (*Hierarchy<sub>INC</sub>*) which is based on the semantic ontology. *Hierarchy<sub>C</sub>* is the default

predicate, while *Hierarchy<sub>INC</sub>* is provided by web service providers or the domain experts. The details of the hierarchy is presented in depth in Section 4.3.

The attribute *Value* represents the specific value of the *Nfp<sub>i</sub>*, and it can be either numerical or categorical as well, depending on the *DataType* of the *Nfp<sub>i</sub>*.

### 4.2.3 QoS based Service Selector

The *QoS based Service Selector* is the core component for establishing relationship between consumers and service providers. It is responsible for filtering the unsatisfied services, and generating the similarity scores to select the services. The single service selection process considered in our research is characterized by two phases performed by the *QoS based Service Selector*.

1. *Phase 1: Matchmaking.* When a request is received, the first phase filters the service providers that are published in the requested functional category.
2. *Phase 2: Selection.* Among the service providers that were filtered in the previous phase, the “best” provider is selected. Specifically, providers are ranked by the similarity score; and they are listed for the selection.

In the following two sections, we will introduce these two phases in details.

## 4.3 The Matchmaking Phase

The first phase, namely the matchmaking phase, operates after the consumers send a request to the system. This component is responsible for selecting the candidate service providers in the same functional service category which is requested by the consumers. In order to become a candidate for the second phase, namely the selection process, the service provider must satisfy the following two constraints at the same time.

1. *Dimension Matching:* The service must provide the values of non-functional properties to all the requested values ( $V_R$ ) expressed by the service requestor.
2. *Value Matching:* The service published values ( $V_P$ ) must cover all the non-negotiable requirements values expressed by the service requestor in  $V_R$ .

The first constraint is focusing on assessing whether the providers have abilities to provide the values to match all the non-functional dimensions presented by the requestors. The second constraint aims to assess whether the values could cover the requests. For the negotiable values of the requests, we only concern the first constraint at this stage because the requestor allows the providers to partially cover the requirements.

#### 4.3.1 Matchmaking on Dimensions matching

The attribute *Name* of  $R_i$  and  $Nfp_i$  represents the definitions of the dimensions which can be used to evaluate the first constraint for the dimensions matching. In order to satisfy the first constraint, the set of the service's non-functional dimensions must be the superset of the set of requestor's. We define the function  $M(V_R, V_P)$  to assess the proportion of containment between the dimensions of  $V_R$  and  $V_P$  as follow:

$$M(V_P, V_R) = \frac{|\{x|x \in V_P \wedge x \in V_R\}|}{|\{x|x \in V_R\}|}$$

Obviously, the value of  $M(V_R, V_P)$  belongs to the interval  $[0,1]$ . If  $M(V_R, V_P) = 1$ , the set of non-functional dimensions of the provider is the superset of, or at least equal to, the set of requested dimensions. If  $M(V_R, V_P) < 1$ , the set of provider's dimensions is either the subset of or does not cover the set of requested dimensions. Therefore, the boolean variable  $match_d(V_P, V_R)$  is computed as:

$$match_d(V_P, V_R) = \begin{cases} 1, & M(V_R, V_P) = 1 \\ 0, & otherwise \end{cases} \quad (4.1)$$

Hence,  $match_d(V_P, V_R)$  is set to one when the set of service dimensions is the superset of the set of requested dimensions. Otherwise, the service provider is not satisfied with the first condition.

#### 4.3.2 Matchmaking on values matching

After filtering the service providers for the first constraint, the boolean variable  $match_v$  assesses the second constraint for matchmaking the values on non-negotiable dimensions of the requirements. In order to evaluate  $match_v$ , we identify the formulation into two groups depending on the attribute *Negotiability*: numerical and

categorical. Whether the value of the service provider is satisfied with the requirement or not depends on the consumer's *Willingness*. For example, if the  $R_i$  is  $< reputation, non - negotiable, \perp, 80 >$ , the value of  $Nfp_i$  must be equal to or more than 80 to satisfy this request. We introduce the computation of  $match_v$  for numerical values and categorical values, separately.

For the numerical values,  $match_v$  can be computed based on the five different *Willingness* as follows:

$$match_v(v_{Nfp_i}, v_{R_i}) = \begin{cases} 1, & \text{if } v_{Nfp_i} \geq v_{R_i} \\ 0, & \text{otherwise} \end{cases} \quad (4.2)$$

with *Willingness* = " $\perp$ "

$$match_v(v_{Nfp_i}, v_{R_i}) = \begin{cases} 1, & \text{if } v_{Nfp_i} \leq v_{R_i} \\ 0, & \text{otherwise} \end{cases} \quad (4.3)$$

with *Willingness* = " $\top$ "

$$match_v(v_{Nfp_i}, v_{R_i}) = \begin{cases} 1, & \text{if } v_{Nfp_i} \equiv v_{R_i} \\ 0, & \text{otherwise} \end{cases} \quad (4.4)$$

with *Willingness* = " $\equiv$ "

Because the *Willingness* of 'Close' and 'Distant' represent the distance relationship between the service value and the request value,  $v_{Nfp_i}$  could be more than, less than, or equal to  $v_{R_i}$ . Thus, any numerical value satisfies the second condition under these two willingness types.

$$match_v(v_{Nfp_i}, v_{R_i}) = 1 \quad (4.5)$$

with *Willingness* = " $\wedge$ " or " $\vee$ "

For the categorical value, our approach assumes that hierarchy relationships, which are provided by the service providers or the domain experts, exist for the categorical values. We classify the hierarchy types into two groups: consecutive hierarchy and inconsecutive hierarchy. The consecutive hierarchy relies on the number of members in the value set, which the distance could be measured based on the dominate relations, while the inconsecutive hierarchy relies on the provided semantic ontology.

On one hand, for the consecutive hierarchy, in order to satisfy the second condition,  $v_{Nfp_i}$  should be a subset/superset or equal to  $v_{R_i}$  depends on the different

*Willingness* types.  $match_v$  can be computed as follow:

$$match_v(v_{Nfp_i}, v_{R_i}) = \begin{cases} 1, & \text{if } v_{Nfp_i} \supseteq v_{R_i} \\ 0, & \text{otherwise} \end{cases} \quad (4.6)$$

with *Willingness* = “ $\perp$ ”

$$match_v(v_{Nfp_i}, v_{R_i}) = \begin{cases} 1, & \text{if } v_{Nfp_i} \subseteq v_{R_i} \\ 0, & \text{otherwise} \end{cases} \quad (4.7)$$

with *Willingness* = “ $\top$ ”

$$match_v(v_{Nfp_i}, v_{R_i}) = \begin{cases} 1, & \text{if } v_{Nfp_i} \equiv v_{R_i} \\ 0, & \text{otherwise} \end{cases} \quad (4.8)$$

with *Willingness* = “ $\equiv$ ”

$$match_v(v_{Nfp_i}, v_{R_i}) = \begin{cases} 1, & \text{if } v_{Nfp_i} \cap v_{R_i} \neq \emptyset \\ 0, & \text{otherwise} \end{cases} \quad (4.9)$$

with *Willingness* = “ $\wedge$ ” or “ $\vee$ ”

On the other hand, for the inconsecutive hierarchy, it is not easy to express the relationship between two values, because the hierarchy is based on the semantics. For instance, there is no evidence to make the decision of matchmaking between the value “professor” and “lecturer” without the semantical hierarchy. Therefore, the hierarchy must be provided by the providers or the domain experts to evaluate the inconsecutive categorical values. We give an example of the category hierarchy in Figure 4.2, where each node represents a categorical value. For example, the parental, general, and restricted movies are classified in the same level in the hierarchy; and the action, carton, and drama are identified as the lower level of the general movies.

In order to satisfy the second constraint, there must be a nominate relationship between two values. For the simplicity in the hierarchy, these two value nodes must in one root-leaf path; otherwise, the service does not satisfy the second condition. In order to express the relationship, we propose to assign each node a hierarchy code (Hcode), indicating the position of each node. In particular, the root node is assigned an Hcode ‘1’, and its children nodes are assigned in the order from left to right by appending their position to the parent’s Hcode with a separator ‘.’, where we will have Hcodes like ‘1.1’ and ‘1.2’. Then the process continues till the leaf level. The number of elements separated by ‘.’ equals to the level at which a node is

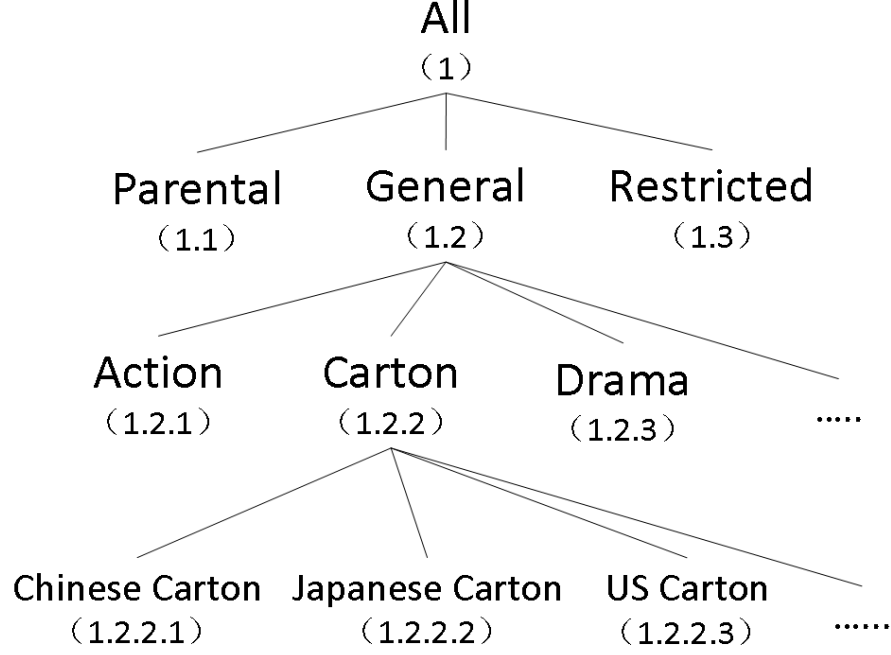


Figure 4.2: The Category Hierarchy

located. From such Hcodes we can easily find the relationship between two nodes, in other words, whether they are in the same root-leaf path or not. For example, the node *General* (1.2) and the node *Chinese Carton* (1.2.2.1) are in the same root-leaf path. We can define that if Hcode( $b$ ) is start from Hcode( $a$ ), we name node  $a$  *Generate* ( $\gg$ ) node  $b$ . Hence, the  $match_v$  between two inconsecutive categorical values can be computed as follow:

$$match_v(v_{Nfp_i}, v_{R_i}) = \begin{cases} 1, & \text{if } v_{Nfp_i} \gg v_{R_i} \\ 0, & \text{otherwise} \end{cases} \quad (4.10)$$

with *Willingness* = “ $\perp$ ”

$$match_v(v_{Nfp_i}, v_{R_i}) = \begin{cases} 1, & \text{if } v_{R_i} \gg v_{Nfp_i} \\ 0, & \text{otherwise} \end{cases} \quad (4.11)$$

with *Willingness* = “ $\top$ ”

$$match_v(v_{Nfp_i}, v_{R_i}) = \begin{cases} 1, & \text{if } v_{Nfp_i} \equiv v_{R_i} \\ 0, & \text{otherwise} \end{cases} \quad (4.12)$$

with *Willingness* = “ $\equiv$ ”

$$match_v(v_{Nfp_i}, v_{R_i}) = \begin{cases} 1, & \text{if } v_{Nfp_i} \gg (\ll) v_{R_i} \\ 0, & \text{otherwise} \end{cases} \quad (4.13)$$

with *Willingness* = “ $\wedge$ ” or “ $\vee$ ”

Finally, we could evaluate whether the service could satisfy those two conditions in the matchmaking phase. We define the function  $MATCH(V_P, V_R)$  as the formulation below:

$$MATCH(V_P, V_R) = match_d(V_P, V_R) \times \prod_{i=1}^n match_v(v_{Nfp_i}, v_{R_i}) \quad (4.14)$$

If  $MATCH(V_P, V_R)$  is 1, the service will be set in the candidate category; otherwise, the service does not match the consumer's requirements. After all the services in the functional category are filtered by matchmaking phase, these matched service providers will be processed in the next phase: selection.

## 4.4 Service Selecting Phase

After the QoS based Service Selector filtered the un-satisfied service providers in the matchmaking phase, suppose  $J$  services matched requestor's requirements. If  $J=0$ , there is no web service left and none of the service providers is satisfied with the requirements. If  $J=1$ , there is one web service matched. Thus, the selection process is unnecessary. If  $J \geq 2$ , the selection process is needed. In the selection process, as we mentioned before, the matched providers are ranked by a similarity score. The consumer selects the service provider offering the service that is the most similar to his demands.

### 4.4.1 Selection Method

Our selection method is developed based on the similarity score. The service provider with the highest similarity score will be selected. Specifically, we compute the similarity function  $S(V_P, V_R)$  between the requirement value  $V_R$  and the published value of the service  $V_P$  as follows:

$$S(V_P, V_R) = \sum_{i=0}^n w_i S(v_{Nfp_i}, v_{R_i}) \quad (4.15)$$

$w_i$  is the weight that can be used for emphasizing the importance of the target non-functional dimension  $i$ . The weight satisfies the constraint  $w_1 + w_2 + w_3 + \dots + w_n = 1$ .  $v_{Nfp_i}$  is the value of non-functional dimension  $i$  of the service; and  $v_{R_i}$  is the value of one non-functional dimension  $i$  of the requirement.

Based on the different DataTypes in published non-functional properties, we introduce two different measurements for the categorical values and numerical values, respectively. For the categorical values, we consider not only the exact match of two values, but also their semantics similarity. It is believed that a hierarchy relationship can be developed based on either the semantic ontology or the set theory. Before we introduce the similarity measurement, we define a order relation: *Exactly Dominate* ( $\succ$ ).

- *Definition 4.1*: Let Set  $B$  be a proper subset of Set  $A$ , if  $A \succ B$ , then, there is no Set  $C$  which is both a proper subset of Set  $A$  and a proper superset of Set  $B$ .

$$A \succ B \Leftrightarrow A \supset B \wedge \nexists C \{C | A \supset C \supset B\}$$

- *Syllogism 4.1*: For Set  $A$  and Set  $B$ , if  $A \succ B$ , then, the cardinality of Set  $A$  is one greater than Set  $B$ .

$$A \succ B \rightarrow \#A = \#B + 1$$

From the transitivity of the order relation  $\succ$ , we formulate the order relation exactly dominate by  $n$  ( $\succ^n$ ) as following:

- *Syllogism 4.2*: For Set  $A$  and Set  $B$ , if  $A \succ^n B$ , then, the cardinality of Set  $A$  is  $n$  greater than Set  $B$ .

$$A \succ^n B \rightarrow \#A = \#B + n$$

The similarity between the value of a request and the value of a published service, is considered as the distance based on consumer's willingness. The numerical values are efficiently measurable, while the categorical values are much complicated to compute. However, any categorical value could be considered as a set; rely on the exactly dominate relation we introduced above, we could also compute the distance between two categorical values. In the following two subsections, we will evaluate the measurement of similarity score for categorical values and numerical values.



### 4.4.2 Similarity Measurement for Categorical Values

Our approach assumes that hierarchy relationships, which are provided by the service providers or the domain experts, are available for the categorical values. We classify the hierarchy types into two groups: consecutive hierarchy and inconsecutive hierarchy. The consecutive hierarchy relies on the number of members in the value set; while the inconsecutive hierarchy relies on the provided semantic ontology.

On one hand, for the consecutive hierarchy, we introduce a function  $LEV[A \succ^n B]$  to present the distance between two categorical values. The formulation of  $LEV[A \succ^n B]$  is given by the following equation:

$$LEV[A] = \#A \quad (4.16)$$

$$LEV[A \succ^n B] = \#A - \#B \quad (4.17)$$

Because our similarity score is considered as the distance under consumer's willingness, the measurements is different for the five different tendencies of the consumers' preferences, i.e., *Top*, *Bot*, *Equivalent*, *Close*, and *Distant* in Section 4.2. Thus, the similarity score between  $v_{Nfp_i}$  and  $v_{R_i}$  is computed as follows:

$$S_{CAT}^C(v_{Nfp_i}, v_{R_i}) = \left\{ \begin{array}{l} \frac{LEV[v_{Nfp_i} \succ^n v_{R_i}]}{LEV[v_{Nfp_i}]} \\ \quad ; with Willingness = "\bot" \\ \\ \frac{LEV[v_{Nfp_i} \succ^n v_{R_i}]}{LEV[v_{R_i}]} \\ \quad ; with Willingness = "\top" \\ \\ 1 - \frac{|LEV[v_{Nfp_i} \succ^n v_{R_i}]|}{\max\{LEV[v_{R_i}], LEV[v_{Nfp_i}]\}} \\ \quad ; with Willingness = "\wedge" \\ \\ \frac{|LEV[v_{Nfp_i} \succ^n v_{R_i}]|}{\max\{LEV[v_{R_i}], LEV[v_{Nfp_i}]\}} \\ \quad ; with Willingness = "\vee" \end{array} \right. \quad (4.18)$$

On the other hand, for the inconsecutive hierarchy, it is very difficult to express the distance between the two values by the numbers of the member of the set, because the hierarchy is based on the semantics. For example, the similarity between

the value “professor” and “student” is not evident without the semantic relationship. Therefore, the measurement of the similarity score between two inconsecutive categorical values is based on the hierarchy.

After the matchmaking phase, there must be a nominate relationship between the value of a request and the value of a published service. In the hierarchy, each node is nominated by its parent node. The underlying idea is that the distance between two value nodes is represented by the number of the nodes between them. We formulate the measurement of the similarity score as follows:

$$S_{CAT}^{INC}(v_{Nfp_i}, v_{R_i}) = \begin{cases} \frac{Path[v_{Nfp_i}, v_{R_i}]}{Path[v_{root}, v_{R_i}]} \\ ; with Willingness = “\perp” \\ \frac{Path[v_{Nfp_i}, v_{R_i}]}{Path[v_{leaf}, v_{R_i}]} \\ ; with Willingness = “\top” \\ 1 - \frac{Path[v_{Nfp_i}, v_{R_i}]}{H[v_{Nfp_i}, v_{R_i}]} \\ ; with Willingness = “\wedge” \\ \frac{Path[v_{Nfp_i}, v_{R_i}]}{H[v_{Nfp_i}, v_{R_i}]} \\ ; with Willingness = “\vee” \end{cases} \quad (4.19)$$

The function  $Path[v_{Nfp_i}, v_{R_i}]$  denotes the distance between  $v_{Nfp_i}$  and  $v_{R_i}$ . The function  $H[v_{Nfp_i}, v_{R_i}]$  indicates the number of the nodes in the longest path from the root node to the leaf node that passes both  $v_{Nfp_i}$  and  $v_{R_i}$ . In order to simplify the computation of the distance, we can use the Hcode which is presented in Section 4.3. From such Hcodes we can easily find the path and compute the number of nodes in that particular path.

#### 4.4.3 Similarity Measurement for Numerical Values

Unlike the categorical values, the numerical values do not have any hierarchical relationship. However, based on the datatype of  $Nfp_i$ , the service providers have to present additional information in the attribute “Metric”. Specifically, for the *single* datatype of numerical values, the maximum and minimum values ( $v_{max}$  and  $v_{min}$ ) of

that non-functional property dimension have to be claimed. If the reputation value  $v_{Nfp_i}$  for instance, is 85, the additional information about  $v_{max}$  (e.g. 100) and  $v_{min}$  (e.g. 0), has to be presented in the attribute “*Metric*”. For any *range* datatype values, although the maximum and minimum values have already presented in the attribute “*Value*”. We formulate the measurement as the expressions below:

$$S_{num}(v_{Nfp_i}, v_{R_i}) = \begin{cases} \frac{v_{Nfp_i} - v_{R_i}}{v_{max} - v_{R_i}} \\ ; with Willingness = “\perp” \\ \frac{v_{R_i} - v_{Nfp_i}}{v_{R_i} - v_{min}} \\ ; with Willingness = “\top” \\ 1 - \frac{|v_{R_i} - v_{Nfp_i}|}{max\{v_{Nfp_i}, v_{R_i}\}} \\ ; with Willingness = “\wedge” \\ \frac{|v_{R_i} - v_{Nfp_i}|}{max\{v_{Nfp_i}, v_{R_i}\}} \\ ; with Willingness = “\vee” \end{cases} \quad (4.20)$$

The similarity score for both the categorical values and the numerical values can be computed as a decimal in  $[0,1]$  in the case that the requirement of negotiability is *Non-negotiable*. For the *Negotiable* values, the similarity score might be computed as a negative number. For example, when the *willingness* is ‘ $\perp$ ’, the value of published service can not satisfy the requirement, which means  $v_{Nfp_i} - v_{R_i} < 0$ . Therefore, in this case that no service perfectly satisfies the consumer’s requirements, the system will recommend the most similar ones. The consumers could preset the threshold and the weights of every non-functional dimensions to choose the “best” service.

## 4.5 Case Study

In this section we provide a detailed example to illustrate how our selecting approach works. Using example of selecting movie downloading services in Section 4.1, we show how our similarity based selecting approach assigns a similarity score to present the distance between the request and the published service. Based on the similarity score computed by our algorithm, a ranking list of all the services could be presented, then the user may check the list and choose the preferred web service.

Once the consumer makes a request, the requirements are formed as a matrix by the QoS request model. The requirements of negotiability and willingnesses of the consumer are shown as follows:

$$\left[ \begin{array}{llll} \textit{Availability} & \textit{Non - negotiable} & \perp & 0.9 \\ \textit{Cost} & \textit{Negotiable} & \top & \$5 \\ \textit{Reputation} & \textit{Non - negotiable} & \perp & 85 \\ \textit{Category} & \textit{Negotiable} & \perp & \textit{JapaneseCarton} \\ \textit{CoveredCountry} & \textit{Non - negotiable} & \perp & \textit{Australia, China} \end{array} \right]$$

Each row in this model indicates a requirement of a non-functional dimension, and four columns present the attributes of the requirement: *Name*, *Negotiability*, *Willingness*, and *Value*.

We assume that there are four service providers in the functional category of movie downloading: A, B, C, and D. Each service provider publishes the non-functional properties as a matrix, following our non-functional property model, Table 4.2 shows the non-functional properties of service A.

Name	Description	DataType	Metric	Value
Availability	Expected percentage of time the service is up and accessible	0011	Percentage	98
Cost	The price of the downloading for each movie	0010	Price Unit (\$)	2
Reputation	The public opinion about the honesty, capability, and reliability of the service provide	0011	$v_{min} = 0$ , $v_{max} = 100$	98
Category	The classification of movies which the service provides	1000	$Hierarchy_{INC}$	All
Covered Country	The country(s) on which the service provides downloading	1110	$Hierarchy_C$	Australia, US, UK, China, Japan

Table 4.2: The non-functional property model for service A

The binary string indicates the four pairs of attributes in the *DataType* column. The *Availability* for instance, 0011 indicates the value of that non-functional dimension is numerical, single, domain-independent, and non-negotiable. We also refer to two hierarchies of different dimensions for service A, namely the category hierarchy (Figure 4.2) and covered the country hierarchy. The category hierarchy is domain-dependent and is provided by the domain expert. The covered country hierarchy is bases on the set theory.

Our similarity based selecting approach is processing as followed:

- Matchmaking Phase: firstly, we compute the  $match_{dimension}$  for these four services. In this case, the publication of A, B, C, and D cover all the dimensions of the request. Thus,  $match_{dimension}$  between each service and the request equal to 1. Secondly, the values matching only considers the non-negotiable dimensions of the request. As shown in Table 4.3, all these four services match the user's request. They are the potential candidates for the selecting phase.

$match_{dimension}(V_A, V_R)$	1	$match_{value}(V_A, V_R)$	1	$MATCH(V_A, V_R)$	1
$match_{dimension}(V_B, V_R)$	1	$match_{value}(V_B, V_R)$	1	$MATCH(V_B, V_R)$	1
$match_{dimension}(V_C, V_R)$	1	$match_{value}(V_C, V_R)$	1	$MATCH(V_C, V_R)$	1
$match_{dimension}(V_D, V_R)$	1	$match_{value}(V_D, V_R)$	1	$MATCH(V_D, V_R)$	1

Table 4.3: The matchmaking values in the case

- Selecting Phase: we compute the similarity score for each non-functional dimension of a candidate. Firstly, the similarity score of each dimension is computed by our algorithm. Secondly, the final similarity score is computed, combined with the weights of the user's preference. The results are shown in Table 4.4. The user could change the weights of each dimension. From the final result, obviously, service A is the best. Among the other three, service B is a good candidate even the dimension of cost does not fulfill the request. Service D is better than C.

