國 立 交 通 大 學

資訊管理研究所

博士論文

以共識為基礎的服務搜尋機制

An Approach to Consensus-Based Service Discovery

研　究　生：黃俊龍

指 導 教 授：羅濟群

共同指導教授：趙國銘

中華民國 九十五 年 九 月

以共識為基礎的服務搜尋機制

An Approach to Consensus-Based Service Discovery

研　究　生：黃俊龍　　Student：Chun-Lung Huang

指導教授：羅濟群　　Advisor：Chi-Chun Lo

共同指導教授：趙國銘　　Co-Advisor：Kuo-Ming Chao

國　立　交　通　大　學

資訊管理研究所

博士論文

A Dissertation

Submitted to Institute of Information Management

College of Management

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Information Management

September 2006

Hsinchu, Taiwan, Republic of China

中華民國　九十五　年　九　月

以共識為基礎的服務搜尋機制

學生：黃俊龍　　　　　　　　　　　　　　　　指導教授：羅濟群 博士

共同指導教授：趙國銘 博士

國立交通大學資訊管理研究所

中文摘要

網路服務(Web services)廣泛的被應用在電子商務、網格服務(Grid Services)與電子化政府等各種不同的領域，它通常分散在異質網路環境之中，而且能夠透過整合(composition)被加以再利用並衍生出其他功能更強大的服務。有效率的服務搜尋機制是網路服務能否被充分利用的先決條件，然而網路服務的搜尋首先遭遇到的問題就是搜尋機制通常根據服務本身所提供的說明標籤、介面的特徵或者服務的功能性描述進行搜尋，無法根據服務本身所包含的內容(content)本身進行搜尋。換言之，服務使用者需要靠其他額外的判斷準則，方能在許多重複或者是相似的服務之中選取真正適合的服務；其次、目前的研究趨勢在於如何動態的搜尋，並同時整合既有的網路服務。唯大部分現有的服務搜尋方法，無論是由服務的功能性或者非功能性觀點進行搜尋，都無法有效解決認知差距所造成之搜尋不準確的問題。因為不論服務使用者對網路服務進行搜尋，或者服務提供者對服務進行宣傳時，對服務內容通常抱持著不同的主觀期待或是偏好，由於雙方對於服務內容有不同的觀點或評價，導致正確搜尋服務的效率顯得不盡人意。本研究提出一個以共識為基礎的服務搜尋方法，嘗試來降低因認知差距所引起的搜尋不準確問題。此法能夠收集個人主觀的意見，以模糊化的方式加以表達，並協助服務的使用者與提供者在進行搜尋之前先達成一個共識，藉以提升服務搜尋的準確率。本法採用一系列的模糊群體決策制定理論以及語意網路(Semantic Web)的技術，使得不同的主觀意見得以整合成為一個共識並利用它對服務內容進行分類與搜尋。本法可以重複的被執行，因此新的主觀意見得以被納入並且持續提升搜尋的效率。此外除了將研究開發成雛型系統之外，也將利用各種不同的數據實際測試之，期待實驗的結果能夠展示出本研究所提方法的效能。