From the case, we observe that the service B does not fulfill the consumer's request on the dimension of cost. In other strict selection approaches, this service provider would be simply blocked even it could provide good quality on many other dimensions. The result of our approach shows the advantage that service B is still staying as a selection candidate because the cost dimension is negotiable. As a result,

Dimension	Weight	Service A	Service B	Service C	Service D
Availability	0.25	0.8	0.8	0.3	0.5
Cost	0.1	0.6	-0.2	0.4	0.2
Reputation	0.3	0.87	0.67	0.33	0.33
Category	0.15	1	1	0.67	0.33
Covered Country	0.2	0.6	0.6	0.33	0.5
Final Score		0.79	0.65	0.38	0.40

Table 4.4: The result of similarity measurement in the case

service B might be a positive choice and the value on cost could be further negotiated. Furthermore, by our selection approach, there is a comparability between all the services with any kind of data, such as the categorical values on category and covered country dimensions.

## 4.6 Summary

In this chapter, we have defined a novel similarity measurement between the consumer's request and the service publication for the single service selection. We have identified the willingness of the consumers representing their tendencies in the non-functional dimensions. Unlike the other similarity measurement, we have obtained the relative distance to compute the similarity score for generating the ranking list. Furthermore, our matchmaking approach is flexible and enables consumers to pre-set the negotiable QoS attributes. For computing the similarity score between two categorical values, a novel method based on the semantics and set theory has been introduced.

# Chapter 5

---

## Attribute Based Service Customization and Selection

As mentioned in Chapter 4, the service selection decisions are commonly driven by non-functional property because the consumers are concerned with the QoS beyond the functionalities. In this chapter, we further investigate how to publish and collect the particular QoS value for the consumers, because many service providers have a need to offer customized QoS to different users or user groups rather than a single QoS publication. This chapter is organized into sections as follow. In Section 5.1, we introduce the problems and the needs of service customization and selection based on the consumers' attributes. Section 5.2 presents the system model and data flow of the attribute based service customization and selection approach. The policy language is defined in Section 5.3. Section 5.4 presents the policy evaluation method which is based on the attribute based access control structure. The similarity based service selection approach is discussed in Section 5.5. The system prototype is demonstrated in Section 5.6. Finally, Section 5.7 summarizes this chapter.

### 5.1 Introduction

As we mentioned in Chapter 1, selecting services with high quality within a large pool of functionally equivalent service providers is the key factor of service selection. Furthermore, in the highly competitive global service market, service customization has become one of the key strategies for a service provider to differentiate itself from the competitors. Service customization, which was originated in marketing, requires providing the customer with customized products and services without exceeding the price of comparable standard products. Since the number of services published on the Internet is increasing rapidly, the users' selection decisions are commonly driven by non-functional property because they are concerned with the QoS beyond the

functionalities. To satisfy the users' requirements, the ability to customize QoS is a desirable feature for the service providers to improve their competitiveness.

As we discussed in Chapter 2, there are a lot of work focused on analyzing the difference or the similarity between the consumer's requirements and the provider's published QoS values, which we called requirement based service selection. However, how to publish and collect the particular QoS value is an issue because many service providers are willing to offer customized QoS to the different people or groups. For example, a hotel service may provide different QoS, cheaper cost for instance, to the VIP members. This means different groups of consumers, distinguished by their attributes, would be offered the different QoS values. Furthermore, in many scenarios, service providers prefer to expose their QoS values only to the consumers who are entitled to access their services. Therefore, to meet the research requirement 3 in Section 2.4.1, the characteristics of the consumers, which are based on their attributes, need to be taken into account when selecting a service. In this case, static QoS publication in the service description is not suitable for customizing the service to different consumers dynamically. To satisfy the research requirement 4 in Section 2.4.1, it is considered that Attribute Based Access Control (ABAC) approaches, in which the access decisions are based on attributes of the consumers and resources rather than their identifications, are helpful to customize the QoS information. However, as the effect of the traditional ABAC approach is defined as an alternative choice "permit" or "deny" without any QoS information, a policy language needs to be extended in order to present the customized QoS information as the additional effects. In this chapter, we propose an attribute based service description model to help service providers publish their customizable QoS values in the ABAC policy. And the Customizing Rule (CRule) is introduced to collect the QoS values for different groups of the consumers.

## 5.2 System Model and Data Flow

To achieve the service customization based on consumer's attributes, service providers need to publish their customizable QoS values within the attribute based access control policies in order to expose the correct information to the specific consumers, and protect themselves from the unqualified consumers. In addition, the consumers need to acquire correct QoS information for service selection. According to the overall architecture that was described in Chapter 3, we further investigate the accessibility



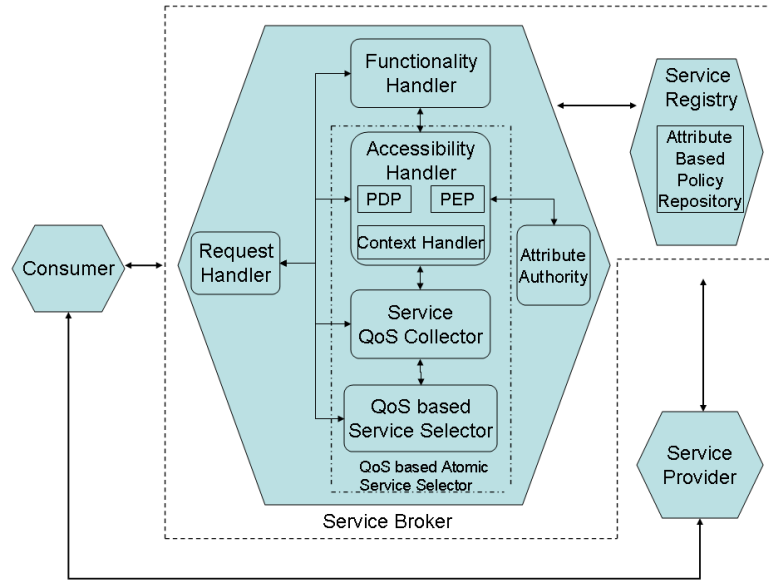


Figure 5.1: The Service Customization and Selection Model

handler and service QoS collector in details, and the attribute based service selection model is presented in Figure 5.1.

Basically, we assume that the service requests from consumers contain three classes of information: the functional interests, the consumer's attributes, and the QoS preferences. In our attribute based service selection approach, there are four basic phases to find the best service provider for satisfying the consumer's request automatically: functionality filtering, accessibility filtering, QoS collecting and service selecting. The six main components are introduced as follows:

- **Functionality Handler.** According to the consumer's functional request, the service providers are classified by the functions based on the service description and the semantic ontology.
- **Accessibility Handler.** It is responsible for making the access decision based on the obtained attributes from the consumer and the attribute based access control policies which are stored in the policy repository.
- **Service QoS Collector.** Depending on the access decisions, the service providers that can be accessed, are clustering as selecting candidates, and the customized QoS information is collected from the policy repository based on the consumer's attributes.

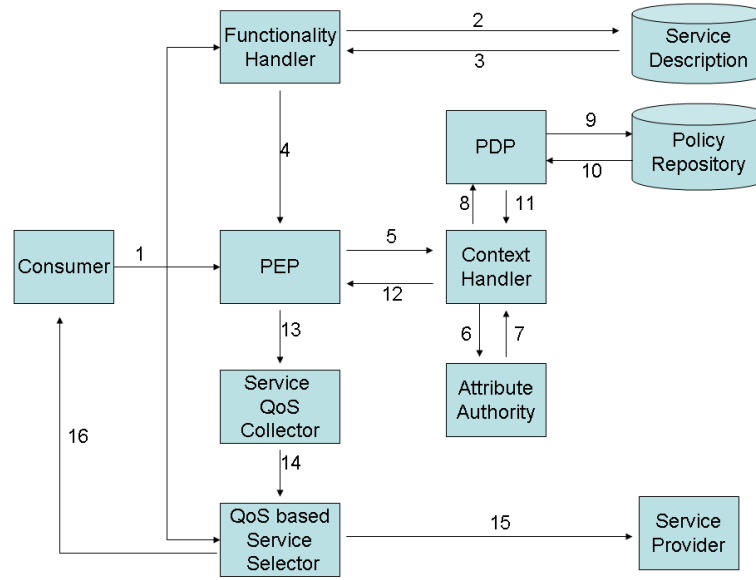


Figure 5.2: The Data Flow Diagram

- QoS based Service Selector. Based on the consumer's QoS requests, the service candidates are ranked, and the most suitable service provider is recommended to the consumer.
- Attribute Authority. It is responsible for authorizing and authenticating consumers' attributes.
- Service Registry. All service providers need to publish their services to the registry in order to be discovered by the consumers. There are two logical components to store the information about the services: service description and policy repository. The basic information of the services such as functionality is published in the service description form while the policy repository stores the attribute based access control policies of the services.

The data flow of the service selection approach is presented in Figure 5.2, and the details are described as follow:

1. The consumer sends the requests to the service broker. The functional request, the access request with user's attributes, and the QoS request are collected by the Functionality Handler, Policy Enforcement Point (PEP), and QoS based Service Selector, respectively.

2. Functionality Handler contacts the Service Registry to discover the available services which could provide the demanded functions.
3. Service Registry returns the potential services based on the functional description and semantic ontology.
4. After the functional filtering, the Functionality Handler forwards the potential services to PEP.
5. PEP sends the authorization request to the context handler.
6. The context handler then contacts the Attribute Authority to verify the consumer's attributes.
7. Attribute Authority authenticates the consumer's attributes and returns them to the context handler.
8. The context handler sends the request context (which contains consumer's attributes, service provider's attributes, and the environmental attributes information) to PDP for authorization decision.
9. PDP contacts the attribute based policy repository to retrieve the corresponding policy of the service provider.
10. The policy repository returns the access control policy.
11. Depending on the policy and the obtained attribute information, PDP makes the authorization decision and returns the decision to the context handler.
12. The context handler forwards the authorization decision to PEP.
13. PEP establishes connections to the accessible services which are the candidates for QoS sorting.
14. The customized QoS of all the service candidates are collected and then forwarded to the QoS based service selector.
15. Depending on the consumer's QoS request and the customized QoS publication, the QoS based service selector finds the most suitable service provider.
16. The QoS based service selector returns the sorting result to the consumer.

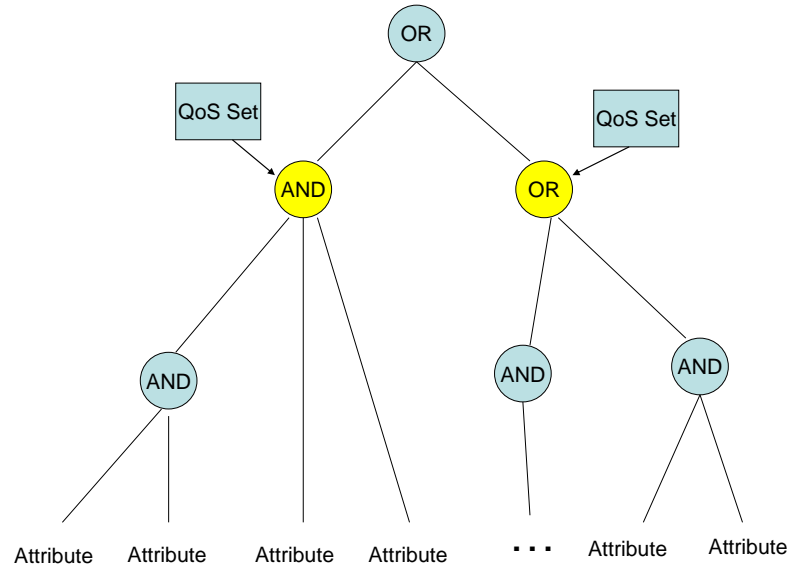


Figure 5.3: Attribute Based Access Control Structure

## 5.3 Attribute Based Policy Language

### 5.3.1 Attribute Based Access Control Structure

For the attribute based access control, consumers are evaluated by their particular attributes. The access control policy only categorizes the consumers into two groups: those who are permitted to access the service, and those who are denied the access to the service. However, for the former group, it is likely that they will be offered different QoS values, based on their attributes. Therefore, the access control policy needs to be extended to support the expression of QoS customization information. Normally, the attribute based access control policy can be formalized as a N-Tree structure[BSW07, GPSW06, CM12]. To group the consumers, the QoS values could be published on the access points as shown in Figure 5.3. Every AND node presents an AND relation which is called ANDSet. Every parent of the leaf nodes that present the attributes must be an AND node. Every OR node is presented as an attribute ORSet. Each customizing rule of the policy statement could be represented by an access point with the QoS publication.

### 5.3.2 Policy Language

In order to customize the QoS value to different consumers based on their attributes, the QoS effects need to be defined in the attribute based access control policies. Based on the consumer's attributes, the service provider's attributes, and the environmental attributes, the components Subject, Resource, and Environment are defined respectively. Normally, the accessibility is presented as either *permit* or *deny* which is the only effect in the traditional attribute based access control policy. In our proposed policy statement, the QoS effects need to be evaluated by the Customizing Rules and presented as an additional effect which is an important component of the policy statement. Based on the components we discussed above, each policy statement is represented as a tuple:

$$\langle \text{Resource, Subject, Environment, CRule Set, Effect} \rangle$$

Resource indicates the attributes of the demanded resource and is defined as follows:

$$\text{Resource} = \text{Attr}(r_1) \wedge \text{Attr}(r_2) \wedge \dots \wedge \text{Attr}(r_n)$$

Resource can be the identification of a service, the characteristics of the services, or the internally structured resource. For simplicity, the resource attribute indicates the identification of the service as the policy issuer in this research.

Subject indicates the target consumer's attributes. The PDP verifies whether the attributes presented by the consumer are satisfied by the Subject in the request context. The subject in our policy language is defined as an AttributeSet. There are two different kinds of AttributeSet: ANDSet and ORSet.

$$\text{Subject} = \text{AttrSet} = \text{ANDSet} | \text{ORSet}$$

ANDSet is a set of attributes or AttributeSets with *AND* relations, while ORSet is a set of AttributeSets with *OR* relations. In this case, the elements of ORSet cannot be an attribute because the single attribute could be presented by an ANDSet with a single attribute element. The ANDSet and ORSet could be defined as follow:

$$\begin{aligned} \text{ORSet} &= \text{AttrSet}(S_1) \vee \text{AttrSet}(S_2) \vee \dots \vee \text{AttrSet}(S_n) \\ \text{ANDSet} &= \text{Attr}(s_1) \wedge \text{Attr}(s_2) \wedge \dots \wedge \text{Attr}(s_n) \wedge \text{AttrSet}(S) \end{aligned}$$

Environment indicates the attributes that describe the technical, operational, or situational environment. For example, the valid date and time of the policy, the network of Internet or Intranet. The environment attributes are sometimes ignored, but they may be relevant in enforcing the access control policy. The environment attributes could be verified by the context handler and sent to PDP as an input for the decision making. The environment attributes are presented as:

$$Environment = Attr(e_1) \wedge Attr(e_2) \wedge \dots \wedge Attr(e_n)$$

CRule Set indicates a set of customizing rules which are defined to customize the QoS values for different groups of the consumers. They are presented by the service providers as the linkages between the subset of the subject and the QoS Set. Therefore, the head of each CRule must be an *AttrSet* which is described in the subject of the policy, and the body of the CRule is an identification of a specific set of QoS values in the QoS Set which is defined in the component of Effect. The formulation of each CRule could be described as:

$$AttrSet(S_i) \rightarrow Q_i$$

Effect indicates the accessibility and QoS Set which reflect the policy issuer's intended consequence of the evaluation of the policy. As we discussed before, the service providers are willing to expose the QoS values to the consumers who are permitted to access. Therefore, the consumers cannot acquire the QoS information if the accessibility requirements are not satisfied. QoS Set contains the information of the published QoS values for the potential consumers who could access the resource. The QoS Set could be presented as  $Q_i$ , and the QoS values on  $m$  different dimensions are presented as follow:

$$Q_i = \{q_i^1, q_i^2, \dots, q_i^m\}$$

The QoS value on each dimension ( $q_i^m$ ) can be published by the service publication model presented in Chapter 4 as:

$$q_i^m = \langle Name, Description, Type, Metric, Value \rangle$$

where *Name* presents the name of the QoS dimension  $m$ , such as execution time, price, or reputation, etc. *Description* indicates the formal description of the QoS dimension. *Type* represents the data types and some special properties that are

defined by the domain experts within a functional category. In Chapter 4, we have identified the data type into four pairs of fundamental properties: numerical or categorical, single or range, domain-dependent or domain-independent, negotiable or non-negotiable. *Metric* represents the data relationships within a functional category which are based on the semantic ontology or set theory. *Value* represents the specific value of the  $q_i^m$ , and it can be either numerical or categorical as well, depending on the type of the  $q_i^m$ .

Once the access control policies are published by the service providers, not only the access decisions could be made by PDP, but also the customized QoS values can be collected from the policy repository for the consumers. To better understand the essence of the policy, the policy statement could be formulated as follow:

$$\{\text{Resource, Subject, Environment, CRules}\} \rightarrow \{\text{Accessibility, QoS Set}\}$$

The components Resource, Subject, Environment, and CRules could be considered as the inputs of the Accessibility Filter, while the Accessibility and QoS Set are considered as the outputs.

## 5.4 Attribute Based Policy Evaluating

According to the attribute based access control structure discussed in Section 5.3, each CRule could be represented as the access point associated with the QoS publication (QoS Set). The subject is represented as the root node of that n-tree, the attribute sets in the subject are represented as the child nodes, and the attributes are the leaf nodes.

The service provider might publish more than one customizing rules according to different QoS Sets to distinguish the user groups in the access control structure. According to the tree structure of the policies, the service provider could add or delete a customizing rule much conveniently. Furthermore, for the policy evaluation process, the system does not have to evaluate all the customizing rules which are published by the same service provider because some of them might be overlapping. Generally, if the consumer holds the higher access level, the service provider could provide better QoS than the lower level.

When the consumer makes a request with his attributes to access the service, the system needs to evaluate whether the consumer are permitted to access the service, and, if he is permitted, which customizing rule is applied to the consumer. As

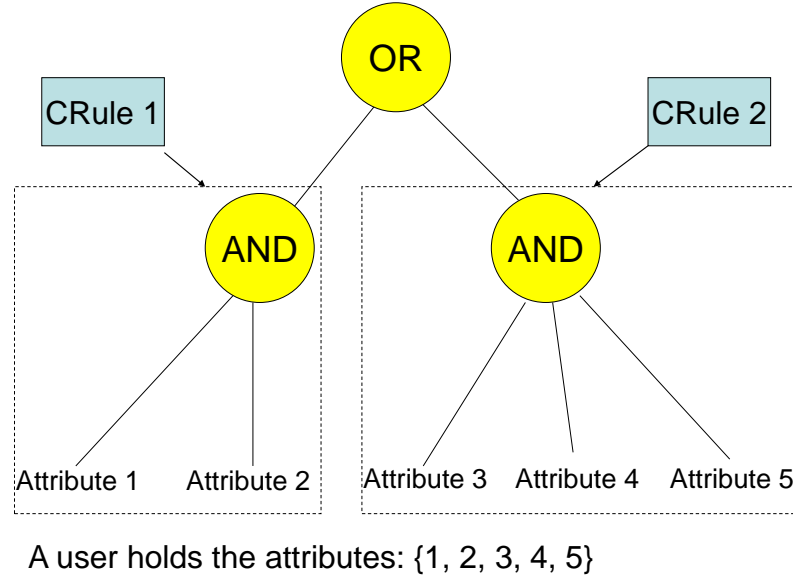


Figure 5.4: An example

we mentioned before, the consumer might satisfy more than one customizing rule because he provides more attributes. For example in Figure 5.4, the policy could be structured as a n-Tree with two different CRules. The policy states that the consumers who could provide  $Attr(u_1)$  and  $Attr(u_2)$ , or provide  $Attr(u_3)$ ,  $Attr(u_4)$ , and  $Attr(u_5)$  are permitted to access the resource. In addition, the QoS values which are presented in CRule 1 are applicable to the consumer group who provides  $Attr(u_1)$  and  $Attr(u_2)$ ; while the QoS values which are defined in CRule 2 are applicable to the consumers who hold  $Attr(u_3)$ ,  $Attr(u_4)$ , and  $Attr(u_5)$ . If a consumer holds all the attributes  $\{Attr(u_1), Attr(u_2), Attr(u_3), Attr(u_4), Attr(u_5)\}$ , obviously, he could access the service under the satisfied environment, and both the CRule 1 and CRule 2 are applicable. However, the service provider needs to return only one set of customized QoS values when the request is received. In order to solve the overlapping issues, the adoption strategy of the customizing rules needs to be determined first. In this paper, we illustrate three different adoption strategies: First-In-Applicable, Pre-Order-Applicable, and More-Attribute-Applicable.

- First-In-Applicable: The effect of the first satisfied customizing rule is the decision of the policy. The evaluating rules must be listed in the order from the root of the structure.
- Pre-Order-Applicable: The effect of the satisfied customizing rule with the



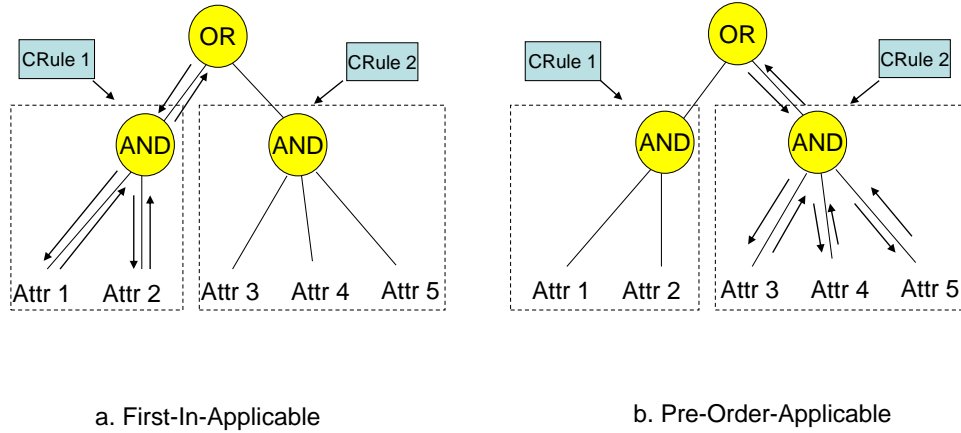


Figure 5.5: Different evaluating strategies

highest priority is the decision of the policy. The priority level must be pre-defined by the policy issuer.

- More-Attribute-Applicable: The effect of the satisfied customizing rule with more attributes included is the decision of the policy.

Based on the different evaluating strategies above, for every published attribute based access control policy, we might not have to traverse all the structured tree as shown in Figure 5.5. For the First-In-Applicable evaluating strategy (Figure 5.5.a), if the CRule 1 is satisfied (the algorithm defines the traversing order as from left child node to the right ones), the evaluating process will be terminated. For the Pre-Order-Applicable evaluating strategy (Figure 5.5.b), if we assume the priority of CRule 2 is higher than CRule 1; then, the traversing order will be from the highest priority node to the lowest one. Therefore, the evaluating process will be terminated if CRule 2 is satisfied. For the More-Attribute-Applicable evaluating strategy, all the structured tree will be traversed, and the CRule with more attributes of the subject will be selected.