關鍵詞：調節式模糊網路服務搜尋、群體共識、模糊喜好關係、語意查詢、語意網路

An Approach to Consensus-Based Service Discovery

Student: Chun-Lung Huang Advisor: Dr. Chi-Chun Lo
Co-Advisor: Dr. Kuo-Ming Chao

Institute of Information Management
National Chiao Tung University

Abstract

Web services are used for developing and integrating highly distributed and heterogeneous systems in different domains such as e-business, Grid services, and e-government systems. Service discovery is a crucial process for Web service utilization. However, the issues associated with service discovery that involves the Web service having data repositories are not well addressed by the existing methods which focus on service capabilities, interface signatures, or functionalities. These methods are inadequate to identify appropriate services among the services which have similar functionalities, so it requires service consumers to include additional aspects (i.e. content of service or reputation) to evaluate the services. An effective service discovery mechanism is able to support the identification of the required services in a dynamic environment and form composite services that provide the required functionality. The service consumers and providers often have different views on the content of services. The existing service discovery approaches, based on either functional or non-functional attributes, do not address the issues associated with the impact of the diverse preferences and subjective expectations of the service consumers and providers which are generally used in searching for or in advertising Web services. This study attempts to alleviate such diversity by proposing a consensus-based service discovery approach to model subjective and fuzzy opinions. This will assist service consumers and providers in reaching a common consensus so that the efficiency of service discovery can be increased. The approach which is based on fuzzy group decision making methods and semantic web technologies can be executed iteratively and therefore further fuzzy opinions and preferences can be added to improve the precision of Web service discovery. The proposed approach will be implemented as a prototype system and to be tested through various experiments in order to demonstrate the effectiveness of the proposed approach.

Keywords: Moderated Fuzzy Discovery, Web Services Discovery and Composition, Group Consensus, Fuzzy Preference Relations, Linguistic Query, Semantic Web, POPM

誌　謝

本論文得以順利完成，首先得感謝我的兩位指導教授——羅濟群博士(台灣)，以及趙國銘博士(英國)於研究過程中的悉心指導。在交通大學碩、博士修業的七年當中，幸有羅老師一路的提攜，使我在研究方向的規劃上，以及研究能力的養成上都能有長足的進步，特別是專案開發的經驗，為我累積了許多技術上的基礎；旅英期間則有賴趙教授在研究方法、研究討論、文章修改、論文投稿以及國外生活上所給予的極大協助，使我能有愉快且充實的國外求學經驗。兩位老師豐富的經驗、宏觀的視野與嚴謹的研究態度讓我受益良多，進而能奠定我日後獨立研究的能力。在此謹致上我最誠摯的感謝之意。

同時由衷地感謝口試委員陳彥良老師、陳鴻基老師、楊建民老師、劉敦仁老師與蔡銘箴老師對論文的細心審閱與費心指導，讓本篇論文的內容能夠更加充實、完善；在此也特別的感謝英國的 Nick Godwin 博士，感謝他對本論文所提供的諸多建議，尤其是在英文文章的潤稿與修訂上所給予的莫大協助。

除了師長們在研究上的教導與指正，在此同時感謝本所諸位優秀的同窗夥伴們，王平、俊傑、宗智、曉芸、鵬雲、怡鎮、芳萍、明華、榮發、順吉、秉文、志坤、美玉、怡璋、鍾斌等人，有你們的陪伴使我在漫長的研究路上不感孤單。尤其王平學長在研究內容上，所適時給予的具體建議讓我備感受用；也感謝通訊網路研究室的學弟妹，道馨、鼎元以及真理大學孫鵬虬等人，在研究期間於學業與生活上的相互支援。也謝謝本所助理淑惠、秀文在許多事務上的幫忙。

最後，更要感謝我的父母與家人一路上的支援，以及女友瓊瑤的鼓勵與支持，讓我能夠無後顧之憂地攻讀學位。如今，完成學業、論文付梓，謹獻上最真誠的謝意給我親愛的家人，並與他們分享這喜悅的一刻。本論文謹獻給任何幫助我的朋友們。謝謝。

黃俊龍

國立交通大學資訊管理研究所

中國民國九十五年九月

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION

1.1 Research background

The Web was originally used for sharing information among scientists. At the beginning, this Document Web was used to store and share static information but it has gained increasing attention due to the advance of Internet technologies and the enhancement of hardware capabilities. The Web has changed its direction into being a Service Web which provides not only support for document sharing, but also the ability to enable organizations to provide their applications or services via the Web – the Internet. Consequently, the concept of Web services [1],[2],[3] has become widely relevant.

A Web service is a set of related functionalities which can be automatically accessed through the Web [3] and it makes information and software available and executable over the Internet, so it can be utilized as a building block for applications [4]. Web services can be registered (advertised) and queried (searched) by using Universal Description, Discovery and Integration (UDDI) [5]. In the meantime the Web Services Description Language (WSDL) [6] and the Simple Object Access Protocol (SOAP) [10] provide a machine-processable interface in which components that contribute to a composite web service can be executed automatically.