According to the different evaluating strategies, only one CRule will be determined even if the consumer satisfies more than one customizing rule. This means only one set of QoS values could be collected if the consumer can access the service. The evaluating algorithm of the First-In-Applicable is presented in Algorithm 1. The algorithm inputs are a set of consumer's attributes and a n-tree structured policy from a service provider. The outputs are a policy satisfaction number  $s$  that indicates whether the consumer is satisfied with the policy, and a CRule number

**Algorithm 1:** Evaluating the policy using First-In-Applicable strategy

---

**Input:** A set  $U = \{u_1, u_2, \dots, u_n\}$  of consumer's attributes; n-Tree(policy)  
**Output:** The policy satisfaction number  $s$ , CRule number  $c$   
**Begin**  
 $node_i = root;$   
 $p=d;$ {Initialization, set parameter  $p$  as default}  
 $visit(node_i, p)$   
function  $visit()$  {  
  **if**  $type(node_i) = AND$  **then**  
     $getcrule(node_i);$   
    **if**  $p \neq 0$  **then**  
      **if**  $\forall visitchild(node_i)=1$ {all the child nodes have been visited} **then**  
         $stop(node_i)$   
      **else**  
         $node_i = leftchild(node_i);$ {move to the left unvisited child node}  
         $visit(node_i);$   
      **else**  $stop(node_i);$   
  **else if**  $type(node_i) = OR$  **then**  
     $getcrule(node_i);$   
    **if**  $p=1$  **then**  
       $stop(node_i)$   
    **else if**  $\forall visitchild(node_i)=1$ {all the child nodes have been visited} **then**  
       $stop(node_i)$   
    **else**  
       $p=d;$   
       $node_i = leftchild(node_i);$ {move to the left unvisited child node}  
       $visit(node_i)$   
  **else if**  $type(node_i) = Attribute$  **then**  
    **if**  $Attri(node_i) \in \{u_1, u_2, \dots, u_n\}$  **then**  
       $p=1;$   
       $node_i = parent(node_i);$ {move to the parent node}  
       $visit(node_i);$   
    **else**  
       $p=0;$   
       $node_i = parent(node_i);$   
       $visit(node_i);$   
  }  
function  $stop()$ {  
  **if**  $node_i = root$  **then**  
     $s=p;$   
     $c = first.getcrule(node_i);$   
    return  $s, c;$   
  **else**  $node_i = parent(node_i);$   
   $visit(node_i);$   
}  
**End**

---

$c$  that could be linked to a set of customized QoS values. The traversing method is depth-first traversal in pre-order. The operation is starting from the root node, then traversing the left subtree to the right one. Within the algorithm, if the visiting node is an AND node, all the child nodes of that AND node have to be traversed. If the visiting node is an OR node, the child nodes need to be traversed from left to right, and the visiting pointer will go back to the parent node of that OR node when a satisfied child node was found. If the visiting node is a leaf node as an attribute, the visiting pointer will go back to the parent node when that attribute is an element of the consumer's attribute set. Therefore, the traversing method is starting from the *root*, and ending to the *root*.

The algorithms to support the other two evaluating strategies could be extended by changing the termination conditions. For simplicity, they are not presented in this thesis.

## 5.5 Customized Service Selection

Once the policies of the functionally equivalent service providers are evaluated, PDP could make the decision which services can be accessed by the consumer and the QoS information in the satisfied CRules will be collected by the service QoS collector. All the potential service providers with the customized QoS values are listed for the selection phase. When the consumers make a request, their QoS requirements are presented to the QoS based service selector as described in Chapter 4 as follow:

$$R(Q)=\{R(q^1), R(q^2),...,R(q^n)\}$$

The QoS based service selector is responsible for deciding which service provider is the most suitable one to the consumer's QoS requirement. The similarity based service selection method was introduced in Chapter 4. As we mentioned, in our similarity based selection method, the consumers could present their QoS requests more accurately and flexibly. On each QoS dimension, the consumer's requirement is presented as following tuple:

$$R(q^i)=\langle Name, Negotiability, Willingness, Value \rangle$$

The consumers could present their requirements with two different types: negotiable or nonnegotiable. The willingness represents the tendency of the consumer's preferences for the particular QoS dimension, which makes the requirements more

accurately. Five basic tendencies, such as top, bottom, equivalent, close, and distant, are illustrated to represent consumer's preferences. Once the consumer's QoS requirement  $R(Q)$  and the customized QoS value  $Q_i$  from the service provider, which is introduced in Section 5.3, are collected by the QoS based service selector, the similarity between  $R(Q)$  and  $Q_i$  is computed to generate the similarity score. The service provider with the highest similarity score will be selected and recommended to the consumer.

## 5.6 System Prototype and Demonstration

The system prototype is developed by c++ on the Microsoft Visual Studio. The major functions are: (1) helping service providers to publish their customizable QoS values, and (2) helping service broker to collect the customizable QoS value with the accessibility which is based on the consumer's attributes. To better understand our attribute based service customization and selection approach, we present the following case to demonstrate the system prototype. A consumer Alice is looking for a online movie download service, and she prefers lower cost and higher download speed. In this case, she holds a lot of memberships from different movie download service providers, and she is a UOW student. When she makes a request with her attributes, such as membership ID and student ID, the service broker needs to find the best deal from a lot of available service providers automatically.

Pop Movie is an online movie download service provider, it provides a special deal for the qualified consumers, which are based on their attributes, during the promotional period. In this case, Pop Movie provides the special deal for their members who are either UOW student members or female members. The ABAC policies with CRules of Pop Movie are demonstrated in Figure 5.6. They could input the six main components of the policy: issuer, description, subject, environment, effect, and CRule. The name of policy issuer is considered as the only attribute of resource in this prototype. The general description specifies what the policy will be used for. Pop Movie could define a number of AttriSets which are considered as the nodes in the access control structure in Section 5.3. The type of AttriSet includes AND Set, OR Set, and Attribute, and the policy issue could specify whether the AttriSet is combining with CRule. For example, the attribute set "Female Member" is an AND set, it's parent node is "Special" which is an OR set, and it is combining with a CRule. After all the AttriSets are defined and added into the database, the

**Service Policy Input**

File Edit View

**Issuer:**  
 Pop Movie OK Cancel

**Description:**  
 Pop Movie provides movie download service to the authorized members OK Cancel

**Subject:**  
 Value: Female Member

AttrSet Type: AND Set Parent Node: Special Combining CRule: YES Add

	Value	Type	Parent	Combining CRule
	Special	OR Set	NA	NO
	UOW Member	AND Set	Special	Yes
	UOW ID	Attribute	UOW Member	NO

Tree Build

**Environment:**  
 Type: Date

Valid From: 2 November 2012 To: 1 December 2012 Add

**Effect:**  
 Accessibility: Permit

QoS Information:  
 QoS Set ID: 2

Name: Cost Type: Numerical

Description: The cost of one movie download

Metric: NA

Value: \$5

Type: Single

Domain-independent

Non-negotiable Add

**CRule:**  
 Subject: Choose From AttrSet QoS Set: Choose From QoS R Add

	CRule ID	AttrSet ID	QoS Set ID
▶	C1	A2	Q1
	C2	A5	Q2
*			

Tree View

Figure 5.6: Service Policy Input

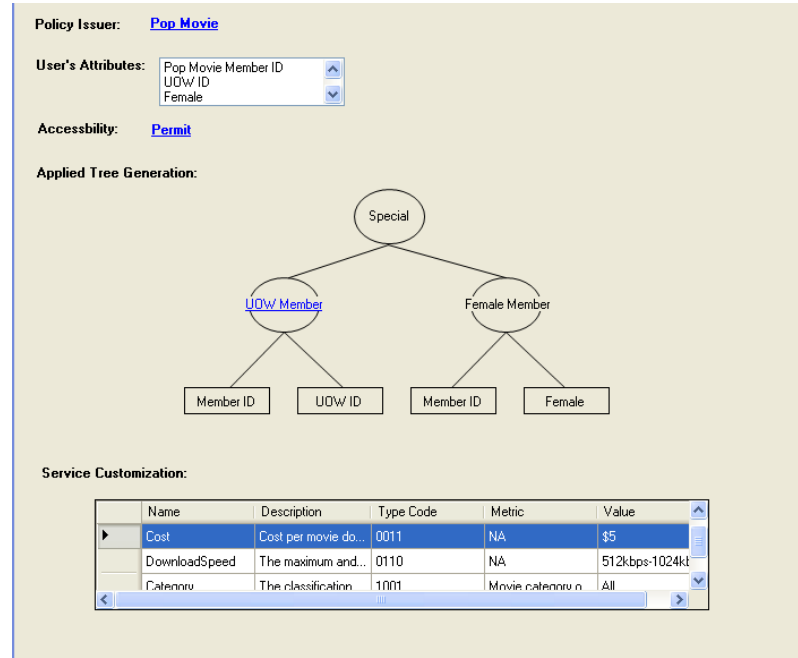


Figure 5.7: Service Customization View

N-Tree structure could be built automatically in the XML based language. In this case, Pop Movie defined the valid date for the policy. The effect includes two main parts: accessability and QoS information. The QoS Set ( $Q_i$ ), which is introduced in Section 5.3, could be defined in the QoS information. For example, the cost with the value of 5 dollars is numerical, single, domain-independent, and non-negotiable. And the CRule is defined as the linkage between the subject and the QoS Set. For example, the CRule “C1” links the attribute set “A2” to the QoS set “Q1”.

Once the consumer Alice made a request with her attributes, the service QoS collector, which is a component of the service broker, will verify the policy statement and collect the customized QoS for her. The result of service customization process is presented in Figure 5.7. The First-In-Applicable evaluating strategy is applied in this case. The QoS Set which is applied on the AttriSet of UOW Member is customized for Alice. The customized QoS information, such as the cost is 5 dollars, and download speed is from 512 kbps to 1024 kbps, etc., in Service Customization field is collected and delivered to the QoS based service selector, and Pop Movie will be a potential candidate in the selection phase.

## 5.7 Summary

This chapter presented an attribute based service customization and selection model to address the need of acquiring the QoS information which is based on consumer's attributes and selecting the demanded services more accurately. The extended attribute based access control policy language has been proposed to publish the additional effects beyond the accessibility; i.e., the customized QoS values by CRules. Different QoS values are provided to different user groups which are distinguished by their attributes. And the service providers expose their QoS information to the consumers who hold the accessibility rather than every entities. The similarity based selection method is adopted to compute the similarity between the consumer's QoS requirements and the customized QoS values.

# Chapter 6

---

## Trust-oriented QoS-aware Composite Service Selection Based on Genetic Algorithm

Chapters 4 and 5 proposed two approaches to addressing the research problems on QoS-aware single service selection. In Chapters 6 and 7, we will turn our attention to the QoS-aware selection approaches for the composite services that compose multiple single services together. In this chapter, we propose a novel trust-oriented QoS-aware selection approach based on genetic algorithm. This chapter is organized as follows. In Section 6.1, we introduce the research problems in the composite service selection area, and list the research contributions of this chapter. Section 6.2 describes the formal definition of service composition architecture for the composite service selection. The trust evaluation of the concrete service composition plan is discussed in Section 6.3. Section 6.4 presents our Trust-Oriented Genetic Algorithm to find a near-optimal service composition with QoS constraints. The experiments are shown in Section 6.5 to illustrate the effectiveness and efficiency of our approach. Section 6.6 is the summary of this chapter.

### 6.1 Introduction

Web service technology provides a promising way to implement web based applications. One of the most important technologies is that various web based applications with complex functions could be implemented by web service composition. That means composing individual web services in accordance with the business process. The benefit of service composition originates from the added value generated by the possible interactions and by their large scale than by the capabilities of its individual services separately. When the service consumers look for their demand services from a large pool, Quality of Service is often critical. However, in addition



to functionality and QoS, the trust aspect, which based on the reputation, becomes more important as a key factor for service selection.

An important part in the composite service selection area is the trust management of service composition. Trust is the measurement of belief on the willingness and ability to achieve some goals in a situation from one entity to another. Trust can be also reflected by the subjective probability if the trust value is presented in the range of  $[0, 1]$  [JIB07a]. Meanwhile, in the composite service environment, trust management becomes more complex because of the different types of service invocation. Therefore, many research problems remain unsolved for trust-oriented and quality driven composite service selection. Firstly, the classical probability theory does not fit for trust evaluation. Instead, subjective probability theory should be adopted for trust evaluation. Secondly, although there are a variety of trust evaluation methods in different areas, no proper mechanism exists for evaluating the global trust of a composite service with a complex structure from the trust values of all service components. Thirdly, taking trust evaluation and the complex structure of composite services into account, effective algorithms are needed for composite service selection, and are expected to be more efficient.

In order to satisfy the users' requirements, how to compose multiple services together as a service composition plan based on the QoS attributes is still a challenge because finding an optimal solution for the multi-dimensional quality driven services selection is NP-hard [CPEV05b]. This problem is concerned as a multi-constraints optimization issue. In the literature, there are some existing studies for service composition and quality driven composite service selection [ASTL07, HMRR08, Men04, XB05, YZL07, ZBD<sup>+</sup>03]. However, taking trust evaluation and the complex structure of composite services into account, effective algorithms are needed for trust-oriented, QoS-aware composite service selection.

In many composite service selection scenarios in the real world, the customers do not have to find the optimum plan. Furthermore, searching an optimum plan is time consuming because it is a multi-objective optimization. Therefore, we adopt the Genetic Algorithm(GA) to find a near-optimal composite service plan efficiently because GA is a powerful tool to solve combinatorial optimizing problems [SP94b]. How to design the genetic algorithm has the great influence on its behavior and performance [RGP<sup>+</sup>02]. In particular, the efficiency and the global astringency of genetic algorithm are determined by the coding scheme of chromosomes and the evolution operators. The special coding scheme should be concerned to solve

different problems. For example, the coding scheme for QoS-aware composite service selection should express not only the services composition but also the information of composition strategies. Furthermore, in the literature, the character of trust has not been taken into account, although the presented genetic algorithms can attain service composition supporting QoS to some extent. Obviously, an advanced trust-oriented genetic algorithm needs to be designed for the composite service selection.

Focusing on the problems of the existing composite service selection approaches, our research contribution in this chapter is listed as follows:

1. We present the formal service composition architecture for composite service selection. We provide the definition of the abstract service composition graph and the concrete service composition graph.
2. We propose the global trust evaluation method for the concrete service composition plans based on the subjective probability theory.
3. We introduce the Trust-Oriented Genetic Algorithm (TOGA) to find a near-optimal composite service plan with QoS constraints. The experiments have been conducted on a composite service that includes a large number of concrete service composition plans, to compare the proposed TOGA with the exhaustive search method.

## 6.2 Service Composition Architecture

The process of the service composition normally includes three main stages: creating the process model, discovering the concrete services, and implementing composite service.

- Creating the process model: the functional representation, service activities and enactment are specified in the process model. It is used to specify the data-flow and control-flow among different sub-tasks. Several process based composition languages have been proposed for web service composition, such as BPEL [Cur03], BPML [Ark02], and WSCI [AAF<sup>+</sup>02]. This stage could be done manually or automatically.
- Discovering the concrete services : in this stage, the concrete (individual) services which accomplish sub-tasks of a service composition are selected and bound.

- Implementing composite service: this refers to delivery of the composite service to service consumers.

According to the sub-architecture for composite service selection which we mentioned in Section 3.3, Figure 6.1 depicts the system model for trust-oriented composite service selection without the consideration of the module of Business Rules Handler. The modules of Trust Manager and GA based Optimization Engine are described in details in Section 6.3 and 6.4, respectively.

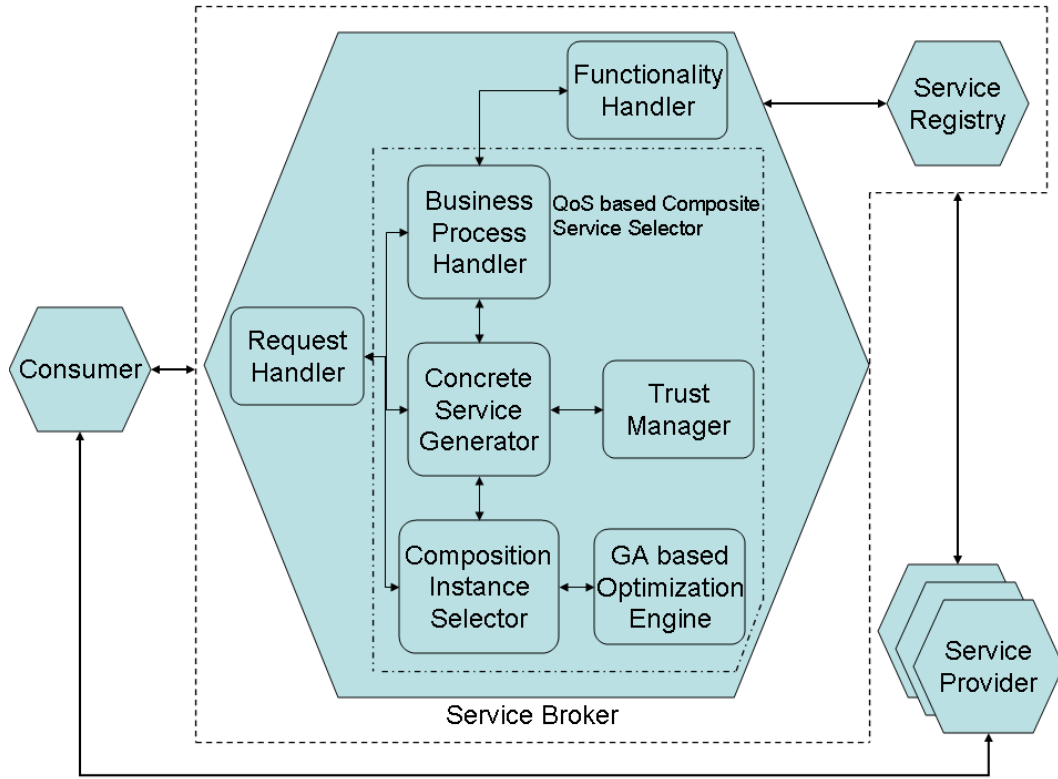


Figure 6.1: The System Model for Trust-Oriented Composite Service Selection

In the following subsections, firstly we introduce the definitions of our Abstract Service Composition Graph and Concrete Service Composition Graph for representing the complex service invocation structures of service composition. In addition, the system notation in this chapter is illustrated. Then, the problem of composite service selection is modeled mathematically. After that, the fitness function which is essential for our genetic algorithm based composite service selection is presented.

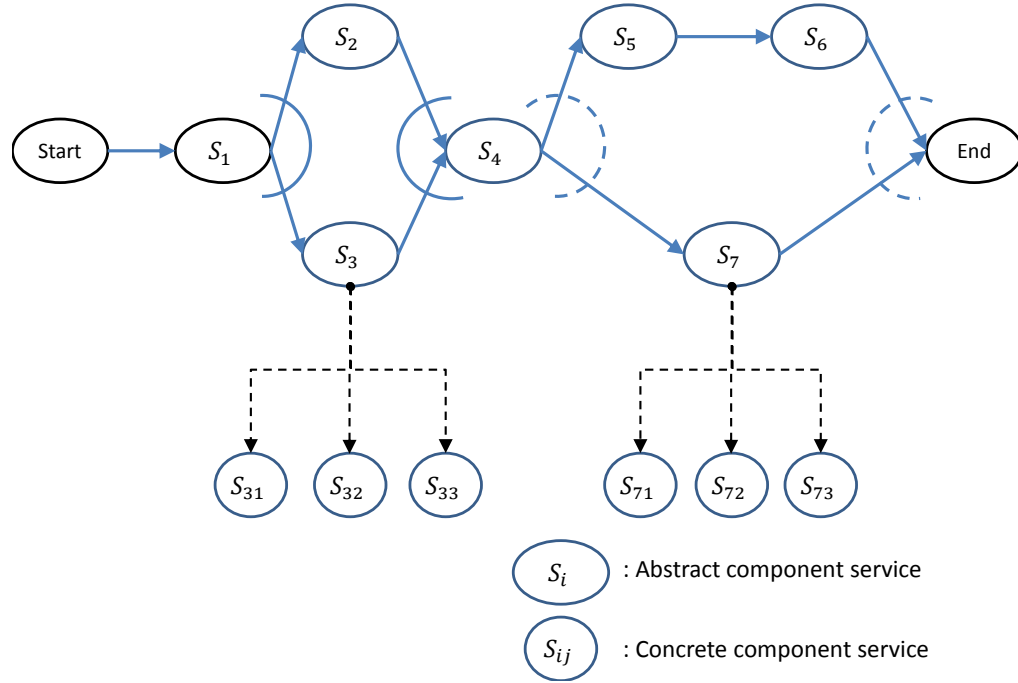


Figure 6.2: Abstract Service Composition Graph

### 6.2.1 Abstract and Concrete Component Service Invocation

A composite service is composed of several abstract services with different invocation relations between them. For every abstract service which is considered as a principle task, multiple concrete services with equivalent functionality could be selected to perform the task. As described in Figure 6.2,  $S_i$  denotes the abstract service as a task in a composite service, and  $S_{ij}$  denotes the  $j^{th}$  concrete service for task  $S_i$ .

The abstract services may be connected by different invocation relations within a composite service. In our study, we consider the six general invocation relations for the process flow: *Sequential Invocation*, *AND Invocation*, *OR Invocation*, *Loop Invocation*, *AND join Invocation*, and *OR join Invocation* which are depicted as follows.

- *Sequential Invocation*:  $S_2$  is the unique succeeding abstract service of  $S_1$  (see Figure 6.3(a)).
- *AND Invocation*: All the abstract services  $S_2 \dots S_n$  are the succeeding services of  $S_1$ , and all of them have to be activated in parallel, so each edge following

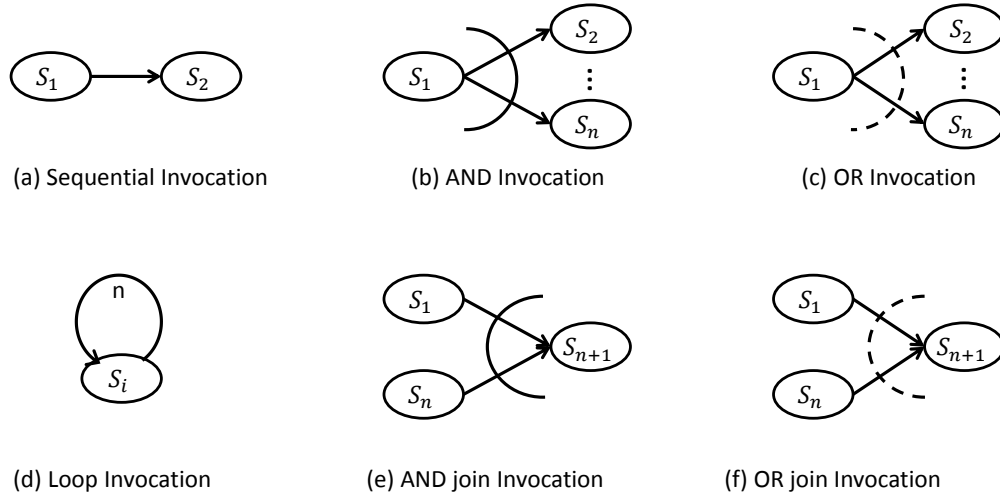


Figure 6.3: General Relations

$S_1$  has a probability of 1 (see Figure 6.3(b)).

- *OR Invocation* All the abstract services  $S_2 \dots S_n$  are the succeeding services of  $S_1$ , only one of them can be activated with a certain probability, and the total probability is 1 (see Figure 6.3(c)).
- *Loop Invocation* The abstract service  $S_i$  invokes itself for  $n$  times (see Figure 6.3(d)).
- *AND join Invocation* An abstract service  $S_{n+1}$  is activated only when all its preceding services  $S_1 \dots S_n$  have been completed (see Figure 6.3(e)).
- *OR join Invocation* An abstract service  $S_{n+1}$  is activated when one of its preceding services  $S_1 \dots S_n$  has been completed (see Figure 6.3(f)).

The abstract service composition graph represents the architecture of a particular service composition, and the definition is presented as follows.

- **Definition 6.1.** The *abstract service composition graph (ACG)* is a directed graph  $G = \langle V, E \rangle$ , where  $V$  is a finite set of vertices, and  $E$  is a finite set of

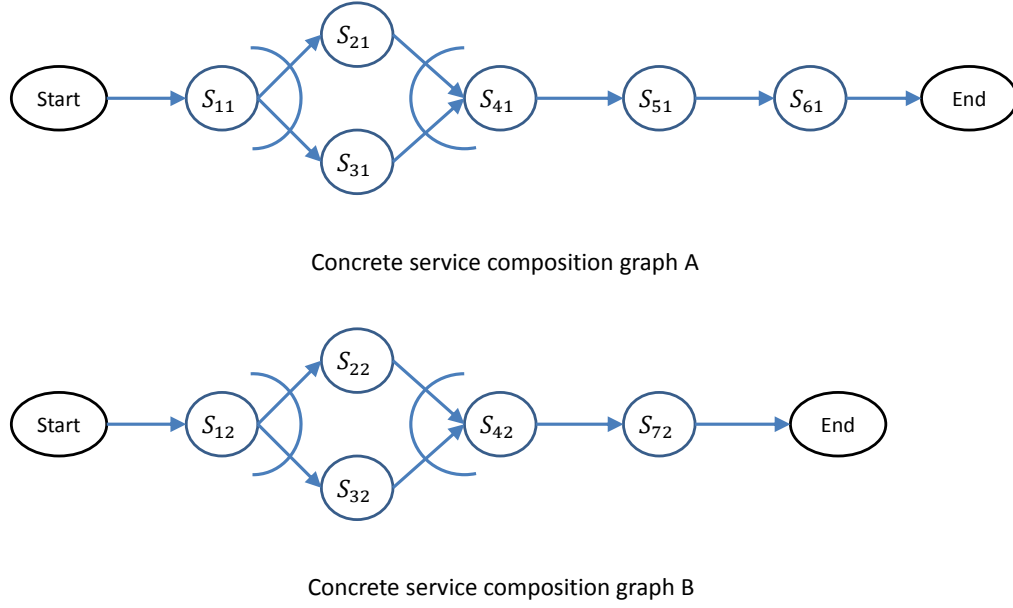


Figure 6.4: Two Concrete Service Composition Graphs

directed edges. For each  $v_i \in V$ ,  $v_i$  represents an abstract component service. For each  $e_j \in E$ ,  $e_j = (v_i, v_{i+1}, r_j)$  is a directed edge between  $v_i$  and  $v_{i+1}$ , where  $v_i$  is the direct predecessor of  $v_{i+1}$ ,  $v_{i+1}$  is the direct successor of  $v_i$ , and  $r_j$  is the invocation relation between  $v_i$  and  $v_{i+1}$ . There is only one  $v_{root} \in V$ , which is the entry vertex without any predecessors, and only one  $v_{terminal} \in V$ , which is the exit vertex without any successors.

As we mentioned before, multiple concrete services could be selected for an abstract service. The concrete service composition graph represents a possible composition plan. For example in Figure 6.4, we present two possible composition plans for the abstract service composition described in Figure 6.2.

Because a concrete service composition graph represents a possible composition plan, every concrete service composition graph is generated from the *ACG*. The definition of the concrete component service graph is presented as follows.

- Definition 6.2. A *concrete service composition graph (CCG)* is a graph  $G' = \langle V', E' \rangle$ , where  $V'$  is a finite set of vertices, and  $E'$  is a finite set of directed

edges. For each  $v'_i \in V'$ ,  $v'_i$  represents an concrete component service. For each  $e'_j \in E'$ ,  $e'_j(v'_i, v'_{i+1}, r'_j)$  is a directed edge between  $v'_i$  and  $v'_{i+1}$ , where  $v'_i$  is the direct predecessor of  $v'_{i+1}$ ,  $v'_{i+1}$  is the direct successor of  $v'_i$ , and  $r'_j$  is the invocation relation between  $v'_i$  and  $v'_{i+1}$ . There is only one  $v'_{root} \in V'$ , which is the same entry vertex in generated ACG without any predecessors, and only one  $v'_{terminal} \in V'$ , which is the same exit vertex in generated ACG without any successors.

### 6.2.2 System Notation

The following notation (Table 6.1) is used in this chapter.

- *Abstract service* ( $S_i$ ): an abstract service is a class of concrete services with the equivalent functionality but different QoS values.
- *Concrete service* ( $s_{ij}$ ): a concrete service providing the functionality of the abstract service  $S_i$ .
- *QoS vector* ( $q_{ij}$ ): every concrete service  $s_{ij}$  is associated with a QoS vector with  $n$  dimensions.
- *Aggregated QoS value* ( $q_g$ ): an aggregated QoS value is computed for a common concrete service plan. Many researchers focused on the aggregation algorithms based on the six basic service invocation relations, which is out of our study scope. We could adopt their aggregation methods for our satisfaction functions described in the following subsection.
- *Trust value of a concrete service* ( $T(s_{ij})$ ): the trust value represents the trust prediction of a service consumer based on the indirect opinions from others and her/his previous direct experience. The trust value is the essential attribute for initializing population group of the first generation in our proposed genetic algorithm.
- *Global trust value* ( $T(g)$ ): a global trust value is the aggregated value of all the concrete services and invocation relations within a particular concrete composite service plan.
- *Fitness function* ( $\mathcal{F}_{fitness}$ ): every single concrete service composition plan has a fitness value which indicates the utility of that plan. The fitness value

$S_i$	Abstract service i
$s_{ij}$	Concrete service j in abstract service i
$q_{ij}$	QoS vector for concrete service $s_{ij}$ with $n$ dimensions, $q_{ij} = [q_{ij}^1, \dots, q_{ij}^n]$
$q_g$	Aggregated QoS value of a concrete service plan with $n$ dimensions, $q_g = [q_g^1, \dots, q_g^n]$
$Q_r$	QoS requirement vector of a concrete service plan, $Q_r = [Q_r^1, Q_r^2, \dots, Q_r^m]$
$T(s_{ij})$	Trust value of a concrete service $s_{ij}$
$T(g)$	Global trust value of a concrete service plan
$F_{fitness}$	Fitness function

Table 6.1: Notations

determines whether the composition plan will be kept in the population or not for our genetic algorithm.

### 6.2.3 Problem Modeling

Based on the service composition architecture, in this chapter, we focus on the composite service selection problem. For a composite service, it is considered that  $n$  abstract services,  $S_1, S_2, \dots$ , and  $S_n$ , are contained in the ACG. For each abstract service  $S_i$ , there are  $l$  concrete services which could perform the required function. In addition, consumers may present some QoS constraints on various dimensions, for example execution time and price. We assume that the consumer acquires a composite service with  $m$  QoS constraints. In fact, the composite service selection problem is how to select the concrete service for each abstract service from the existing candidates in order to satisfy the constraints and maximize the overall QoS of the constructed composite service. From the mathematical point of view, we map the composite service selection with multiple global constraints problem to a Multi-dimension Multi-choice Knapsack Problem (MMKP) which could be formalized as follows:

$$\begin{aligned}
& \max \left\{ \sum_{i=1}^n \sum_{j \in S_i} x_{ij} f_{ij} \right\} \\
& \text{Subject to } \sum_{i=1}^n \sum_{j \in S_i} x_{ij} q_{ij}^k \leq Q_r^k \quad (k = 1, 2, \dots, m) \\
& \sum_{j \in S_i} x_{ij} = 1 \\
& x_{ij} \in \{0, 1\}, i = 1, \dots, n, j \in S_i
\end{aligned} \tag{6.1}$$

where  $x_{ij} \in \{0, 1\}$  are picking variables which satisfy  $\forall i \in \{1, 2, \dots, n\}, j \in S_i$ ,  $\sum x_{ij} = 1$ , which means  $x_{ij}$  is set to 1 if the concrete service  $s_{ij}$  is selected for



the abstract service  $S_i$ , otherwise  $x_{ij}$  is set to 0.  $f_{ij}$  is the QoS satisfaction score which reflects the utility of the  $j^{th}$  concrete service for the  $i^{th}$  abstract service in the composition. As QoS for composite services involves multiple dimensions, the satisfaction score is justified by the fitness value which will be described in the following section.  $q_{ij}=[q_{ij}^1, ..., q_{ij}^m]$  is the QoS of each concrete service  $s_{ij}$  for abstract service  $S_i$ ; the sum of QoS provided by all concrete services must satisfy the overall constraints  $Q_r=[Q_r^1, ..., Q_r^m]$  on all dimensions.

### 6.2.4 Fitness Function

As we mentioned, the goal of the composite service selection algorithms is to select possible concrete service composition plans which meet user's QoS constraints and also provide the best value for the user-centric fitness function  $\mathcal{F}_{fitness}$ . Besides the published QoS of a service, the users become more concerned about the trustworthiness of that service because they want to justify whether the claimed QoS values are trusted or not. Therefore, our proposed  $\mathcal{F}_{fitness}$  function is defined by a weighted sum of the trust prediction value and the aggregated QoS satisfaction value. Suppose a user has  $m$  QoS constraints in a composite service request as  $Q_r = [Q_r^1, Q_r^2, ..., Q_r^m]$ . For each dimension of QoS request,  $Q_r^i$  is presented as  $Q_r^i = \langle Name, Negotiability, Willingness, Value \rangle$  by the Service Request Model which is described in Chapter 4. In this study, the fitness function is defined as follows:

$$\mathcal{F}_{fitness} = \begin{cases} \lambda \cdot T(g) \sum_{i=1}^m w_i f_i(Q_r^i, q_g^i), & \text{if } \prod_{i=1}^m f_i(Q_r^i, q_g^i) = 0 \\ T(g) \sum_{i=1}^m w_i f_i(Q_r^i, q_g^i), & \text{otherwise} \end{cases} \quad (6.2)$$

where  $\lambda$  defines a penalty factor in (0,1) if the composition plan does not meet the QoS constraint.  $T(g)$  denotes the global trust value of a concrete composite service plan.  $w_i$  is the weights for each QoS constraint where  $\sum_{i=1}^m w_i = 1$ .  $q_g^i$  is the global QoS aggregation value of the  $i^{th}$  dimension. The global QoS aggregation value is another main factor. For different invocation relations, we propose the QoS aggregation methods for time, price, availability and reliability in Table 6.2.  $f_i$  is the satisfaction function on the  $i^{th}$  dimension which depends on the *Willingness* of QoS requirement. In the Service Request Model, the *Willingness* of user's preference

QoS Attribute	Sequential Invocation	AND Invocation	Loop Invocation
Time	$\sum_{i=1}^n T(S_i)$	$Max\{T(S_1), \dots, T(S_j)\}$	$k \cdot T(S_i)$
Price	$\sum_{i=1}^n P(S_i)$	$\sum_{i=1}^n P(S_i)$	$k \cdot P(S_i)$
Availability	$\prod_{i=1}^n A(S_i)$	$\prod_{i=1}^n A(S_i)$	$A(S_i)^k$
Reliability	$\prod_{i=1}^n R(S_i)$	$\prod_{i=1}^n R(S_i)$	$R(S_i)^k$

Table 6.2: QoS Aggregation Methods

is classified into five categories: *Top*, *Bot*, *Equivalent*, *Close*, and *Distant*, they represent the tendency of the service consumer's preferences for the particular QoS requirement dimensions. The satisfaction function  $f_i(Q_r^i, q_g^i)$  is defined as follows:

$$f_i(Q_r^i, q_g^i) = \begin{cases} \frac{Q_r^i - q_g^i}{Q_r^i}, & \text{if } Q_r^i > q_g^i \\ 0 & , \text{otherwise} \end{cases}, \text{with } Willingness(i) = Top \quad (6.3)$$

$$f_i(Q_r^i, q_g^i) = \begin{cases} \frac{q_g^i - Q_r^i}{q_g^i}, & \text{if } Q_r^i < q_g^i \\ 0 & , \text{otherwise} \end{cases}, \text{with } Willingness(i) = Bot \quad (6.4)$$

$$f_i(Q_r^i, q_g^i) = \begin{cases} 1, & \text{if } Q_r^i \equiv q_g^i \\ 0, & \text{otherwise} \end{cases}, \text{with } Willingness(i) = Equivalent \quad (6.5)$$

$$f_i(Q_r^i, q_g^i) = \frac{|Q_r^i - q_g^i|}{\max\{Q_r^i, q_g^i\}}, \text{with } Willingness(i) = Close/Distant \quad (6.6)$$

## 6.3 Trust Evaluation for Concrete Service Composition Plans

In this section, we introduce the trust evaluation methods for the concrete service composition plans. On one hand, we estimate the trust value of every single concrete service within a composition plan from a series of ratings according to Bayesian inference [LW10], an important component in subjective probability theory [JIB07b]. On the other hand, we present the global trust computation model to estimate the aggregated trust value of a concrete service composition plan based on the trust value of all concrete services within the plan.

### 6.3.1 Trust value estimation for a concrete service

The trust value of a concrete service reflects the personal trust prediction which consists of the direct experience of the service consumer and the indirect experience from others. The indirect trust ratings can be collected based on honest-feedback-incentive mechanisms [JF06, JF07]. Because every single rating, which is a value in  $[0, 1]$ , represents the subjective judgement of the personal belief on a certain event, the subjective probability theory is the right tool for dealing with trust ratings.

Subjective probability logic is compatible with Bayesian inference reputation systems which combines with belief and trust reasoning [JQ09]. The goal of adopting Bayesian inference is to summarize the available information that defines the distribution of trust ratings through the specification of probability density function, such as prior distribution and posterior distribution [LW10]. The prior distribution indicates the subjective opinion about the trust before obtaining the rating samples. The prior distribution can be updated as soon as the sample is obtained. The updated trust rating is called the posterior distribution. According to [HPZ06], if all service clients give ratings for the same service, the provided ratings conform to normal distribution. Therefore, the trust value of a concrete service  $s_{ij}$  with  $n$  ratings  $x_1, x_2, \dots, x_n$  ( $x_\alpha \in [0, 1]$ ) based on the Bayesian estimation is

$$T(s_{ij}) = \frac{\bar{x} + \beta}{2} = \frac{\sum_{\alpha=1}^n x_\alpha + n\beta}{2n} \quad (6.7)$$

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

Since the trust value estimation for a concrete service is based on the subjective probability theory we discussed before, it is more reasonable to introduce the user's direct experience about the trust. If the user has no prior experience about the trust of the concrete service, by default, let  $\beta = \frac{1}{2}$  since  $\frac{1}{2}$  reflects the neutral belief between trust and distrust.

### 6.3.2 Global trust value for a concrete service composition plan

The global trust value of a concrete service composition plan is determined by the trust values of concrete services and invocation relations between them. According to Definition 6.2, for a CCG, we only need to consider three kinds of relations: *Sequential Invocation*, *AND Invocation*, and *Loop Invocation*. *Loop Invocation* could

be considered as *Sequential Invocation* for  $n$  times. Thus, two kinds of invocation relations should be defined for computing the global trust value of a possible composition plan.

For the *Sequential Invocation* relation, the global trust value  $T_g$  can be computed by

$$T_g = T(s_{1j}) * T(s_{2j}) \quad (6.8)$$

where  $T_{s_{1j}}$  and  $T_{s_{2j}}$  are the trust values of concrete service  $s_{1j}$  and  $s_{2j}$  respectively, which are evaluated from Equation 6.7. Because  $s_{1j}$  and  $s_{2j}$  are independent as single services, the probability that  $s_{1j}$  and  $s_{2j}$  both occur equals to the product of the probability that  $s_{1j}$  occurs and the probability that  $s_{2j}$  occurs.

For the *AND Invocation* relation, the global trust value  $T_g$  can be computed by

$$T_g = \min\{T(s_{1j}), T(s_{2j}), \dots, T(s_{nj})\} \quad (6.9)$$

where  $n$  is the number of *AND invocation* splits. Since each *AND invocation* split has the same probability of 1 as we described before,  $T_g$  reflects the worst case of  $n$  parallel *AND invocation*.

According to Equation 6.8 and 6.9, a CCG consisting of *Sequential Invocation*, *AND Invocation*, and *Loop Invocation* can be converted to a single vertex with its trust value computed as the global trust. In order to obtain the global trust value of a CCG, firstly the trust value of each atomic *Sequential Invocation* relation in the CCG should be computed by Equation 6.8. Then, each computed atomic *Sequential Invocation* relation is taken as a vertex in the CCG. After that, each *AND Invocation* relation is computed by Equation 6.9 as a vertex as well. Finally, by repeating the computation, the CCG is simplified as a vertex, and the global trust value is obtained.

## 6.4 Genetic Algorithm for Service Selection

### 6.4.1 Algorithm overview and assumptions

For an abstract service composition, there are multiple choices as concrete service composition plans. With different kinds of invocations, composite service selection with QoS constraints can be modeled as the Multi-dimension Multi-choice Knapsack Problem which is a NP-Complete problem [CPEV05b]. The Genetic Algorithm is one of the techniques for processing NP-complete problems.

The Genetic Algorithm is a search heuristic that mimics the ideas of natural selection and evolution. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques basically inspired by *selection*, *crossover* and *mutation*. Our proposed advanced genetic algorithm represents an intelligent exploitation of a trust-oriented random search within a defined search space to find a near-optimal concrete service composition plan. The fitness function  $\mathcal{F}_{fitness}$  described before is the criteria for certain selection. And the trust value of a concrete service ( $T(s_{ij})$ ) reflects the probability that the concrete service ( $s_{ij}$ ) is selected in the initialized population. The overview of the execution flow of our genetic algorithm is shown in Figure 6.5. The algorithm starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness value of every individual in the population is evaluated, the more fit individuals are selected from the current population, and each individual is modified to form a new population which is used in the next iteration of the algorithm. After the first generation is initialized, the crossover and the mutation operations are invoked with the certain probabilities. Crossover operation is analogous to reproduction and biological crossover. It is a process of taking more than one parent individuals and producing a child individual from them. Mutation operation is used to maintain genetic diversity from one generation to the next. The algorithm will be terminated by the stopping criterions such as fixed number of generations, computation time, and the highest ranking solution's fitness has reached a plateau.

In our trust-oriented genetic algorithm, for computing the trust values of concrete services, we assume that a large volume of services with their ratings are stored by the third party trust management authority. The users could present their QoS constraints when they request the composite services. For the QoS aggregation, we only focus on the numerical QoS attributes in this chapter. The details of our trust-oriented genetic algorithm is shown in Algorithm 2. The inputs of the algorithm are a set of abstract services  $S_i$ , a set of concrete services  $s_{ij}$  with the trust values  $T(s_{ij})$ , the consumer's QoS requirement  $Q_r$  with the weights of each dimension  $w_i$ , the published QoS information of each concrete services  $q_{ij}$ , and the max generation  $l$  which is applied as the stopping criteria. In the first generation,  $N$  individuals are picked as the initialized population based on their trust values. The initialization policy will be further described in Section 6.4.3. Then, the fitness value of each individual is computed. After that, a random number  $r \in (0, 1)$  is used to

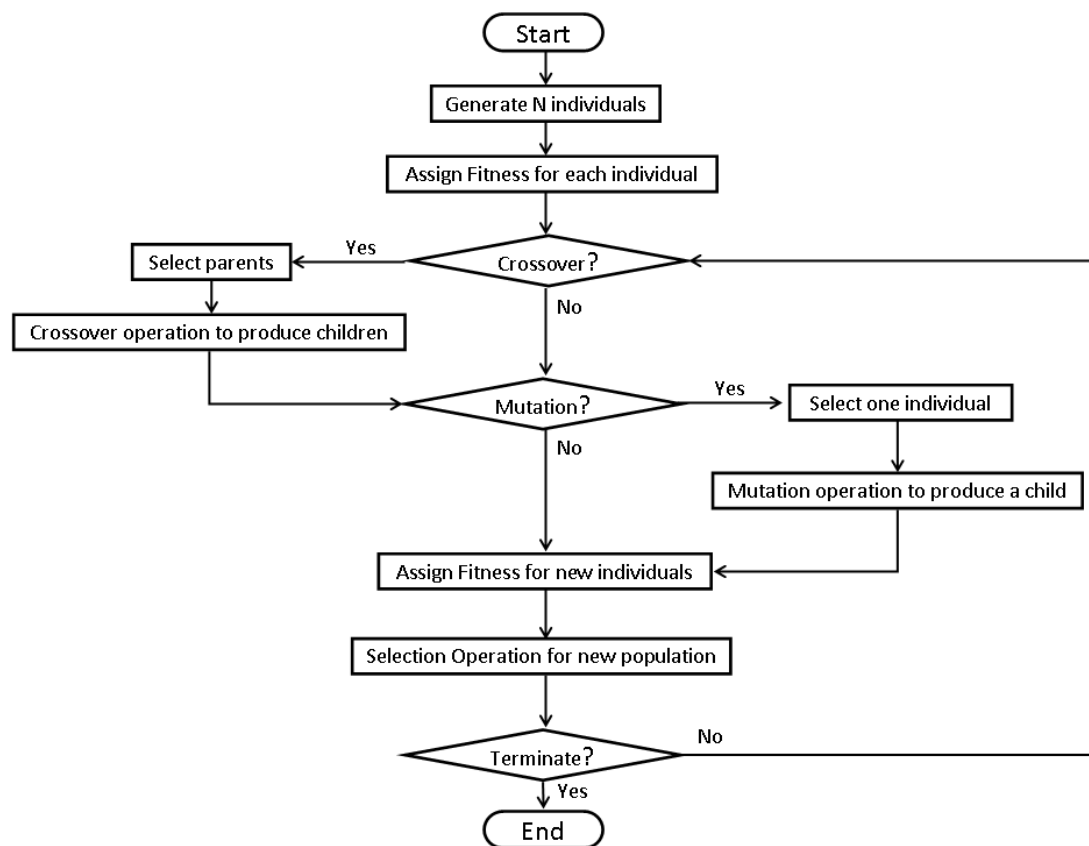


Figure 6.5: Overview of Genetic Algorithm

determine whether the crossover operation will be applied on that generation. If the random number  $r$  is equivalent or less than the crossover operation bound parameter  $P_{crossover}$ , the new individuals will be generated by the crossover operator and added to the population. Otherwise the crossover operation will not be applied. The crossover operation is used to vary the chromosomes from one generation to the next. The algorithm convergence depends on the method of the crossover operation. The triggering condition of the mutation operation is similar to that of the crossover operation. The mutation operation is used to maintain the diversity of the chromosomes. The details of our crossover and mutation policies are described in Section 6.4.3. After the crossover and mutation operations are applied, the fitness value of each new individual is computed. Finally, all the fitness values are sorted, and the better  $N$  individuals will be selected for the next generation. After the population evolves  $l$  generations, the algorithm will output the optimal individual  $I_{optimal}$  as the best concrete service composition plan.

Given a fitness function, the composite service selection problem is to select the optimal concrete service composition plan which receives the largest  $\mathcal{F}_{fitness}$  value and meets all QoS constraints. The selection problems considered in this paper are divided into two categories.

1. *Abstract service composition with the sequential invocation flow.* In this category, all the abstract services are constructed by the *Sequential Invocation* relation.
2. *Abstract service composition with the general invocation flow.* In this category, the relationships among abstract services of a composition may contain the six general invocation relations as described in Figure 6.3.

### 6.4.2 Genetic algorithm for sequential invocation flow

For a service composition which has  $n$  abstract services, we could build the chromosomes (individuals) which are constructed by the invocation order in a length of  $n$ . For each abstract service  $S_i$ , there are  $j$  concrete services with  $m$  QoS constraints which are presented by the user. The example of a chromosome is shown in Figure 6.6 which  $n = 7$  and  $j = 5$ .

Obviously, there are  $5^7$  concrete service composition plans. Unlike the normal genetic algorithm, we select a concrete service  $s_{ij}$  for a chromosome by using roulette

**Algorithm 2:** Genetic Algorithm for Composite Service Selection

**Data:**  $ACG < V, E, R >$  with  $n$  abstract services  $\{S_i \mid 1 \leq i \leq n\}$ ;  
Concrete services  $\{s_{ij} \mid 1 \leq i \leq n, 1 \leq j \leq n\}$ ;  
 $T(s_{ij}) \in [0, 1]$ : the trust value of each  $s_{ij}$ ;  
Qos constraint conditions  $Q_r$ ;  
Constraint weight  $w_i$ ;  
 $q_{ij}$ : Qos vector for  $s_{ij}$  ;  
Max generation  $l$ .

**Result:** The optimal individual  $I_{optimal}$ .

**begin**

```

    Generate  $N$  individuals  $(I_1, I_2, \dots, I_N)$ ;

    //Compute fitness value for each individual//;
    for each  $I_i$  do
        |  $Compute\_Fitness(I_i)$  by formula(6.2);
    for all generations  $Gen$  that  $1 \leq Gen \leq l$  do
        | //Crossover//;
        | Pick a random  $r \in (0, 1)$ ;
        | if  $r \leq P_{crossover}$  then
        | |  $Crossover\_operation()$ ;
        | | send the new individuals  $I_\alpha \rightarrow New\_population$ ;
        | else
        | | send  $null \rightarrow New\_population$ ;

        | //Mutation//;
        | Pick a random  $u \in (0, 1)$ ;
        | if  $u \leq P_{mutation}$  then
        | |  $Mutation\_operation()$ ;
        | | send a new individual  $I_\beta \rightarrow New\_population()$ ;
        | else
        | | send  $null \rightarrow New\_population$ ;

        | //Compute fitness value for new individuals//;
        |  $Compute\_Fitness(I_\alpha)$ ;
        |  $Compute\_Fitness(I_\beta)$ ;
        | //Sorting the individuals according to their fitness values//;
        |  $Sort(I_\alpha, I_\beta, I_1, I_2, \dots, I_N)$ ;
        | Select the better  $N$  individuals;

    return the best individual  $I_{optimal}$ ;

```

**end**



$S_{12}$	$S_{23}$	$S_{34}$	$S_{43}$	$S_{52}$	$S_{61}$	$S_{75}$
----------	----------	----------	----------	----------	----------	----------

Figure 6.6: An example of chromosome

wheel method which adopts the trust value  $T(s_{ij})$  as the probability. This means a concrete service with higher trust value has higher probability to be chosen in the first generation. Thus,  $N$  chromosomes are selected randomly but conditionally for the initialized population.