Two types of Web services are identified: *simple* and *composite* [9]. A *simple* service, a primitive service, is standalone and Internet-based application which can fulfill consumers' requests without any other Web services. A *composite* service is conceived as a conglomeration of several simple Web services over a designated flow structure [7],[8]. These primitive services can be grouped and they interact with each other to provide consumers a complete value-added service. For example, car dealer, insurance, and financial loan services can be combined as a comprehensive car sale service.

Widely available and standardized Web service technologies make collaboration among different organizations possible. Web services are now used for developing and integrating highly distributed and heterogeneous systems in different domains such as e-business, grid services, and e-government systems. With service popularity and complexity, the concept of Service Oriented Architecture (SOA) or Service Oriented Computing (SOC) [23],[38],[55] has been introduced and it has gained increasing significance in the research field of information systems. It attempts to provide a systematic approach for service composition in order to achieve Web services (components) sharing and reuse.

Two related concept in SOA / SOC are composite services and service composition. A composite service is a *complex* Web service which is composed of a number of *simple* (primitive) Web services. Service composition is the construction process of composite services [8]. Ideally, different primitive Web services within a *complex* Web service can come from diverse service providers and this leads to a new arduous problem – the issue of service semantics. Even though one output parameter of one primitive Web service has the same name and type with an input parameter of another primitive Web service, it is not necessary that these parameters can be linked up to form a consistent composite service.

The Semantic Web [13] has been proposed to dispose of the semantic issues. Many academic researchers and developers are endeavouring to build ontology for Web services. The goal of the Semantic Web is to make the Web services not only understandable by humans but also by machines through adding semantic information to the advertised services and to the service requirement specifications in order to increase interoperability among primitive Web services. Several standards such as the Web Ontology Language (OWL) [14] and the Web Ontology Language for Services (OWL-S or formerly DAML-S) [15] have been introduced to achieve this goal. These standards help to handle the semantic issues that occur in Web service utilization.

1.2 Motivation and objective

In order to fully realize the benefit of the automation of Web service composition, service discovery is a crucial process for Web service utilization. Most of existing service composition approaches [38],[42],[43] assume that all primitive services are ready-to-use in someplace or can be identified via simple UDDI queries. Discovery or so-called matchmaking[1] is considered as a search problem in a bounded space. It takes service consumers' requests and a collection of services from services providers as input via its discovery mechanism to identify a list of best matched pairs. Some researches [16],[18] handle the discovery problem by incorporating Semantic Web technologies. However, there are still two major problems that should be tackled in order to achieve an effective and efficient discovery process.

Principally, the issues associated with service discovery that involve the Web service having data repositories are not well addressed by the existing methods which focus on service capabilities, interface signatures, or functionalities. These methods are inadequate to identify appropriate services among the services which have similar functionalities, so it requires service consumers to include additional aspects (i.e. content of service or reputation) to evaluate services. In the context of Web service discovery, the representation of services and the selection of searching criteria are the critical factors to determine the quality of the output. Measuring the similarity between a service consumer's requests and the provided services in terms of: the software signatures; the capabilities; the syntax, and the semantics of services is a common element in discovery [12],[16],[18],[51]. However, recent work on

[1] In this thesis, the term *discovery* will be used for consistency when discussing search or matchmaking.

Web service discovery has not paid sufficient attention to the use of underlying data and information about services as a search criterion.

Second, an effective service discovery mechanism to enable the formation of the required composite services that provide the required functionality is able to support the identification of the required services in a dynamic environment. The service consumers and providers often have different views on the content of services. Most of the existing service discovery approaches such as [12],[16],[17],[18],[19],[49],[50] and [51], based on either functional or non-functional attributes, do not address the issues associated with the impact of the diverse preferences and subjective expectations of the service consumers and providers which are generally used in searching for or in advertising Web services. Because consumers and providers often have different views on the content of services, the selected results may not conform to consumers' expectations and this hinders the efficiency of service discovery.