The fitness function mentioned above is used to select some better plans in the current generation. Two-points crossover is performed to produce the next generation, and mutation is also carried out according to the given probability. An example of the crossover and mutation operations is shown in Figure 6.7. In the crossover operation, all the concrete services between the two points are swapped between the parent chromosomes to render two child chromosomes. In the mutation operation, the mutated concrete service is selected by “roulette wheel” method based on the trust value, to replace the current concrete service.

The evolution will continue until the max number of generations is searched or when the best composition plan in the current generation remains unchanged for a given number of generations. The resulting concrete service composition plan is the near optimal composition plan for the service composition.

### 6.4.3 Genetic algorithm for general invocation flow

Many real-world composite services have services that are not in strictly sequential order. They may have *AND Invocations*, *OR Invocations*, and *Loop Invocations*. In order to simplify the problem and construct a concrete service composition plan as a CCG, we first remove the *Loop Invocation* by unfolding the cycles as in [ZBN<sup>+</sup>04]. A *Loop Invocation* is unfolded by cloning the *Sequential Invocation* as many times as the maximum loop count.

Depending on the number of *OR Invocation* and *Loop Invocation* in an ACG, there might be more than one execution strategy that is generated from the task flow of the abstract service composition. For example, two possible composition plans with different execution strategies are shown in Figure 6.4. Meanwhile, the

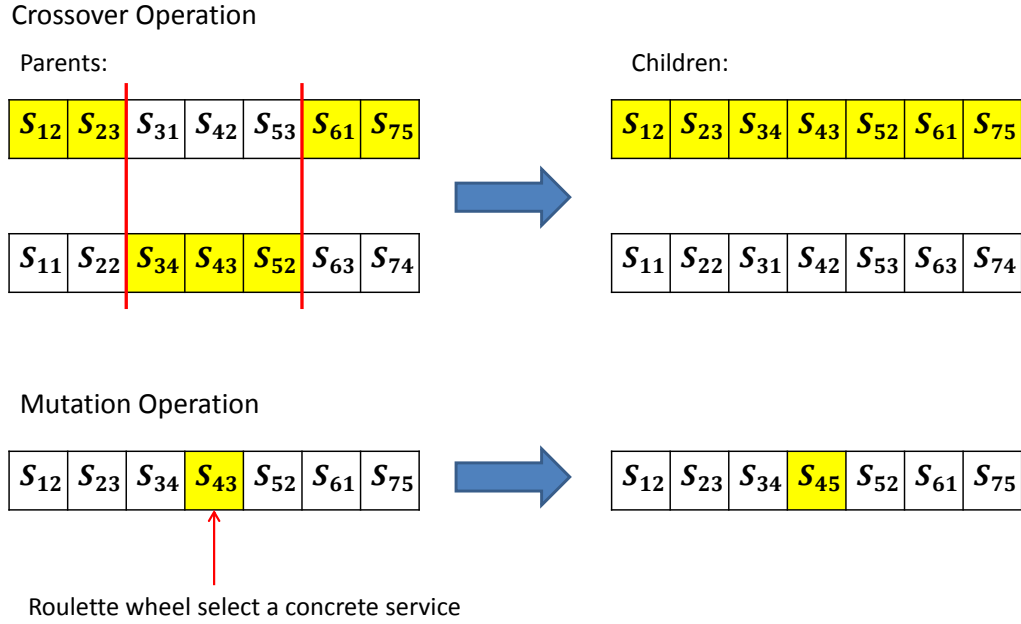


Figure 6.7: An example of crossover and mutation operations

length of the chromosome in our model depends on the number of concrete services which are activated in a concrete service composition plan. In order to operate the crossover and mutation functions, we need to normalize all the chromosomes into the same length. Thus, we combine a binary stream which can present the execution strategy with a chromosome. The chromosomes are structured from the ACG by following rules:

- *Overall Containment*: all the abstract services must be contained as elements in a chromosome, and the length of a chromosome is the same as the number of abstract services within the ACG.
- *Unique Structure*: all the elements of each chromosome are ordered by the unique execution flow from *Start* to *End*.
- *Sequential Order*: all the successors with *Sequential Invocation* must be the next element in a chromosome.
- *Closed Circuit*: all the successors with *AND JOIN Invocation/ OR JOIN*

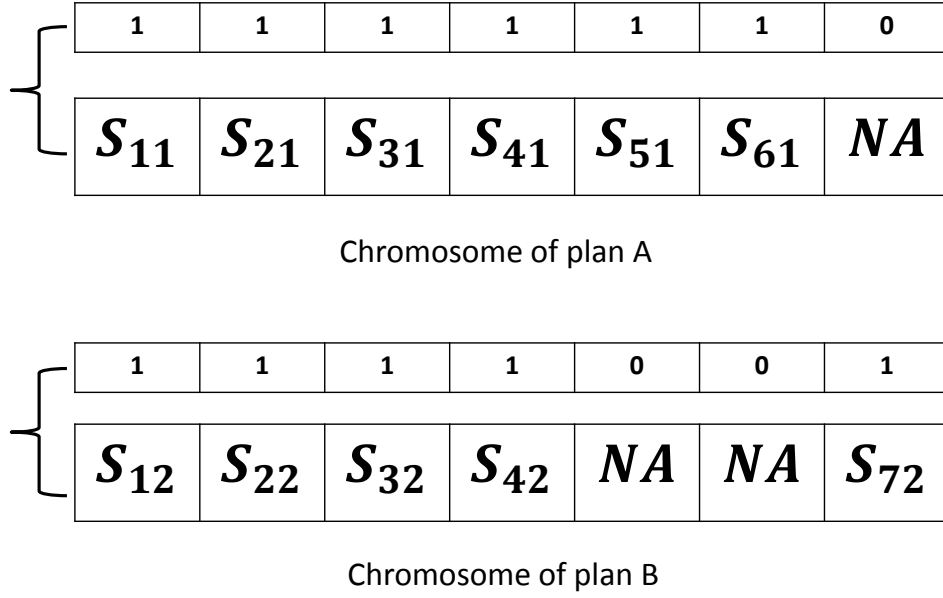


Figure 6.8: A example of encoded chromosomes

*Invocation* is the next element only when all the split paths are encoded from the prior *AND Invocation/OR Invocation*.

The example of two chromosomes from different concrete service composition plans in Figure 6.4 is shown in Figure 6.8. If the abstract service from the ACG is executed in the particular chromosome, the element of that abstract service in the binary stream will be 1, otherwise the element is 0 because the execution flow does not include that abstract service. The different execution strategies will impact the functions of crossover and mutation which is discussed below.

### Initialization Policy

There may be a lot of different execution strategies for the different concrete service composition plans as explained before. For each execution strategy, it has a probability ( $\rho_i$ ). Therefore,  $\sum_{i=1}^K \rho_i = 1$  if there are K possible execution strategies in a composite service. For example, there are 2 execution strategies in Figure 6.2:

*Execution Strategy 1*:  $\{S_1, S_2, S_3, S_4, S_5, S_6\}$ , with a probability of  $\rho_1$ ; *Execution Strategy 2*:  $\{S_1, S_2, S_3, S_4, S_7\}$ , with a probability of  $\rho_2$ ;

and  $\rho_1 + \rho_2 = 1$ .

In order to find the global optimized solution and avoid the situation of “prematurity”, we need to optimize the initial population first. The quality of the chromosomes in the initial population will impact the final result [ZSC06]. The initialization policy is proposed as follows.

Firstly, it is necessary to confirm the proportion of chromosomes for every execution strategy to the size of the population. The method of calculating the probability ( $\rho_i$ ) of every strategy is to estimate the proportion of concrete service composition plans of one execution strategy to the sum of all possible composition plans of all strategies. The more the number of composition plans of one strategy is, the more chromosomes for the strategy are in the initial population.

Secondly, the value of every concrete service in chromosome is confirmed according to a trust-oriented optimized method. The larger the trust value of a concrete service is, the larger the probability that the service will be selected for the abstract service is. The probability that one concrete service will be selected is the result of the trust value itself divided by the sum of trust values of all concrete services for the same abstract service. The “roulette wheel selection” is the mechanism to select concrete services for every task.

The result of the above policy shows that most of the chromosomes in the initial population have potentially high fitness based on their trust values. The special initial population becomes the basis to get higher fitness during the later evolution.

### Crossover and Mutation Policy

In the standard genetic algorithm, crossover operator is used to vary chromosomes from one generation to the next, and the parent chromosome is selected with a probability based on the fitness value. However, in the composite service selection scenario, the parent composition plans cannot produce a child plan if designed parents have different execution strategies. The crossover policy works as follows. Before crossover operation, the binary streams in Figure 6.8 are used to confirm whether the execution strategies are the same or not. Only the chromosomes with the same execution strategy can be processed in the crossover operation.

Mutation operation is used to maintain genetic diversity from one generation of chromosomes to the next. In the composite service selection scenario, the composition plan after mutation may not be a possible path. Therefore, we need to check

the invocation relations of the successor between mutated concrete service and current concrete service. If the successor is different which means the mutated concrete service can not activate the current successor, the mutation operation will be failed.

## 6.5 Experiments

To verify the effectiveness and efficiency of the proposed Trust-Oriented Genetic Algorithm (TOGA), simulation comparison has been performed on Trust-oriented QoS-aware composite service selection.

### 6.5.1 Experiment settings

All experiments were taken on the same software and hardware, which used Matlab 7.0.0.19920(R14) running on a Dell Optiplex 745 desktop with an Intel Core 2 Duo 6400 2.13GHz CPU and a 2GM RAM, Windows XP Pro. The stopping criterion is set by fixed generation: 20, 40, 60, 80, 100, 150, and 200. The number of repetition times for each simulation is set of 20 respectively. The service composition size is considerably large. There are 4 abstract strategies in the case, average 5 abstract services in every strategy, and average 15 concrete services for each abstract service. Thus, the number of concrete service composition plans is  $4 \times 15^5$ . The concrete services for each task are created with one ID, one local trust degree, and values of two QoS properties such as execution time and cost. Additionally, the global users constraints of the QoS properties are provided for every concrete composition plans.

Our experiments of TOGA are set up with the following details: the population size is 20, the crossover operation probability is 0.7, and the mutation probability is 0.1. The selection mechanism of individuals is the roulette wheel selection. The global QoS aggregation method in Section 6.2 is used for them. The penalty technique is used for constrained optimization problems in algorithms.

### 6.5.2 Effectiveness of TOGA

According to our TOGA, the fitness value of every concrete service composition plan reflects the utility which can be calculated. As the subjective trust degree from the user and the objective QoS values of the service are not uniform distributed, there are two properties for reflecting the utility of our TOGA: optimality and accuracy. The optimality of a concrete service composition plan is computed as:

$$O(f_i) = \frac{f_i}{\text{Max}(f_i)} \quad (6.10)$$

And the accuracy of a concrete service composition plan is computed as:

$$A(f_i) = \frac{\text{Number of worse plans}(which f_j < f_i)}{\text{Total number of composition plans}(N)} \quad (6.11)$$

From the equations above, we can observe that  $O(f_i)$  values are distributed in  $[0, 1]$ . The corresponding histogram of  $O(f_i)$  values is plotted in Figure 6.9, we can observe that 87% of values are less than 0.6. Thus, if we select a concrete service composition plan randomly, it is very likely to obtain a composition plan with a low utility value.

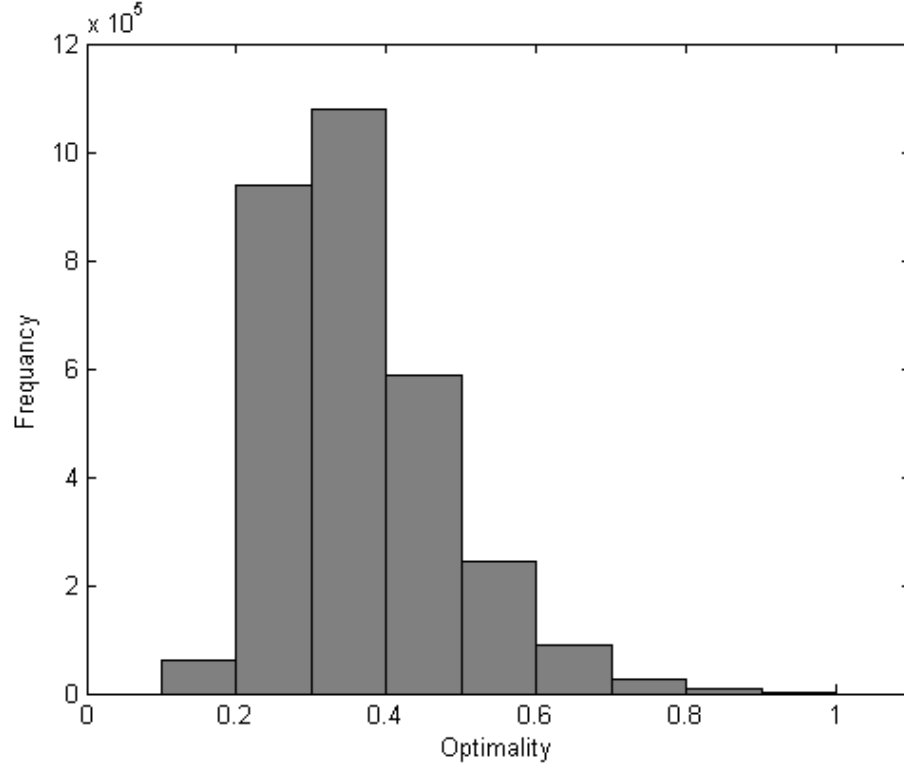


Figure 6.9: Histogram of  $O(f_i)$  for each composition plan

In this experiment, we compare our proposed TOGA with the exhaustive search method which is inefficient as it aims to enumerate all solutions. The  $A(f_i)$  values are calculated by the exhaustive search method in order to test the utility of our

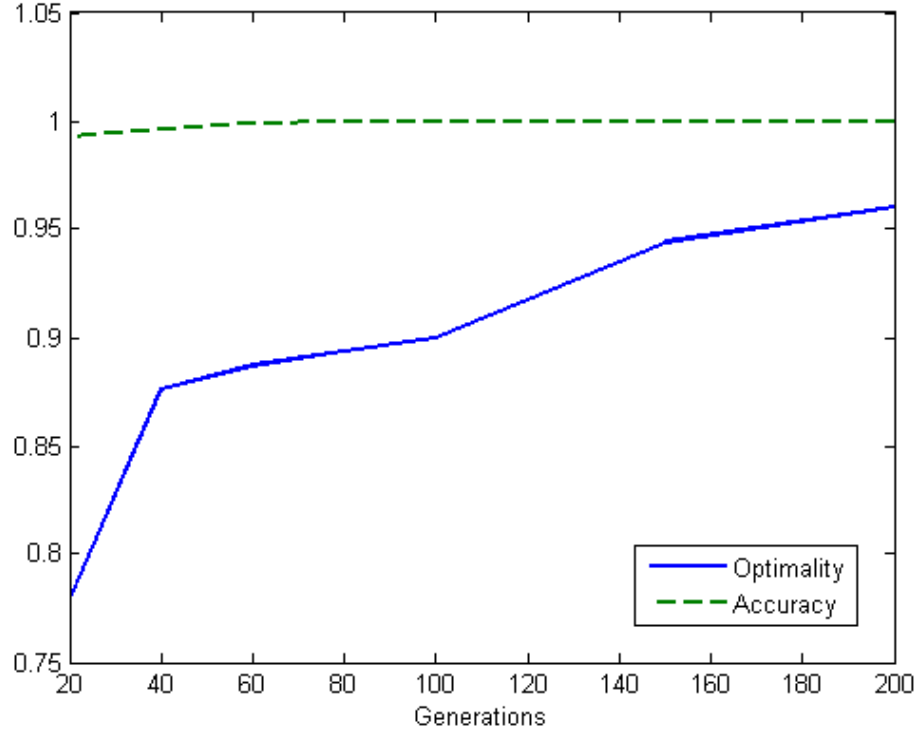


Figure 6.10: TOGA utility with different generations

TOGA. The following is about how these simulation data are produced. The numbers of generation are set as 20, 40, 60, 80, 100, 150, and 200 respectively. The simulation times are 20 for every round of the experiment based on the different initialized populations which are created follow by the initialization policy. The average  $O(f_i)$  and  $A(f_i)$  values are recorded in Table 6.3. From Figure 6.10, we can see that the average accuracy is 99.99% and the optimality is 89.35% when there are 80 generations, which means the effectiveness of TOGA is high within a reasonable time. Theoretically, the larger the population size is, the more effective for our TOGA. Our experimental population size is relatively small where 20 concrete service composition plans from  $4 \times 15^5$  composition plans. It has a great chance to obtain the near-optimal one when the population size is larger than 20.

The effectiveness of TOGA is also tested on different experimental settings. First of all, the number of services is increased. For 10 abstract services, there are 15 single services in each abstract service. It means that the total number of concrete service composition plan is  $4 \times 15^{10}$ . The experiment is also tested on 15 abstract services,

Generation	20	40	60	80	100	150	200
Average $O(f_i)$	77.89%	87.61%	88.72%	89.35%	89.99%	94.42%	96.05%
Average $A(f_i)$	99.27%	99.61%	99.90%	99.99%	99.99%	99.99%	99.99%
CPU time (s)	0.89	1.65	2.25	3.06	4.05	7.02	10.62

Table 6.3: Experiment records

which means the total number of concrete service composition plan is  $4 \times 15^{15}$ . To analyze the results on different number of services, the population size is set as 20, and the mutation probability is 0.1. The stopping criterion is set by fixed generation: 20, 40, 60, 80, 100, 150, and 200. The number of repetition times for each simulation is set of 20 respectively. The average  $O(f_i)$  on different number of services are recorded in Table 6.4.

Number of Composition plans	20	40	60	80	100	150	200
$4 \times 15^5$	77.9%	87.6%	88.7%	89.4%	90.0%	94.4%	96.1%
$4 \times 15^{10}$	58.7%	66.5%	70.1%	74.1%	79.6%	86.9%	92.1%
$4 \times 15^{15}$	36.8%	39.5%	45.9%	47.8%	54.9%	66.4%	67.3%

Table 6.4: Average  $O(f_i)$  on different number of composition plans

In addition, in order to estimate the effectiveness of TOGA, the mutation probability is set as 0.05, 0.1, 0.2, and 0.3 respectively. In this group of experiments, the number of concrete service composition plan is set as  $4 \times 15^{10}$ , and the population size is set as 20. The average  $O(f_i)$  on different mutation probability are recorded in Table 6.5.

Mutation probability	20	40	60	80	100	150	200
0.05	53.5%	61.5%	64.4%	66.2%	69.0%	79.8%	84.6%
0.1	58.7%	66.5%	70.1%	74.1%	79.6%	86.9%	92.1%
0.2	60.9%	73.3%	81.0%	85.3%	90.5%	94.1%	96.1%
0.3	67.5%	79.4%	84.8%	90.5%	93.0%	96.8%	98.9%

Table 6.5: Average  $O(f_i)$  on different mutation probability

Furthermore, the effectiveness of TOGA is tested on different setting of population size. In this group of experiments, the number of services is set as  $4 \times 15^{10}$ , and the mutation probability is 0.1. The population size is estimated on 20, 100, 500,



and 1000 respectively. The average  $O(f_i)$  on different population size are recorded in Table 6.6.

Population size	20	40	60	80	100	150	200
20	58.7%	66.5%	70.1%	74.1%	79.6%	86.9%	92.1%
100	60.2%	78.7%	83.7%	85.4%	88.3%	93.0%	96.2%
500	64.2%	80.9%	86.0%	89.5%	92.0%	95.8%	98.2%
1000	72.8%	80.9%	89.0%	92.0%	94.5%	96.0%	98.2%

Table 6.6: Average  $O(f_i)$  on different population size

From Table 6.4, it is clear that the average  $O(f_i)$  is decreased when the number of concrete service composition plan increases from  $4 \times 15^5$  to  $4 \times 15^{15}$ . With a small population size which is set as 20, a near-optimal solution could also be found within 200 generation when the service pool is as large as  $4 \times 15^{10}$ . However, when the number of concrete service composition plan increases to  $4 \times 15^{15}$ , the average  $O(f_i)$  will decrease to 67.3%, which means the final result is not considered as near-optimal. Therefore, other parameters need to be adjusted for TOGA to find a near-optimal solution.

From Table 6.5, the result shows that the average  $O(f_i)$  increases when the mutation probability increases from 0.05 to 0.3. Furthermore, when the mutation probability increases, the near-optimal concrete service composition plan could be found within a less generation time. From Table 6.6, the result represented that the average  $O(f_i)$  increases when the initial population size becomes larger from 20 to 1000. And the near-optimal concrete service composition plan could be found within a less generation time. In addition, the population size is still relatively small where 1000 concrete service composition plans from  $4 \times 15^{10}$  composition plans.

Therefore, from the experimental result, it shows that the effectiveness of TOGA is relative to the setting of initial population size and the mutation probability. If the setting is not well adjusted, the effectiveness of TOGA would decrease when the number of concrete service composition plan increases. As a result, if the number of concrete service composition plan becomes larger, it is suggested that the population size needs to be adjusted larger, or the mutation probability is set to a higher level.

### 6.5.3 Efficiency of TOGA

Theoretically, the time complexity of the exhaustive search method for composite service selection is  $O(n^m)$ , where  $n$  is the average number of concrete services for an abstract service, and  $m$  is the average number of abstract services for a composition plan. While the time complexity of our TOGA is  $O(N \log l)$ , where  $N$  is the number of generation, and  $l$  is the initial population size. Obviously, the execution time of TOGA depends neither on the number of abstract services nor the number of concrete services. It is only affected by the generation times and the population size which are under our adjustments.

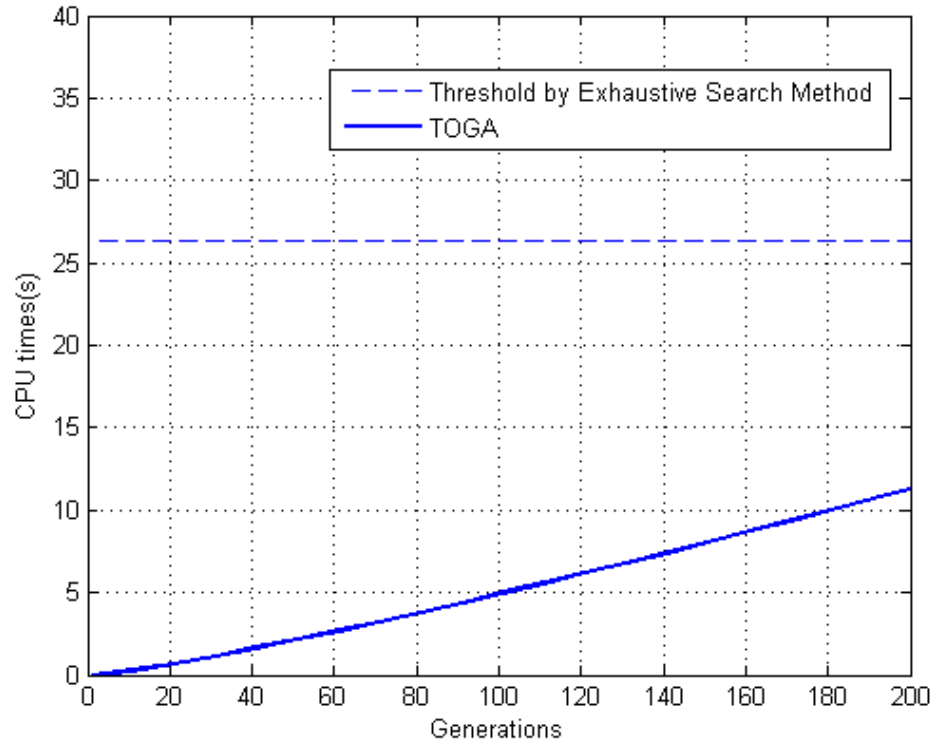


Figure 6.11: CPU time with different generations

According to Figure 6.11, we compare the execution time of TOGA with that of the exhaustive search method. Each CPU time in this paper is the average of 20 independent executions. We can observe that our TOGA is much faster than the exhaustive search method.

In summary, to address the composite service selection problem which is NP-hard, our proposed TOGA could obtain a near-optimal concrete service composition plan after a certain number of generations. As the average CPU time for a single generation in TOGA is as short as 0.04 second. After 80 generations, the execution time of our TOGA is only about 13.67% of the exhaustive search method. Meanwhile, the average accuracy and optimality are high which means the overall effectiveness of TOGA is good. In addition, TOGA is suitable for parallel computing as the traditional genetic algorithm. This can greatly speed up computations and shorten the overall CPU time. Therefore, our proposed TOGA is realistic and efficient for composite service selection.

## 6.6 Summary

The trust-oriented QoS-aware service selection is an attractive research area. In this chapter, we have provided a formal service invocation architecture for service composition. The abstract service composition graph and the concrete service composition graph have been defined. In addition, we have proposed the trust evaluation method for the concrete service composition plan based on the subjective probability theory. Focusing on the QoS aware composite service selection problem which is NP-hard, our Trust-Oriented Genetic Algorithm (TOGA) has been proposed to find a near-optimal composite service plan with QoS constraints. Experimental results have illustrated that our proposed approach can discover the near-optimal solution effectively and efficiently.

# Chapter 7

---

## Business Rule aware Composite Service Selection

In Chapter 6, the Trust-Oriented Genetic Algorithm (TOGA) was proposed to solve the QoS based composite service problem which is NP-hard. In this chapter, we will further investigate the issues in the service composition, such as dependencies and conflicts caused by the business relationships among the concrete service providers. TOGA will be extended to support business rule aware composite service selection. This chapter is organized into sections as follow. In Section 7.1, we introduce the research challenges that will be addressed. Section 7.2 presents a motivating use case which will be studied throughout this chapter. The system model is identified in Section 7.3. In Section 7.4, the business to business rules are illustrated. The extended TOGA selection method is described in Section 7.5. After that, the case study and experiments are discussed in Section 7.6. Section 7.7 is the summary of this chapter.

### 7.1 Introduction

In the SOA environment, the individual applications can be published, discovered and invoked across the web. The complex service based applications can be applied to deliver greater values by composite services rather than individual services. This emerging application architecture has increased the demand for web services. As we mentioned in the Chapter 6, the single services from different providers can be used, composed, and coordinated in a loosely coupled manner. In addition, QoS is often critical when the service consumer acquires a service from a large pool of services offered by different providers. The QoS of a composite service is aggregated in a forthright manner because the concrete services in the composition are seen as atomic and independent. As the business relationships between service providers

become more and more complicated, the dependencies and conflicts may exist, thus affecting some QoS dimensions, cost for instance, of the service composition.

In this chapter, we are focusing on analyzing the business relationships between the concrete service providers in the concrete service discovering stage, and detecting the influences on QoS. According to concrete service selection and binding, we argue that the traditional work-flow based approaches for web service composition exhibit an important shortcoming, as they hardly concern about the integration of business to business rules. Depending on different kinds of business rules, the service providers may not keep their QoS values consistently. For example, the hotel service could offer a special price to the customers who are served by the affiliated travel agent. Therefore, QoS influences which are affected by business rules need to be investigated, especially for the composite service optimization.

Several QoS driven composite service optimization approaches [ASTL07, HMRR08, Men04, XB05, YZL07, ZBD<sup>+</sup>03] have been proposed because finding a solution for quality concerned services selection is a NP-hard problem [CPEV05b]. However, they ignored the business relationships, which would affect some QoS dimensions significantly, among the concrete services, such as dependencies and conflicts. In this chapter, we extend our TOGA, which was introduced in Chapter 6, to find a near-optimal concrete service composition plan with the consideration of business rules. The business rules are represented in the Business Relationship Matrix (BRM) to support the GA for the composite service selection.

## 7.2 A Motivating Case

In order to understand the needs to consider the requirements of business rules in a service composition, let us consider the following scenario.

In this scenario, the Roadside Assistant (RA) service center in Figure 7.1 is responsible for assisting people when their cars are broken down. This comprehensive RA service center is a kind of composite service built upon some single services, such as the car insurance service, the car towing service, the car repairing service, the car rental service and the accommodation service. Based on the customer agreement and the location of the breakdown point, there might be more than one individual service provider who is able to accomplish each sub-task. In order to satisfy the QoS requirements of the customer, for example, the cost and the serving time, an optimized composition plan needs to be provided by RA service center. Therefore,

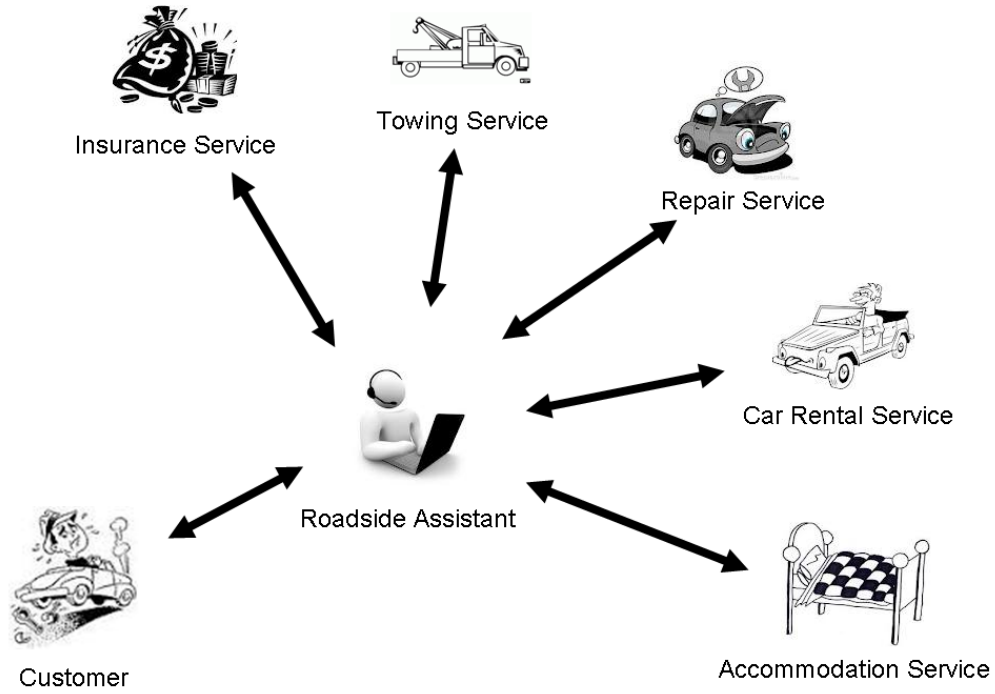


Figure 7.1: The Roadside Assistant Service

the RA service center is responsible for selecting and composing such individual services in an optimal way. Very likely, some business rules might exist among those individual service providers. For example, a car insurance company might use only approved car repairing services. Thus, there is a need for the RA service center to investigate and handle the business rules among the individual service candidates. Some sample business rules among the individual service providers are illustrated as follows:

- Rule with dependency: the customer can only choose the car repairing service  $s(p_1)$  or  $s(p_2)$  when he has the car insurance from the insurance company  $s(i_1)$ .
- Rule with positive inference: the accommodation service  $s(a_1)$  provides 20% discount if the customer rents the car from the car rental service  $s(r_1)$ .
- Rule with conflict: the towing service  $s(t_1)$  refuses to tow the car to any car repairing services located out of City A.

Based on these business rules, there are some dependencies, conflicts, and cost

influences among the service candidates for the service composition. The traditional approaches, which assume concrete services are atomic and independent, are not able to identify and cater for these relationships. As a consequence, the resultant composition may fail to deliver the required services as expected. Therefore, the hybrid web service composition which takes the business rules into account, is needed.

### 7.3 System Model

In this chapter, we extend our system architecture by the consideration of business rules. The business rules handler is developed to support the business rule driven web service composition model as shown in Figure 7.2.

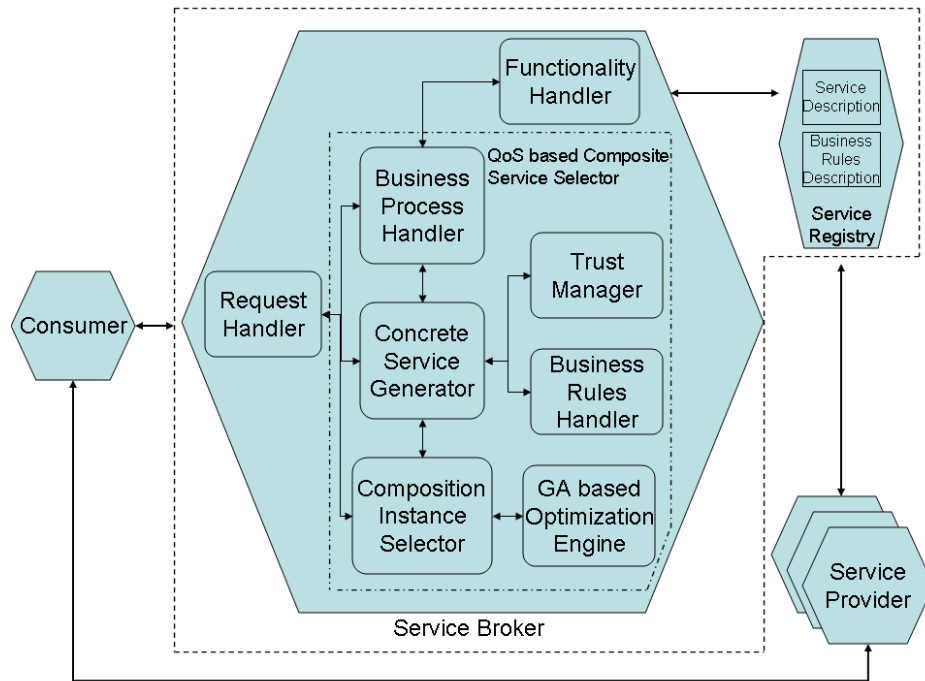


Figure 7.2: Business Rule Driven Service Composition Model

In the stage of creating the process model, the web services are composed by the business process handler which is supported by the process-oriented composition languages such as BPEL and BPML. The components of a process are described as abstract services or tasks. For the abstract service composition, the process model defines the invocation structures which are described in Chapter 6, such as

Sequential invocation, AND invocation, OR invocation, AND join invocation, OR join invocation, and Loop invocation. In order to achieve the task requirements for the composition, the concrete services are classified in functional categories based on the service functional descriptions. The process of discovering abstract services is out of the scope of this work. We assume the abstract service composition is provided by the business process handler, and we are focusing on the concrete service selection and binding for the actual service composition which will be done based on the functional description.

In the stage of discovering the concrete services as we mentioned in Chapter 6, the service instances are selected from the service repository to accomplish the particular tasks which are generated from the abstract service composition. At the same time, the business rules, which are published by these service candidates, are collected and analyzed by the business rules handler. Once the business rules are expressed explicitly and are allied with the service description in the service repository, they can be reused across several composition. In other words, the same rule can be applied to many web service compositions. As a result, the dependencies, conflicts or the QoS influences among the individual service providers are presented in the BRM. In order to find the optimal composition instance with considering the business rules, our proposed TOGA is extended by evaluating business relationships in BRM.

## 7.4 Business Rules

The business rules for web service composition distinguish two phases: the description phase and the analysis phase. The web service providers publish their business rules, which are allied with the service description, when they register to the service repository. Once the process of a service composition is determined, the business rules are collected from the rules repository, which is a logical part of the service repository. These collected rules are analyzed as a matrix based model called BRM, and are used by the Rules Analyzer.

### 7.4.1 Business Rules Description

Business Rules are a significant part of the business. They could represent business policies for the internal or external regulations. A business rule is “a proposition



```

<set of X> ::= <X>|<X><set of X>

<rule-setting> ::= <rule-issuer><set of rule-stmt>
<rule-issuer> ::= <task>". "<name>
<rule-stmt> ::= <task>". "<set of condition>"→"<set of effect>

<condition> ::= <unit>|<threshold>
<unit> ::= <name>
<threshold> ::= <threshold-type>":"<attribute>="(<var>|<constant>)"
<threshold-type> ::= ("top"|"bot"|"equivalent")

<effect> ::= <effect-type>("(<attribute>="(<var>|<constant>))")
<effect-type> ::= ("dependent"|"positive"|"negative"|"conflicting")

```

Table 7.1: The syntax of the business rule description language

about business things, relationships between them and operations applied to them, from the business enterprise viewpoint” [ANS96]. In our research, we focus on the relationships between various service providers rather than the internal regulations. The business rules discussed in this paper exhibit the nature of B2B rules.

In addition, the business rules have been classified by the Business Rules Group into three categories: structural assertions, action assertions and derivations[HH00]. The business structures are considered in the structural assertions, and the action assertions concern rules of activities, behaviors, and process flows. Derivations concern rules on effects and calculation methods. The business rules in our research are used to represent the dependencies and the QoS inferences between the service providers; thus, they could be classified into the derivations.

The business rules are normally presented in a natural language, but the expression of logical language is needed in order to increase the applicability. The business rules are published in the 1:n basis for a service provider, and the rule set consists of two main parts: rule-issuer and rule-statement. The syntax of the business rule description language is represented in BNF (*Backus Naur Form*) as in Table 7.1.

For flexibility and reusability, the business rules reflect a service provider’s relationships and the inferences unidirectionally to another service provider rather than the bidirectional inferences. The business rules are normally described by a natural language in the form of Event-Condition-Action [BW93, Her95]. The alternative expression can be also in the form of ⟨Subject⟩ Must ⟨Constraints⟩, or in the form

of If-Then [THHD11]. According to the business rules driven web service composition model described in Section 7.3, the business rules need to be able to express the abstract services which are already generated, and concern the QoS inferences. Therefore, the rule-statement is described by taking the following form as:

$$\langle \mathbf{Task} \rangle . \langle \mathbf{Condition} \rangle \rightarrow \langle \mathbf{Effect} \rangle$$

The three basic elements of the rule-statement: Task, Conditions, and Effects are discussed, and their properties are described as follows:

- **Task:** it represents the abstract service, as well as the business function, to which the rules are related. The functionality of the service is well described by OWL-S and WSDL-S which are the standards for publishing services semantically.
- **Conditions:** they describe the constraints of the concrete service that issues the rule. A condition could present two types of constraints: *unit* or *threshold*. The type of *unit* constrains a number of specified concrete services by names; while *threshold* constrains a group of concrete services by the constraints of the attributes.

For the *unit* condition, the concrete services, to which the rules are related, are strictly specified. For example, for the *threshold* type, the constraints could be defined by three properties: *top*, *bot*, or *equivalent*. The property *top* defines the top element of the dimension as the upper bound, while the property *bot* defines the bottom element of the dimension as the lower bound. The property *equivalent* indicates the certain value of the dimension. The value of the attribute dimension could be either numerical or categorical. Comparing to the numerical value, the categorical value is more complicated to detect the boundaries. As presented in Chapter 4, the hierarchy relationships (either consecutive hierarchy or inconsecutive hierarchy) could be provided by the domain experts. The consecutive hierarchy relies on the number of members in the value set, while the inconsecutive hierarchy relies on the provided semantic ontology. On one hand, for the consecutive hierarchy, the rules can affect the service providers whose attributes are subset/superset or equivalent to the value of the bound depends on the different constraint properties. On the other hand, for the inconsecutive hierarchy, the rules can affect the service

providers whose attributes are the parent/child node which is based on the ontology.

- **Effects:** they concern the logical expression of the relationships between the antecedent service(s) and the rule issuer, as well as the inferences on the QoS dimensions of the issuer. The effects are classified into four main types: dependent, positive, negative, and conflicting which are defined as:
  1. *dependent* indicates the rule-issuer is valid when the antecedent services are composed to accomplish the specified task in the composition.
  2. *positive* indicates the rule-issuer is running in a positive manner on the appointed QoS dimensions when the antecedent services are composed to accomplish the specified task in the composition.
  3. *negative* indicates the rule-issuer is running in a negative manner on the appointed QoS dimensions when the antecedent services are composed to accomplish the specified task in the composition.
  4. *conflicting* indicates the rule-issuer is valid when the antecedent services are not composed to accomplish the specified task in the composition.

They are used to represent the business relationships in BRM which will be discussed in the following subsection.

The business rules express the effects on some QoS values, which are concerned by the approach of the composite service selection. Therefore, the business rules, especially the types of the business relationships between the concrete services, should be analyzed before the selection.

#### 7.4.2 Business Relationship Matrix

Once the business rules are presented, they could be collected by the Rules Analyzer in order to analyze the service composition inference. The business relationships among the concrete service candidates for the determined abstract service composition can be represented as a graph or a matrix based model. In this approach, the matrix is used to represent the types of the effects that are discussed above. The matrix that models the dependencies and conflicts will be a square matrix ( $n \times n$ ) where  $n$  is the number of the available web service candidates to form the service

composition. Each row and column represent the concrete services for the composite service. If a service in the  $i^{th}$  column is affected by a service in the  $j^{th}$  row, then the  $R_{ij}$  value of the matrix will be presented as  $d$ ,  $p$ ,  $n$ , or  $c$ , which indicates the type of effect *dependent*, *positive*, *negative*, and *conflicting*, respectively.

Let the composite service be created by  $n$  concrete service candidates. The BRM can be defined as follows:

$$\text{BRM} = \begin{pmatrix} R_{11} & R_{12} & R_{13} & \dots & R_{1n} \\ R_{21} & R_{22} & R_{23} & \dots & R_{2n} \\ R_{31} & R_{32} & R_{33} & \dots & R_{3n} \\ \vdots & \dots & \dots & \dots & \vdots \\ R_{n1} & R_{n2} & R_{n3} & \dots & R_{nn} \end{pmatrix}; \text{where } R_{ij} = \begin{cases} d \\ p \\ n \\ c \end{cases}$$

In the selection phase, the different business relationships could be collected from the BRM for the information retrieval. By the genetic algorithm, a possible service composition plan is presented as a chromosome, and the better solution is generated by the crossover and mutation functions. However, the business related dependencies and conflicts might exist when the crossover and mutation are operating, which will generate unfeasible composition plans. Therefore, the business relationships need to be checked in the BRM to adjust the utilities of the composition plans. The selection schema is carried out, which depends on the four types of effect, by extending our TOGA.

In this thesis, BRM represents the initial business relationship between two single services. If the service consumers have no knowledge or previous experience on composite services, any transitive relationship among multiple services would be based on the initial business rules. For example, if  $S_1$  and  $S_2$  is positive relationship, and  $S_2$  and  $S_3$  is positive relationship as well, the service composition which  $S_1$ ,  $S_2$  and  $S_3$  are composed together would be positive, and the final effect is based on two initial effects which are presented in the initial business relationships. In addition, there is another research dimension which focuses on the relationship among more than two single services. Two or more services would be seen as a service group, for example, if  $S_1$ ,  $S_2$ , and  $S_3$  are dependent,  $S_1$  and  $S_2$  can be seen as a service group which has dependent relationship with  $S_3$ . In this chapter, we focus on the influences by the business relationships for the composite service selection. The transitive relationships and the service grouping are out of the research scope because they are based on the initial single service relationships which are presented in BRM.

## 7.5 Business Rule Based Composite Service Selection

With a number of concrete service candidates, composite service selection with QoS constraints can be modeled as the Multi-dimension Multi-choice Knapsack Problem which is an NP-Complete problem. As we mentioned in Chapter 6, the genetic algorithm is one of the best techniques for processing NP-Complete problems. In this chapter, we extend our TOGA by considering the business relationships in the fitness function.

Every single concrete service composition plan has a fitness value which indicates the utility of that plan. The fitness value determines whether the composition plan will be kept in the population or not for the genetic algorithm. The goal of the composite service selection is to select possible concrete service composition plans which meet user's QoS constraints and also provide the best value for the user-centric fitness function  $\mathcal{F}_{fitness}$ .  $\mathcal{F}_{fitness}$  is defined by a weighted sum of the aggregated QoS satisfaction value. Suppose a user has  $m$  QoS constraints in a composite service request as  $Q_r = [Q_r^1, Q_r^2, \dots, Q_r^m]$ . For each dimension of QoS request,  $Q_r^i$  is presented as  $Q_r^i = \langle Name, Negotiability, Willingness, Value \rangle$  in the request model which is described in Chapter 4. In this chapter, the fitness function is extended by considering the business relationships from the one in Chapter 6. The business relationship aware fitness function is defined as follows:

$$\mathcal{F}_{fitness} = \begin{cases} \lambda_c \cdot T(g) \sum_{i=1}^m w_i f_i(Q_r^i, q_g^i), & \text{if } \prod_{i=1}^m f_i(Q_r^i, q_g^i) = 0 \\ \lambda_r \cdot T(g) \sum_{i=1}^m w_i f_i(Q_r^i, q_g^i), & \text{if } c \in BRM \vee \exists d \in BRM \text{ with others} \\ T(g) \sum_{i=1}^m w_i f_i(Q_r^i, q_g^i), & \text{otherwise} \end{cases} \quad (7.1)$$

where  $\lambda_r$  and  $\lambda_c$  define the penalty factors in (0,1) which are applied for any unsatisfied situations in order to decrease the fitness value. In our genetic algorithm, the higher fitness value allied with the composition plan, the optimizer for selection. In addition, to increase the robustness of the algorithm, every chromosome needs to be assigned a fitness value rather than a simple zero, which means the composition with penalty still has a chance to generate a better solution. In this chapter, the unsatisfied conditions are specified into two different dimensions: 1) the composition

plan does not follow the business rules, and 2) the composition plan does not meet the QoS constraints. Before the fitness value is computed, the business relationships need to be checked from BRM. For a concrete service composition plan, each row of the concrete service in BRM needs to be enumerated to collect the business relationships within the composition.  $\lambda_r$  will be applied when the conflicting rules ( $c$  in the BRM) are found or the dependent services ( $d$  in the BRM) are not involved in the composition. For the relationships of *positive* and *negative*, the *Effects* of the rules will be collected to adjust the QoS value. For example in the RA case, if the insurance service  $s(i_1)$  is composed with neither the car repairing service  $s(p_1)$  nor  $s(p_2)$ , or the car towing service  $s(t_1)$  and the car repairing service  $s(p_3)$  are composed together, the  $\lambda_r$  will be applied for computing the fitness value. In addition, the cost of the accommodation service  $s(a_1)$  needs to decrease by 20% if the car rental service  $s(r_1)$  is composed together. The details of QoS aggregation method is well presented in [JRG05], and we focus on the influences of the QoS changes of the concrete services in the composition. As we mentioned in Chapter 6,  $\lambda_c$  is the penalty factor of QoS constraint which will be applied when the QoS value of the composition does not satisfy the customer's QoS requirement. Because  $f_i$  represents the satisfaction value on  $i^{th}$  QoS dimension,  $\lambda_c$  will be activated when  $\prod_{i=1}^m f_i(Q_r^i, q_g^i) = 0$ .

$T(g)$  denotes the global trust value of a concrete service composition plan.

$w_i$  is the weight for importance of each QoS dimension, where  $\sum_{i=1}^m w_i = 1$ .

$q_g^i$  is the global QoS aggregation value of the  $i^{th}$  dimension. In our approach, the aggregated value is affected by different kinds of business rules because the independent QoS value of the concrete service might be changed by the *Effects* that are described in business rules.

$f_i$  is the satisfaction function on the  $i^{th}$  dimension which depends on the *Willingness* of QoS requirement. The equations of  $f_i$  were described in Section 6.2.4.

The fitness function mentioned above is used to select some better plans in the current generation. As we mentioned in Chapter 6, crossover operator is used to vary chromosomes from one generation to the next, and the parent chromosome is selected with a probability based on the fitness value. Mutation operation is used to maintain genetic diversity from one generation of chromosomes to the next. An example of the crossover and mutation operations is shown in Chapter 6. In the crossover operation, all the concrete services between the two points are swapped between the parent chromosomes to render two child chromosomes. In the mutation operation, the mutated concrete service is selected by "roulette wheel" method based

on the trust value, to replace the current concrete service.

The evolution will continue until the max number of generations is searched or will stop when the best composition plan in the current generation remains unchanged for a given number of generations. The resulting concrete service composition plan is the near optimal composition plan for the service composition.