This study proposed a consensus-based service discovery approach which attempts to use the underlying data and information about services as a search criterion (quality rating). Pre-classified services provide supplementary information with a higher level of abstraction, such as a quality rating for *Cheap*. This represents the capabilities and the underlying data associated with services. The proposed method attempts to refine the search space and to increase the precision rate in discovery. In the meanwhile, consumers are allowed to do search by using linguistic terms such as *Cheap* or *Comfortable*. Moreover, this consensus-based approach models subjective and fuzzy opinions and assists service consumers and providers in reaching a common consensus so that the cognitive differences among service consumers can be mitigated and the efficiency of service discovery can be increased. This approach, which is based on fuzzy group decision making methods and Semantic Web technologies, can be executed iteratively and therefore further fuzzy opinions and preferences can be added to improve the precision of Web service discovery. The proposed approach will be implemented as a prototype system and tested through various experiments to demonstrate the effectiveness of the proposed approach.

1.3 Research approach

In this study, five research steps were adopted to solve the problems of service discovery mentioned above. These are described as follows:

(1) Overall literature review: reviewing the existing works about the Web services, Web services composition, Service Oriented Architecture (SOA) / Service Oriented Computing (SOC), the Semantic Web, and the tools for realizing these concepts.

(2) Research concept development: the research concept was generated mainly through literature review. In addition brainstorming and consultation with experts at international conferences was used to help identify a viable solution to the problem.

(3) Detailed literature review: the use of fuzzy set concepts being identified the research process focussed on studies of fuzzy set and theory, related to reaching a consensus. These were concerns with concepts such as: fuzzy opinion representation; fuzzy majorities; fuzzy similarity measurement; fuzzy aggregation; reaching consensus; resolution methods for group decision problems; and, the methodologies used to collect imprecise preferences.

(4) Architecture development: the proposed approach was implemented as a prototype system to solve the two major issues of service discovery mentioned in Section 1.2. The main modules were employed to pre-classify the Web services in terms of different QoS terms. A method for reaching consensus over these terms used among service consumers and providers was implemented.

(5) Architecture verification: the proposed approach was tested through various experiments in which the data were collected from several famous website. The results demonstrate the effectiveness of the proposed approach.

1.4 Organization

The rest of this dissertation is structured as follows. Chapter 2 contains the literature reviews which include the descriptions of Web service composition, discovery and its related technologies. Chapter 3 will describe the fuzzy set theories which are used in reaching a consensus. The proposed architecture, including its scope, constraints and implementation considerations, will be presented in Chapter 4. Case studies and performance evaluations are presented in Chapter 5. Chapter 6 concludes this work and proposes the future work. Finally, the references and appendix are attached at the end of the dissertation.

CHAPTER 2 LITERATURE REVIEW

Based on SOA / SOC, several Web services can be assembled into a composite web service. In this chapter, literature related to Web services, service composition, service discovery, and Semantic Web, will be reviewed and discussed briefly.

2.1 Web service

The major differences between traditional Web applications and Web services is that the former are designed to be read and used by human and the latter are designed to increase the interoperability among machines by providing a machine-processable format which can help Web services to be searched and utilized autonomously. A W3C definition for Web services is: *"A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards."* [20]

In other words, Web services are self-contained, self-describing and modular applications which can be published, located, and invoked across the web [22]. At the initial stage, Web service is just a concept, but with the efforts contributed by the worldwide researchers and developers, many standards [1],[5],[6],[10],[13],[14],[15] have been proposed to support the implementation of Web services. Up to now, Web services are treated as the basic components of the Service Oriented Architecture (SOA) shown in Figure 2-1.