## 7.6 Case Study and Experiment

To better illustrate our key ideas, we further investigate the RA case discussed in Section 7.2. The customers call for help from the roadside assistant center when their cars are broken down on the road. The car towing service, the car repairing service, the car rental service, and the accommodation service are composed as a package if needed when the customers provide their car insurance companies. When the number of service providers that are able to accomplish each task increases, there might be hundreds even thousands of composition plans. For simplicity, we assume three different service providers are available for each task. Therefore, 243 possible composition plans can be provided by the roadside assistant service center. We also take three kinds of QoS attributes into account, i.e., cost, reliability, and distance to the breakdown point. Meanwhile, the customers are willing to minimize the cost and distance and maximize the reliability of the composition plan. The details of QoS values are presented in Table 7.2.

According to the three types of business rules illustrated in Section 7.2, we introduce three business rules in this case: (i) the customers can only choose the car repair services Frankies Auto ( $s(p_1)$ ) and AVA Auto Repairs ( $s(p_2)$ ) when the car insurance is covered by NRMA ( $s(i_1)$ ); (ii) the accommodation service Novotel ( $s(a_1)$ ) provides 20% off if the customer rents the car from the car rental service Thrifty ( $s(r_1)$ ); and (iii) the towing service Citywide ( $s(t_1)$ ) refuses to tow the car to any car repairing services located out of Sydney. In this case, the Holden Service Station ( $s(p_3)$ ) is located out of Sydney. The representation of these three business rules could be described in Table 7.3. The functionalities of the concrete services are acquired from the web service descriptions. According to these three business rules in the case, there are 54 composition plans with dependent relationship, 27 composition plans with positive relationship, and 27 composition plans with conflicting relationship. In addition, if more business rules are published, the proportion of the composition plans with different business relationships will increase. Therefore,

Towing service	Citywide	NRMA	AHR
Cost(\$)	29.99	49.99	39.99
Reliability	82	95	85
Distance(km)	8.7	13.4	15.3
Repairing service	Frankies Auto	AVA Auto Repairs	Holden Service Station
Cost(\$)	126	89	95
Reliability	92	86	91
Distance(km)	8.7	5.4	17.6
Rental service	Thrifty	Budget	AVIS
Cost(\$)	60.90	49.95	39.95
Reliability	96	93	88
Distance(km)	15.6	12.8	8.9
Accommodation service	Novotel	Chifley Hotel	Belmore Hotel
Cost(\$)	85	109	99
Reliability	98	92	94
Distance(km)	5.8	6.4	9.4

Table 7.2: QoS information of each concrete service in the RA case

without considering the business rules among the concrete service providers, the composition plan might be infeasible because of the dependent and conflicting rules; or inaccurate because of the positive and negative rules.

Once the business rules are published by the service providers, the business relationships will be established in BRM by the business rules handler. The BRM in this case is presented in Table 7.4. For each concrete service composition plan, the business relationships among the concrete service providers could be searched from the BRM to support the selection algorithm.

Before computing the fitness value of every concrete service composition plan,

rule-issuer: CarInsurance.NRMA rule-stmt: CarRepairing.Frankies Auto AVA Auto→dependent
rule-issuer: Accommodation.Novotel rule-stmt: CarRenting.Thrifty→positive(cost="20% discount")
rule-issuer: CarTowing.Citywide rule-stmt: CarRepairing.any::location≠"CityA"→conflicting

Table 7.3: Examples for Business Rules Description



	Frankies Auto	AVA Auto	Holden Service Station	Novotel
NRMA	d	d	NA	NA
Citywide	NA	NA	c	NA
Thrifty	NA	NA	NA	p

Table 7.4: An Example of BRM

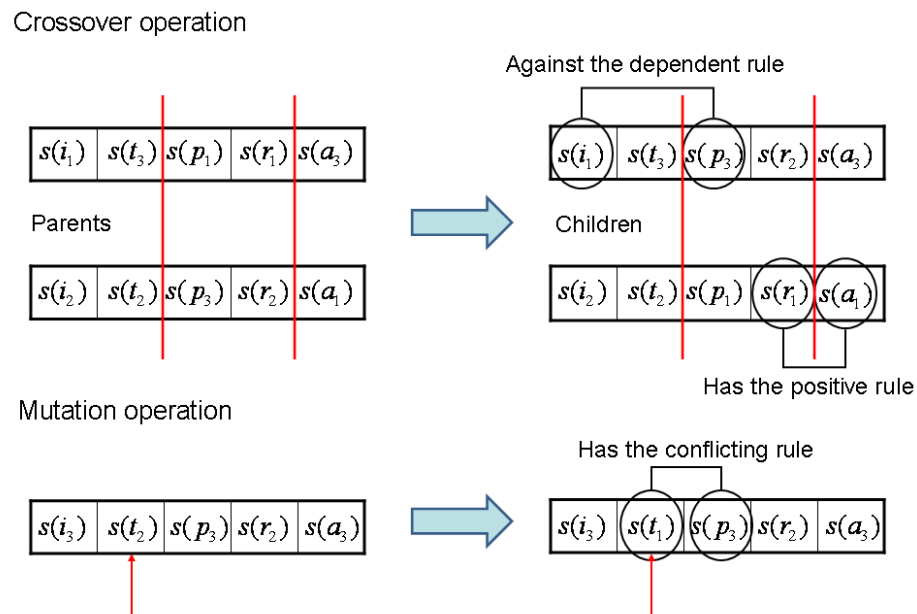


Figure 7.3: Crossover and mutation operations for RA case

the business relationships need to be checked in BRM. An example of the crossover and mutation operation for RA case is presented in Figure 7.3. In the crossover operation, two children chromosomes are generated from the parents by two-points crossover function. When the business relationships are checked in BRM, the infeasible composition plan, a child chromosome where  $s(i_1)$  and  $s(p_3)$  are composed together, is detected because it is running against the dependent rule of the insurance service  $s(i_1)$ . In this case, without considering the business rules, the fitness value of this child chromosome is 0.681, which means it is more likely to survive to the next generation in GA. While checking the business rules in BRM, the penalty factor  $\lambda_r = 0.01$  must be applied for computing the fitness value to 0.07. In consequence, this infeasible composition plan would be eliminated in this generation. In addition, the positive rule of the accommodation service  $s(a_1)$  is detected in BRM for the other child chromosome where  $s(r_1)$  and  $s(a_1)$  are composed together. To compute the fitness value of this child chromosome, the cost of  $s(a_1)$  needs to be adjusted to 80% of the original one for the aggregate cost. Thus, with consideration of the business rules, the fitness value will be adjusted from 0.729 to 0.735, which means this child chromosome has higher possibility to be selected in GA. In the mutation operation, there is a probability to mutate the car towing service from  $s(t_2)$  to  $s(t_1)$ . When it happens, the conflicting rule is detected in BRM between the car towing service  $s(t_1)$  and the car repairing service  $s(p_3)$  because the location of  $s(p_3)$  is out of Sydney. Therefore, the penalty factor  $\lambda_r$  must be applied, and the fitness value of the mutated chromosome is adjusted from 0.675 to 0.007, which means this mutated chromosome has lower possibility to survive in the generation.

In the experiment, we increase the number of concrete service from 3 to 15 for each abstract service. Thus, there are  $15^5$  concrete service composition plans. All experiments were taken on the same software and hardware as the experiment settings in Chapter 6. Matlab 7.0.0.19920(R14) was running on a Dell Optiplex 745 with an Intel Core 2 Duo 6400 2.13GHz CPU and a 2GM RAM, Windows XP Pro. The concrete services for each task are created with three QoS properties as in the case study as cost, reputation, and distance. 20 business rules, including dependent, positive, negative, and conflicting, are randomly assigned among the concrete services.

Firstly, the genetic algorithm is tested without considering business rules to find the near-optimal solution. Figure 7.4 represents the result of a single running of genetic algorithm with 100 generations. It shows that the fitness value of the best

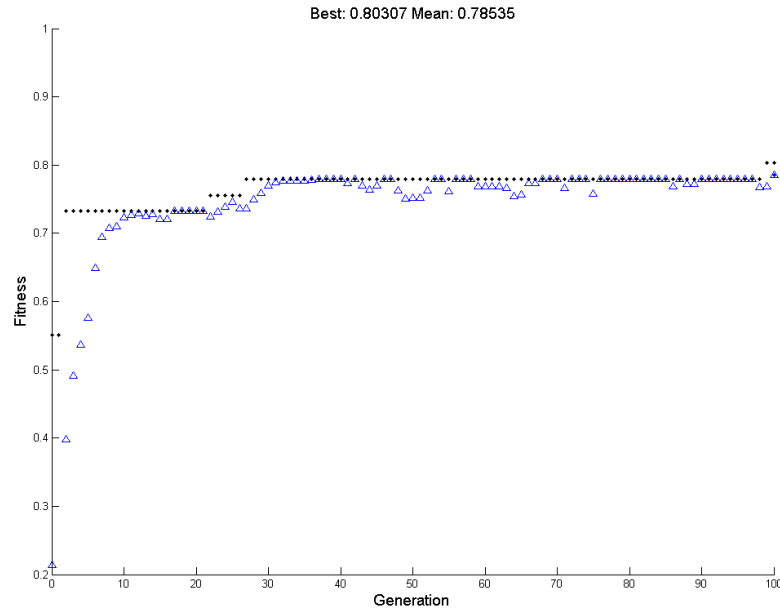


Figure 7.4: Genetic algorithm running without the consideration of the business rules

concrete service composition plan is 0.80307 and the average fitness value in the final population is 0.78535. In addition, the best composition plan is composed by  $s(i_8)$ ,  $s(t_2)$ ,  $s(p_{14})$ ,  $s(r_7)$ , and  $s(a_3)$ . Unfortunately, when we check the business relationships in BRM, this concrete composition plan will not be a feasible solution because there is a conflicting business rule between  $s(i_8)$  and  $s(p_{14})$  as described as follow:

```
rule-issuer:
CarInsurance.s(i8)
rule-stmt:
CarRepairing.s(p3) | s(p7) | s(p14) →conflicting
```

As we mentioned in Section 7.5, the penalty factor  $\lambda_r$ , as 0.01, must be applied to adjust the fitness value to 0.00803. Therefore, the genetic algorithm without considering business rules might find an infeasible concrete service composition plan.

Secondly, the genetic algorithm is tested with the consideration of the business rules. Figure 7.5 represents the result of a single running. The fitness value of the best concrete service composition plan is 0.70859 and the average fitness value of

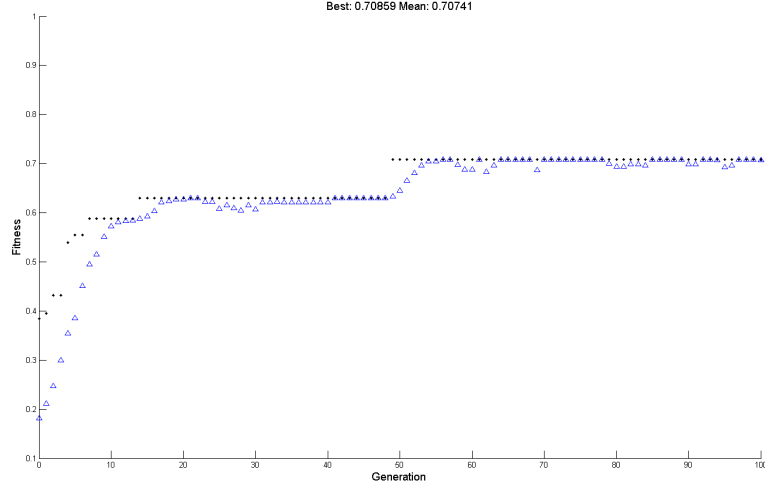


Figure 7.5: Genetic algorithm running with the consideration of the business rules the final population is 0.70741. The best solution is composed by  $s(i_3)$ ,  $s(t_4)$ ,  $s(p_{14})$ ,  $s(r_7)$ , and  $s(a_9)$ . Furthermore, this solution is the same as the optimal solution which is computed by the exhaustive search method.

To evaluate the feasibility of genetic algorithms, Figure 7.6 represents the optimality of two GAs based on the exhaustive search. The computation method of  $O(f_i)$  is presented in Section 6.5. The number of generation are set as 20, 40, 80, 100, 150, and 200, respectively. The simulation times are 20 for every round of the experiment based on the different initialized populations. The average  $O(f_i)$  values for two types of TOGA are recorded in Table 7.5.

Generation	20	40	60	80	100	150	200
Without BRM	76.89%	92.99%	98.94%	105.59%	110.25%	113.51%	114.14%
With BRM	76.25%	87.65%	89.98%	90.20%	91.36%	94.95%	96.95%

Table 7.5: Experiment records

In Figure 7.6,  $O(f) = 1$  means the best composition plan which is computed by the exhaustive search method. The optimality of GA without the consideration of the business rules is greater than 1, which means the fitness value of the best solution by GA is greater than the real best one. Therefore, a number of solutions in the population are infeasible because they might against the business rules. With the consideration of the business rules, the fitness value may be decreased by dependent, conflicting, or negative business rules. We can observe that the average  $O(f_i)$  is

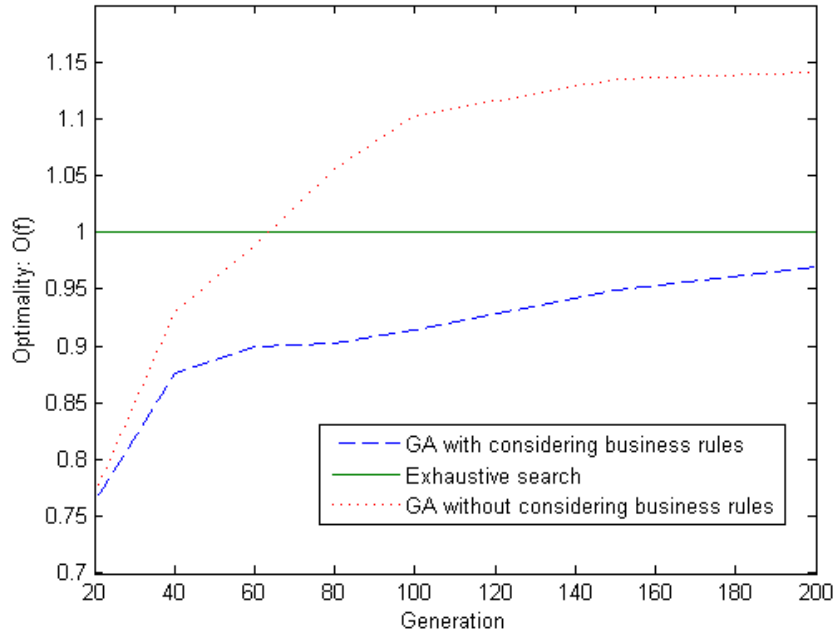


Figure 7.6: Optimality of GAs

96.95% when there are 200 generations. That means the fitness value of the near-optimal solution with the consideration of the business rules is very closed to the real best one.

As a result, BRM could help us detect the different business relationships within a service composition. In addition to filtering the dependencies and conflicts, the aggregated QoS value could be adjusted based on *Effects* in the rules description effectively.

## 7.7 Summary

The business rules aware composite service selection is an attractive research area. In this chapter, we illustrated the effects of business rules among concrete service providers in the composition, such as dependency, conflict, and positive/negative inference on the QoS dimension. The formal business rule description language was defined in BNF to support the implementation. Once the business rules were captured, the Business Relationship Matrix will be generated for the selection phase. From the case study and the experiment, the extended TOGA with the consideration

---

of the business rules can filter the infeasible composition plans, and it is realistic for composite service selection.

# Chapter 8

---

## Conclusion and Future Works

During the last few years, Service-Oriented Computing (SOC) has emerged to be a significantly important research area that has attracted increasing attention from both the research and industry communities. In the SOC environment, the service-oriented applications are developed in a loosely-coupled way, and the consumers could discover and invoke their preferred services from the rich service pool. With the ever-increasing number of services published on the Internet, how to discover a desired service effectively and efficiently becomes a challenging issue. Specifically, selecting web service with high quality while satisfying consumer's requirements within a large pool of functionally equivalent service providers is the key factor in service selection.

In addition to the functional property, the service selection is commonly driven by the non-functional property. Therefore, it is critical to evaluate the QoS accurately and objectively. Considering the typical web service discovery scenario, a comprehensive selection mechanism is needed to help the consumer to choose the “right” and “best” services automatically or semi-automatically.

After recognizing the challenges in the QoS based service selection process, this thesis has deeply investigated the problems in both single and composite service selection. In this chapter, the major contribution of this thesis is summarized, limitations of this research are discussed, and future research directions are outlined.

### 8.1 Summary of Major Contribution

In this thesis, we have made several noteworthy contribution to the QoS based service selection area. The major contribution of this thesis can be described in two main research dimensions: QoS based single service selection and the QoS based

composite service selection. For single service selection, we have proposed a similarity based evaluation method to detect the difference between the consumer's requirements and the published QoS information. In addition, we have introduced an attribute based service customization approach that allows the service providers to publish the customizable QoS information for different user groups. For composite service selection, we have proposed a novel Trust-Oriented Genetic Algorithm (TOGA) to find a near-optimal concrete service composition plan with the QoS constraints. Furthermore, the dependencies and conflicts which are caused by the business to business relationships have been analyzed within the composite services. The detailed contribution to QoS based service selection is enumerated and discussed as follows.

#### 1. Contribution to QoS based single service selection

- Chapter 4 studied the similarity based service selection approach by evaluating the similarity score between a QoS request and published QoS information. We found that the traditional similarity value that reflects the absolute distance between a request and published QoS information cannot depict the similarity very well under certain circumstances. In order to observe the tolerance on some QoS dimensions that a service could provide, a robust similarity assessment has been proposed by measuring the relative distance. For calculating this relative distance, we have proposed a comprehensive service request model to express the consumer's request more accurately. In this request model, the service consumers could present the tendencies of the preferences by the attributes of *Willingness* which are classified as Bot, Top, Equivalent, Close, and Distant. In addition, the QoS value on each dimension for both the service request and the published QoS information could be further described by the attribute of *DataType*, such as numerical or categorical, single or range, domain-dependent or domain-independent, and negotiable or non-negotiable. These values made the request and the publication more flexible. For the categorical values which were rarely mentioned in other's works, we have proposed a novel measurement to calculate the similarity between them. The main idea was based on the set theory and semantic ontology. Specifically, the consecutive hierarchy relied on the number of members in the value set, in which the distance could be measured based



on the dominate relations, and the inconsecutive hierarchy relied on the provided semantic ontology.

- Chapter 5 introduced an attribute based service customization and selection model to address the need of acquiring the QoS information based on consumer's attributes, and selecting the demanded services more accurately. The extended attribute based access control policy language has been proposed to publish the additional effects beyond the accessibility. Specifically, the customized QoS values were identified by CRules which were presented by the service providers as the linkages between the subsets of the subject and the QoS sets. Different QoS values are provided to different user groups which are distinguished by their attributes. By the extended attribute based access control policy, the service providers expose their QoS information to the consumers who hold the accessibility. In addition, three different policy evaluation strategies, such as First-In-Applicable, Pre-Order-Applicable, and More-Attribute-Applicable, were illustrated. The similarity based selection method is also applied to compute the similarity between the consumer's QoS requirements and the customized QoS values. To better understand this approach, a case of movie download service was demonstrated by the system prototype.

## 2. Contribution on the QoS based composite service selection

- Chapter 6 studied the trust-oriented QoS based composite. We have provided a formal service invocation architecture for the service composition. The abstract service composition graph and the concrete service composition graph have been defined. In addition, we have proposed the trust evaluation method for the concrete service composition plan based on the subjective probability theory. Focusing on the QoS based composite service selection problem which is NP-hard, the novel Trust-Oriented Genetic Algorithm (TOGA) has been proposed to find a near-optimal concrete service composition plan with QoS constraints. The experiments have been conducted on a composite service that includes a large number of concrete service composition plans, to compare the proposed TOGA with the exhaustive search method. The experimental results have illustrated that our proposed approach can discover the near-optimal solution effectively and efficiently.

- Chapter 7 studied the business rules driven composite service selection. We have investigated the different business relationships, which would affect some QoS dimensions significantly, among the concrete services. Four basic effect types of business rules have been demonstrated for the composite service, such as dependency, conflict, positive inference, and negative inference on the particular QoS dimension. In addition, a formal business rule description language that was defined in BNF has been introduced to support the implementation of the service composition. Once the business rules were captured, the Business Relationship Matrix (BRM) will be generated for the selection phase. Furthermore, we adopted the GA to find the near-optimal solution. To investigate the feasibility of our approach, the case of Roadside Assistant has been studied. The experimental result showed that the proposed BRM could help us detect the different business relationships within a service composition. The aggregated QoS value could be adjusted based on the business rules effectively.

According to the major contribution of this research, a comprehensive approach to QoS based single and composite service selection could be practically established. With this approach, it can automatically help consumers to select both the single and the composite services in their demands accurately.

## 8.2 Limitations

This research focused on some challenges for QoS based service selection. Although our service selection approaches are novel, effective, and efficient, there are some limitations in our approach.

- First of all, we assumed that the service providers were classified by their functions, and we focused on the non-functional properties only. However, the functional clustering of the services is another issue in the service publishing and discovery. There is no standard for defining the functions of the services in a unified form, although the web service description language is already widely used. Furthermore, in the semantic web, the functional classifications which are called ontology can not be standardized neither.

- Secondly, the domain ontology is needed especially for the categorical QoS values. For our similarity based service selection approach, the similarity score can be detected based on the domain ontology which needs to be presented by the service providers or the domain experts. Unfortunately, in some specific domains, the ontology might not be provided. Although there are some description languages such as WSMO, OWL-S, and WSDL-S, the proper domain specific categorical ontology are needed.
- Thirdly, the security and encryption issues on Attribute Based Access Control (ABAC) were not further discussed in this thesis. ABAC is a hot topic in the encryption research area. Unfortunately, the attribute based authorization and authentication mechanisms are not widely used for industries currently due to its complexity.
- Fourthly, our trust-oriented composite service selection approach was proposed with the assumption of the centralized trust authority. There are many well-known trust authorities which are based on the reputation and feedbacks, such as Amazon and eBay. However, a lot of problems and issues still remain open in the reputation based trust mechanisms.
- Finally, in the service composition research area, the QoS aggregation methods were normally driven on the numerical values, the categorical value type was rarely mentioned. In this thesis, the categorical values were discussed in the single service selection only. Therefore, the aggregation methods for the categorical values in the composite service selection need to be further investigated.

### 8.3 Future work

Based on the promising research findings presented in this thesis, several future research directions have been identified and planned. These future directions aim at making our approach more practically feasible. In particular, the following work will be carried out in the near future.

- The attribute based service customization was only discussed in the single service selection in Chapter 5. For the composite service, the attribute based access control and QoS customization methods will be further investigated. In

addition, the aggregation of the policies, the dependencies and conflicts of the policies in the composition will be addressed.

- For our Trust-Oriented Genetic Algorithm (TOGA), we only compared the efficiency and the effectiveness of TOGA with the exhaustive search method in Chapter 6. In the future, the performance of our TOGA will be compared with other optimization approaches, such as Particle Swarm Optimization (PSO).
- For service composition, we only considered some numerical QoS objectives, such as cost, availability, and response time. We will extend the developed TOGA to support optimizing multiple QoS objectives, especially the categorical ones. In addition, the QoS aggregation method for the categorical values will be further investigated.
- This thesis was focusing on service selection and composing the concrete services based on QoS. The opposite research direction of decomposing a composite service in its pieces by combining potential concrete services led by a given overall QoS goal would be an interesting research field.

# Bibliography

---

- [AAF<sup>+</sup>02] Assaf Arkin, Sid Askary, Scott Fordin, Wolfgang Jekeli, and Kohsuke Kawaguchi. Web service choreography interface (wsci) 1.0. <http://www.w3.org/TR/wsci/>, 2002.
- [ABH<sup>+</sup>02] Anupriya Ankolekar, Mark Burstein, Jerry Hobbs, Ora Lassila, David Martin, Drew McDermott, Sheila McIlraith, Srini Narayanan, Massimo Paolucci, Terry Payne, and Katia Sycara. Daml-s: Web service description for the semantic web. In Ian Horrocks and James Hendler, editors, *The Semantic Web*, volume 2342, pages 348–363. Springer Berlin, 2002.
- [Alo04] Gustavo Alonso. *Web services: concepts, architectures and applications*. Springer Verlag, 2004.
- [ANS96] ANSI. Business rules for the enterprise viewpoint of rm-odp. 7 December 1996.
- [AP07] Danilo Ardagna and Barbara Pernici. Adaptive service composition in flexible processes. *IEEE Transactions on Software Engineering*, 33(6):369–384, 2007.
- [AR09] Mohammad Alrifai and Thomas Risse. Combining global optimization with local selection for efficient qos-aware service composition. In *Proceedings of the 18th International Conference on World Wide Web*, pages 881–890, 2009.
- [Ark02] A. Arkin. Business process modeling language. Technical Report 1.0, 2002.

- [ASTL07] Poulia Adamopoulou, Evangelos Sakkopoulos, Athanasios K. Tsakalidis, and Miltiadis D. Lytras. Web service selection based on qos knowledge management. *J. UCS*, 13(9):1138–1156, 2007.
- [Bad08] Youakim Badr. Framework for web service selection based on non-functional properties. *International Journal of Web Services Practices*, 3(2):94–109, 2008.
- [BBMP07] Matteo Baldon, Cristina Baroglio, Alberto Martelli, and Viviana Patti. Reasoning about interaction protocols for customizing web service selection and composition. *Journal of Logic and Algebraic Programming*, 70:53–73, 2007.
- [BCCC04] Tom Bellwood, Steve Capell, Luc Clement, and John Colgrave. Universal description discovery and integration, 2004.
- [BFHS03] Tevfik Bultan, Xiang Fu, Richard Hull, and Jianwen Su. Conversation specification: a new approach to design and analysis of e-service composition. In *In Proceedings of the 12th international conference on World Wide Web*, pages 403–410, New York, NY, USA, 2003. ACM.
- [BSW07] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, SP '07, pages 321–334, Washington, DC, USA, 2007. IEEE Computer Society.
- [BW93] Jr Bubenko, J. and B. Wangler. Objectives driven capture of business rules and of information systems requirements. In *Proceedings of the International Conference on Systems, Man and Cybernetics*, pages 670–677, 1993.
- [Car06] J. Cardoso. Discovering semantic web services with and without a common ontology commitment. In *Proceedings of the IEEE International Conference on Service Computing*, pages 183–190, 2006.
- [CCMW04] Erik Christensen, Francisco Curbera, Greg Meredith, and Sanjiva Weerawarana. Web services description language (wsdl) 1.1, 2004.

- [CDK02] Francisco Curbera, Matthew J. Duftler, and Rania Khalaf. A platform for creating and executing bpm4ws processes. <http://www.alphaworks.ibm.com/tech/bpws4j>, 2002.
- [CLTW04] W.K. Cheung, Jiming Liu, Kevin H. Tsang, and R.K. Wong. Dynamic resource selection for service composition in the grid. In *Proceedings of the IEEE International Conference on Web Intelligence*, pages 412–418, 2004.
- [CM12] Jason Crampton and Charles Morisset. Ptacl: A language for attribute-based access control in open systems. In Pierpaolo Degano and Joshua D. Guttman, editors, *Principles of Security and Trust*, volume 7215 of *Lecture Notes in Computer Science*, pages 390–409. Springer Berlin Heidelberg, 2012.
- [CMK98] Il H. Cho, John D. McGregor, and Lee Krause. A protocol based approach to specifying interoperability between objects. In *Proceedings of the 26th Technology of Object-Oriented languages and Systems*, pages 84–96, 1998.
- [CPEV05a] Gerardo Canfora, Massimiliano Di Penta, Raffaele Esposito, and Maria Luisa Villani. An approach for qos-aware service composition based on genetic algorithm. In *Proceedings of the International Conference on Genetic and Evolutionary Computation*, pages 1069–1075, 2005.
- [CPEV05b] Gerardo Canfora, Massimiliano Di Penta, Raffaele Esposito, and Maria Luisa Villani. Qos-aware replanning of composite web services. In *ICWS*, pages 121–129, 2005.
- [CSM<sup>+</sup>04] Jorge Cardoso, Amit Sheth, John Miller, Jonathan Arnold, and Krys Kochut. Quality of service for workflows and web service processes. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(3):281–308, 2004.
- [Cur03] F. Curbera. Business process execution language for web services. Technical Report version 1.1, May 2003.

- [DAS07] D.Skoutas, A.Simitsis, and T. Sellis. A ranking mechanism for semantic web service discovery. In *Proceedings of the IEEE International Conference on Services*, pages 41–48, 2007.
- [DLS05] Glen Dobson, Russell Lock, and Ian Sommerville. Qosont: A qos ontology for service-centric systems. In *31th EUROMICRO Conference on Software Engineering and Advanced Application*, pages 80–87, 2005.
- [DPJ05] D.B.Claro, P.Albers, and J.K.Hao. Selecting web services for optimal composition. In *Proceedings of the 2nd International Workshop on Semantic and Dynamic Web Processes*, pages 32–45, 2005.
- [DSG06] V. Deora, J. Shao, and W. A. Gray. Modelling quality of service in service oriented computing. In *Proceedings of the Second IEEE International Symposium on Service-Oriented System Engineering*, pages 95–101, Washington, DC, USA, 2006. IEEE Computer Society.
- [DSGF04] V. Deora, J. Shao, W. Gray, and N. Fiddian. A quality of service management framework based on user expectations. *Service-Oriented Computing of Lecture Notes in Computer Science*, 2910:104–114, 2004.
- [DSGF06a] V. Deora, J. Shao, W. Gray, and N. Fiddian. Expectation-based quality of service assessment. *International Journal on Digital Libraries*, 6:260–269, 2006.
- [DSGF06b] V. Deora, J. Shao, W. Gray, and N. Fiddian. Expectation-based quality of service assessment. *International Journal on Digital Libraries*, 6:260–269, 2006.
- [DSL04] Seema Degwekar, Stanley Y.W. Su, and Herman Lam. Constraint specification and processing in web services publication and discovery. In *Proceedings of the 2004 IEEE International Conference on Web Service*, pages 210–218, 2004.
- [DSS<sup>+</sup>04] V. Deora, J. Shao, G. Shercliff, P.J.Stockreisser, W.A.Gray, and N.J.Fiddian. Incorporating qos specifications in service discovery.



- Web Information Systems Workshops of Lecture Notes in Computer Science*, 3307:252–263, 2004.
- [DZC09] D.Liu, Z.Shao, and C.Yu. Optimizing service selection by users qos expectation. In *Proceedings of the Internaitional MultiConference of Engineers and Computer Scientists*, pages 294–301, 2009.
- [El05] Shahram Esmailsabzali and Kate larson. Service allocation for composite web services based on quality attributes. In *Proceedings of the 7th IEEE International Conference on e-Commencer Technology Workshops*, pages 71–82, 2005.
- [ELMT09] Jeff A. Estefan, Ken Laskey, Francis G. McCabe, and Danny Thornton. Oasis reference architecture foundation for service oriented architecture 1.0 draft 2. <http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra-cd-02.pdf>, 2009.
- [Fie00] Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. Phd thesis, Information and Computer Science, University of California, Irvine, 2000.
- [GGD07] Stefania Galizia, Alessio Gugliotta, and John Domingue. A trust based methodology for web service selection. In *Proceedings of IEEE International Conference on Semantic Computing*, pages 193–200, 2007.
- [GHM<sup>+</sup>07] Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen, Anish Karmarkar, and Yves Lafon. Simple object access protocol, 2007.
- [GPST04] John Garofalakis, Yannis Panagis, Evangelos Sakkopoulos, and Athanasios Tsakalidis. Web service discovery mechanisms: Looking for a needle in a haystack? In *In: International Workshop on Web Engineering*, 2004.

- [GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM conference on Computer and communications security*, CCS '06, pages 89–98. ACM, 2006.
- [Gro05] Object Management Group. Uml profile for modeling qos and fault tolerance characteristics and mechanisms. [http://www.omg.org/technology/documents/profile\\_catalog.htm](http://www.omg.org/technology/documents/profile_catalog.htm), 2005.
- [HBM08] Philip F. Hingston, Luigi C. Barone, and Zbigniew Michalewicz, editors. *Design by Evolution: Advanced in Evolutionary Design*. Springer, 2008.
- [Her95] H. Herbst. A meta-model for specifying business rules in system analysis. In *Proceedings of the International Conference on Advanced Information Systems Engineering*, pages 186–199, 1995.
- [HH00] D. Hey and K.A. Healy. Defining business rules: What are they really? Technical Report Rev 1.3, the Business Rules Group, July 2000.
- [HMRR08] Joyce El Haddad, Maude Manouvrier, Guillermo Ramirez, and Marta Rukoz. Qos-driven selection of web services for transactional composition. In *ICWS*, pages 653–660, 2008.
- [HPZ06] Nan Hu, Paul A. Pavlou, and Jennifer Zhang. Can online reviews reveal a product’s true quality?: empirical findings and analytical modeling of online word-of-mouth communication. In *ACM Conference on Electronic Commerce*, pages 324–330, 2006.
- [hVHA05] Le hung Vu, Manfred Hauswirth, and Karl Aberer. Qos-based service selection and ranking with trust and reputation management. In *Proceedings of the Cooperative Information System Conference*, pages 446–483, 2005.
- [JdW04] Damien B. Jourdan and Olivier L. de Weck. Layout optimization for a wireless sensor network using a multi-objective genetic algorithm.

- In *Proceedings of IEEE 59th Vehicular Technology Conference*, pages 2466–2470, 2004.
- [JF06] Radu Jurca and Boi Faltings. Minimum payments that reward honest reputation feedback. In *ACM Conference on Electronic Commerce*, pages 190–199, 2006.
- [JF07] Radu Jurca and Boi Faltings. Collusion-resistant, incentive-compatible feedback payments. In *ACM Conference on Electronic Commerce*, pages 200–209, 2007.
- [JIB07a] Audun Jøsang, Roslan Ismail, and Colin Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, 2007.
- [JIB07b] Audun Jøsang, Roslan Ismail, and Colin Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, 2007.
- [JN04] Jingwen Jin and Klara Nahrstedt. On exploring performance optimizations in web service composition. In *Proceedings of the 5th ACM International Conference on Middleware*, pages 115–134, 2004.
- [JQ09] Audun Jøsang and Walter Quattrociocchi. Advanced features in bayesian reputation systems. In *TrustBus*, pages 105–114, 2009.
- [JRG05] Michael C. Jaeger, Gregor Rojec-Goldmann, and Gero Mühl. Qos aggregation in web service compositions. In *Proceedings of the IEEE International Conference on e-Technology, e-Commerce and e-Service*, pages 181–185, 2005.
- [JXX<sup>+</sup>10] C. Jiuxin, S. Xuesheng, Z. Xiao, L. Bo, and M. Bo. Efficient multi-objective service selection algorithm based on particle swarm optimization. In *Proceedings of the IEEE International Conference on Services Computing*, pages 603–608, 2010.
- [LC99] Jason D. Lohn and Silvano P. Colombano. A circuit representation technique for automated circuit design. *IEEE Transactions on Evolutionary Computation*, 3(3):205–219, 1999.

- [LCOZ10] Li Li, Peng Cheng, Ling Ou, and Zili Zhang. Applying multi-objective evolutionary algorithms to qos-aware web service composition. In *Proceedings of the 6th International Conference on Advanced Data Mining and Applications*, pages 270–281, 2010.
- [Ley01] Ford Leymann. Web services flow language (wsfl) 1.0. <http://xml.coverpages.org/WSFL-Guide-200110.pdf>, 2001.
- [LH03] Lei Li and Ian Horrocks. A software framework for matchmaking based on semantic web technology. In *Proceedings of the 12th international conference on World Wide Web*, pages 331–339, 2003.
- [LHH<sup>+</sup>11] Fei Li, Yanxiang He, Wensheng Hu, Libing Wu, and Peng Wen. Web service selection based on fuzzy qos attributes. *Journal of Computational Information Systems*, 7(1):198–205, 2011.
- [LKD<sup>+</sup>03] Heiko Ludwig, Alexander Keller, Asit Dan, Richard P. King, and Richard Franck. Web service level agreement language specification. <http://www.research.ibm.com/wsla/WSLASpecV1-20030128.pdf?ref=Guzels.TV>, 2003.
- [LLJ<sup>+</sup>05] Shulei Liu, Yunxiang Liu, Ning Jing, Guifen Tang, and Yu Tang. A dynamic web service selection strategy with qos global optimization based on multi-objective genetic algorithm. *Grid and Cooperative Computing of Lecture Notes in Computer Science*, 3795:84–89, 2005.
- [LNZ04a] Yutu Liu, Anne H. Ngu, and Liang Z. Zeng. Qos computation and policing in dynamic web service selection. In *Proceedings of the 13th International Conference on World Wide Web*, pages 66–73, 2004.
- [LNZ04b] Yutu Liu, Anne H. Ngu, and Liang Z. Zeng. Qos computation and policing in dynamic web service selection. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, WWW Alt. '04*, pages 66–73, New York, NY, USA, 2004. ACM.
- [LRZ11] Lu Li, Mei Rong, and Guangquan Zhang. A web service qos prediction approach based on multi-dimension qos. In *Proceedings of the*

- 6th International Conference on Computer Science and Education*, pages 1319–1322, 2011.
- [LW10] Lei Li and Yan Wang. Subjective trust inference in composite services. In *AAAI*, 2010.
- [LWL10] Lei Li, Yan Wang, and Ee-Peng Lim. Trust-oriented composite service selection with qos constraints. *Journal of Universal Computer Science*, 16(13):1720–1744, 2010.
- [LYO<sup>+</sup>10] Li Li, Pengyi Yang, Ling Ou, Zili Zhang, and Peng Cheng. Genetic algorithm-based multi-objective optimization for qos-aware web services composition. *Knowledge Science, Engineering and Management of Lecture Notes in Computer Science*, 6291:549–554, 2010.
- [Men02] Daniel A. Menascé. Qos issues in web services. *IEEE Internet Computing*, 6(6):72–75, 2002.
- [Men04] Daniel A. Menascé. Composing web services: A qos view. *IEEE Internet Computing*, 8(6):88–90, 2004.
- [MG07] M.C.Jaeger and G.Muhl. Qos-based selection of services: the implementation of a genetic algorithm. In *Proceedings of International Conference on Service-Oriented Architectures and Service Oriented Computing Workshop*, pages 359–370, 2007.
- [MM04] Nikola Milanovic and Miroslaw Malek. Current solutions for web service composition. *Internet Computing*, 8(6):51–59, 2004.
- [MPSF08] M.P.Papazoglou, P.Traverso, S.Dustdar, and F.Leymann. Service-oriented computing: a research roadmap. *Int.J.Cooperative Inf. Syst*, 17(2):223–255, 2008.
- [MR99] M.Gen and R.Cheng. *Genetic Algorithms and Engineering Optimization (Engineering Design and Automation)*. Wiley-Interscience, 1999.
- [MRG07] Daniel A. Menascé, Honglei Ruan, and Hassan Gomaa. Qos management in service-oriented architectures. *Performance Evaluation*, 64(7):646–663, 2007.

- [MS01] A. Maedhe and S. Staab. Comparing ontologies-similarity measures and a comparison study. Internal report, AIFB Univeristy of Karlsruhe, Germany, 2001.
- [MS02a] A. Maedhe and S. Staab. Measuring similarity between ontologies. In *Proceedings of the European Conference on Knowledge Acquisition and Management*, pages 119–225, 2002.
- [MS02b] Sheila McIlraith and Tran Cao Son. Adapting golog for compsoition of semantic web services. In *International Conference on Principles of Knowledge Representation and Reasoning*, pages 369–384, 2002.
- [MSD02] Vijay Machiraju, Akhil Sahai, and Anna Durante. Towards automated sla management for web services. <http://www.hpl.hp.com/techreports/2001/HPL-2001-310R1.pdf>, 2002.
- [OEH02] Justin O’Sullivan, David Edmond, and Arthur Ter Hofstede. What’s in a service? *Distributed Parallel Databases*, 12(2):117–133, 2002.
- [Ope09] Opengroup soa reference architecture. <https://www.opengroup.org/projects/soa-ref-arch/uploads/40/19713/soa-ra-public-050609.pdf>, 2009.
- [Pap03] Michael P. Papazoglou. Web services and business transactions. *World Wide Web*, 6:49–91, 2003.
- [Pap08] Michael Papazoglou. *Web services: principles and technology*. Addison Wesley, 2008.
- [Pel03] Chris Peltz. Web services orchestration and choreography. *Computer*, 36(10):46–52, 2003.
- [PTR06] Ioannis V. Papaioannou, Dimitrios T. Tsesmetzis, Ioanna G. Rousaki, and Miltiades E. Anagnostou. A qos ontology language for web-servies. In *Proceedings of the 20th International Conference on Advanced Information networking and Applications*, pages 101–106, Washington, DC, USA, 2006. IEEE Computer Society.
- [Ran03] Shuping Ran. A model for web services discovery with qos. *SIGecom Exch*, 4(1):1–10, 2003.

- [Ren09] Colin Renouf, editor. *Pro(IBM) WebSphere Application Server 7 Internals*. Springer, 2009.
- [RGP<sup>+</sup>02] Ignacio Rojas, Jesús González, Héctor Pomares, Juan J. Merelo Guervós, Pedro A. Castillo Valdivieso, and Gustavo Romero. Statistical analysis of the main parameters involved in the design of a genetic algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 32(1):31–37, 2002.
- [SF07] Yimin Shen and Yushun Fan. Cooperative mixed strategy for service selection in service oriented architecture. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, pages 1452–1457, 2007.
- [SHP03] Evren Sirin, James Hendler, and Bijan Parsia. Semi-automatic composition of web services using semantic descriptions. In *Web Services: Modeling, Architecture and Infrastructure workshop in conjunction with ICEIS*, pages 167–179, 2003.
- [SJM06] Poornachandra Sarang, Matjaz B. Juric, and Benny Mathew. *Business Process Execution Language for Web Services*. Packt Publishing, 2006.
- [SLS09] Dong H. Shin, Kyong H. Lee, and Tatsuya Suda. Automated generation of compiste web services based on functional semantics. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 7:332–343, 2009.
- [SP94a] M. Srinivas and Lalit M. Patnaik. Genetic algorithms: A survey. *IEEE Computer*, 27(6):17–26, 1994.
- [SP94b] M. Srinivas and Lalit M. Patnaik. Genetic algorithms: A survey. *IEEE Computer*, 27(6):17–26, 1994.
- [SSML02] Katia Sycara, Widoff S, Klusch M, and Jianguo Lu. Larks: Dynamic matchmaking among heterogeneous software agents in cyberspace. In *Proceedings of Cyberspace, Autonomous Agents, and Multi-Agent Systems*, pages 173–203, 2002.

- [SWY10] Qibo Sun, Shangguang Wang, and Fangchun Yang. Quick service selection approach based on particle swarm optimization. In *Proceedings of the IEEE International Conference on Bio-Inspired Computing: Theories and Applications*, pages 278–284, 2010.
- [TAB08] Hassina Nacer Talantikite, Djamil Aissani, and Nacer Boudjlida. Semantic annotations for web services discovery and composition. *Journal of Computer Standards and Interfaces*, 32:1–10, 2008.
- [TGRS04] M. Tian, A. Gramm, H. Ritter, and J. Schiller. Efficient selection and monitoring of qos-aware web services with the ws-qos framework. In *Proceedings of the IEEE International Conference of Web Intelligence*, pages 152–158, 2004.
- [Tha01] Satish Thatte. Xlang: Web services for business process design. <http://xml.coverpages.org/XLANG-C-200106.html>, 2001.
- [THHD11] Thanh Thoa Pham Thi, Markus Helfert, Fakir Hossain, and Thang Le Dinh. Discovering business rules from business process models. In *Proceedings of the International Conference on Computer Systems and Technologies*, pages 259–265, 2011.
- [TP02] Aphrodite Tsalgatidou and Thomi Pilioura. An overview of standards and related technology in web services. *Distributed and Parallel Databases*, 12:135–162, 2002.
- [TPP02] Vladimir Tasic, Kruti Patel, and Bernard Pagurek. Wsol-web service offerings language. *Web Services, e-Business, and the Semantic Web of Lecture Notes in Computer Science*, 2512:57–67, 2002.
- [TPP<sup>+</sup>05] Vladimir Tasic, Bernard Pagurek, Kruti Patel, Babak Esfandiari, and Wei Ma. Management applications of the web service offerings language (wsol). *Information Systems*, 30(7):564–586, 2005.
- [vdAtHKB03] W.M.P. van der Aalst, A. H. M. ter Hofstede, B. Kiepuszewski, and A. P. Barros. Workflow patterns. *Distributed and Parallel Databases*, 14(1):5–51, 2003.



- [WCLH06] Ping Wang, Kuoming Chao, Chichun Lo, and Chunlung Huang. A fuzzy model for selection of qos-aware web services. In *Proceedings of the IEEE International Conference on e-Business Engineering*, pages 585–593, 2006.
- [WDWY99] W.H.Chen, D.J.Ballance, W.Feng, and Y.Li. Genetic algorithm enabled computer-automated design of qft control systems. In *Proceedings of the IEEE International Symposium on Computer Aided Control System Design*, pages 492–497, 1999.
- [WMC10] Rongxi Wang, Li Ma, and Yanping Chen. The research of web service selection based on the ant colony algorithm. In *Proceedings of International Conference on Artificial Intelligence and Computational Intelligence*, pages 551–555, 2010.
- [Woo07] Daniel Woolston, editor. *Foundations of BizTalk server 2006*. Apress, 2007.
- [WS03] Y. Wang and E. Stroulia. Semantic structure matching for assessing web-service similarity. In *Proceedings of the IEEE International Conference on Service-Oriented Computing*, pages 82–87, 2003.
- [WSH<sup>+</sup>04] Dan Wu, Evren Sirin, James A. Hendler, Dana S. Nau, and Bijan Parsia. Htn planning for web service composition using shop2. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(4):377–396, 2004.
- [XB05] Jin Xiao and Raouf Boutaba. Qos-aware service composition and adaptation in autonomic communication. *IEEE Journal on Selected Areas in Communications*, 23(12):2344–2360, 2005.
- [XCL<sup>+</sup>09] Hong Xia, Yan Chen, Zengzhi Li, Haichang Gao, and Yanping Chen. Web service selection algorithm based on particle swarm optimization. In *Proceedings of IEEE International Conference on Dependable, Autonomic and Secure Computing*, pages 467–472, 2009.

- [YK05] Tao Yu and K.J.Lin. A broker-based framework for qos-aware web service composition. In *Proceedings of the IEEE International Conference on e-Technology, e-Commerce and e-Service*, pages 22–29, 2005.
- [YL04] Tao Yu and Kwei-Jay Lin. The design of qos broker algorithms for qos-capable web services. In *Proceedings of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04)*, EEE '04, pages 17–24, Washington, DC, USA, 2004. IEEE Computer Society.
- [YP06] Jian Yang and Mike Papazoglou. Web component: A substrate for web service reuse and composition. *Lecture Notes in Computer Science*, 2348:21–36, 2006.
- [YRM08] Hong QIng Yu and Stephan Reiff-Marganiec. A method for automated web service selection. In *Proceedings of the IEEE Congress on Services*, pages 513–520, 2008.
- [YZL07] Tao Yu, Yue Zhang, and Kwei-Jay Lin. Efficient algorithms for web services selection with end-to-end qos constraints. *TWEB*, 1(1), 2007.
- [ZBD<sup>+</sup>03] Liangzhao Zeng, Boualem Benatallah, Marlon Dumas, Jayant Kalagnanam, and Quan Z. Sheng. Quality driven web services composition. In *WWW*, pages 411–421, 2003.
- [ZBN<sup>+</sup>04] Liangzhao Zeng, Boualem Benatallah, Anne H. H. Ngu, Marlon Dumas, Jayant Kalagnanam, and Henry Chang. Qos-aware middleware for web services composition. *IEEE Trans. Software Eng.*, 30(5):311–327, 2004.
- [ZLC10] Liangzhao Zeng, Hui Lei, and Henry Chang. Monitoring the qos for web services. *Service-Oriented Computing of Lecture Notes in Computer Science*, 4749:132–144, 2010.
- [ZSC06] Chengwen Zhang, Sen Su, and Junliang Chen. A novel genetic algorithm for qos-aware web services selection. In *DEECS*, pages 224–235, 2006.

- 
- [ZSX06] Baopeng Zhang, Yuanchun Shi, and Xin Xiao. A policy-driven service composition method for adaptation in pervasive computing environment. In *First International Symposium on Pervasive Computing and Applications*, pages 619–624, 2006.
- [ZW97] Amy M. Zaremski and Jeannette M. Wing. Specifications matching of software components. *ACM Transactions on Software engineering and Methodology*, 6(4):333–369, 1997.
- [ZYZ10] Huiyuan Zheng, Jian Yang, and Weiliang Zhao. Qos analysis and service selection for composite services. In *IEEE International Conference on Service Computing*, pages 425–433, 2010.