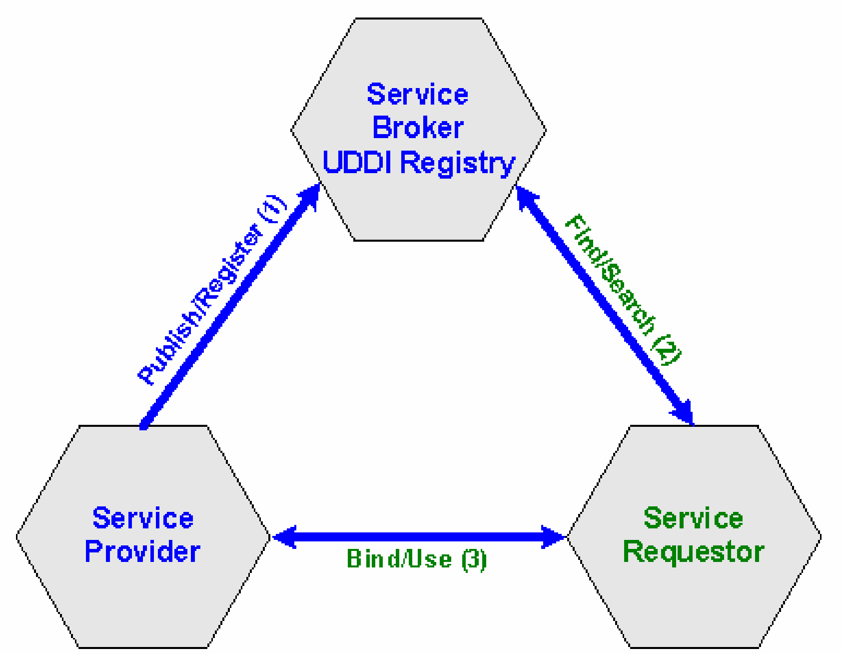

Figure 2-1 Service Oriented Architecture [23]

Three different roles are identified within a SOA which are Service Broker, Service Provider and Service Requestor. A Service Provider is the one who actually builds Web services and provides services for Service Requestors to consume. The Service Provider can choose either actively or passively to advertise its Web services with detail descriptions on the Web. A Service Requestor is the service consumer which seeks the Web services it needs. A Service Broker acts as the mediator for Web services. It provides Service Providers the ability to advertise their services in a registry and provides Service Requestors a channel to find the suitable Web services. A Service Broker helps Service Providers and Service Requestors find each other, but it is not necessarily for the broker to deal with the subsequent contracts and executions.

There are several technologies that contribute to the realization of a SOA such as: Universal Description, Discovery and Integration (UDDI) [5]: Web Services Description Language (WSDL) [6]: Simple Object Access Protocol (SOAP) [10]: and, Web Ontology Language for Services (OWL-S or formerly DAML-S) [15]. The detail of these technologies will be explained in the following sections.

2.1.1 Universal Description, Discovery and Integration (UDDI)

Web services can be registered (advertised) and queried (searched) by using Universal Description, Discovery and Integration (UDDI) [5]. It is provided by the OASIS Standard and consists of an XML schema for defining its data structures and APIs. Microsoft and IBM made considerable efforts to develop the UDDI specifications to support more complicated business logic and to promote UDDI as a public standard [23].

Table 2-1 Four core data structures in UDDI Registry [5],[23]

Categories	Type	Description
White pages	businessEntity	Containing descriptive information about a business or organization such as the name, address, telephone number, and other contact information of a given business.
Yellow pages	businessService	Describing a service belong to a businessEntity. Representing a logical service and containing descriptive information in business terms such as the names and categories of the services.
Green pages	bindingTemplate	Providing technical information necessary to invoke a Web service, typically given in the form of a URL and information about method names, argument types, and so on.
	tModel	Service Specification Detail: This is metadata about the various specifications implemented by a given Web service. Human and programs can discover how to interact with Web services they do not know much about.

UDDI acts as a directory service. A Service Provider implements its Web services and registers them along with their detail descriptions at an UDDI server. When a Service Requestor seeks for a service from UDDI, the UDDI server looks up the database for the suitable services by searching the detail descriptions from a list of the registered services.

Once a matched service is found, UDDI returns the identified service and its related information to the Service Requestor. These descriptions are classified into four core data types which are shown in Table 2-1.

One businessEntity (business company) could have many businessServices. Besides, one businessService contains a list of binding Templates that in turn contain a tModel. The bindingTemplate and the tModel represent the technical information about how to access and exploit Web services. As mentioned above UDDI acts as a directory service, Service Requestors use SOAP (detail at 2.1.3) as the communication protocol to negotiate with a UDDI server to obtain the WSDL (detail at 2.1.2) information of any suitable Web services. Figure 2-2 shows the necessary detailed descriptions when a Service Provider advertises its services in a UDDI registry server.

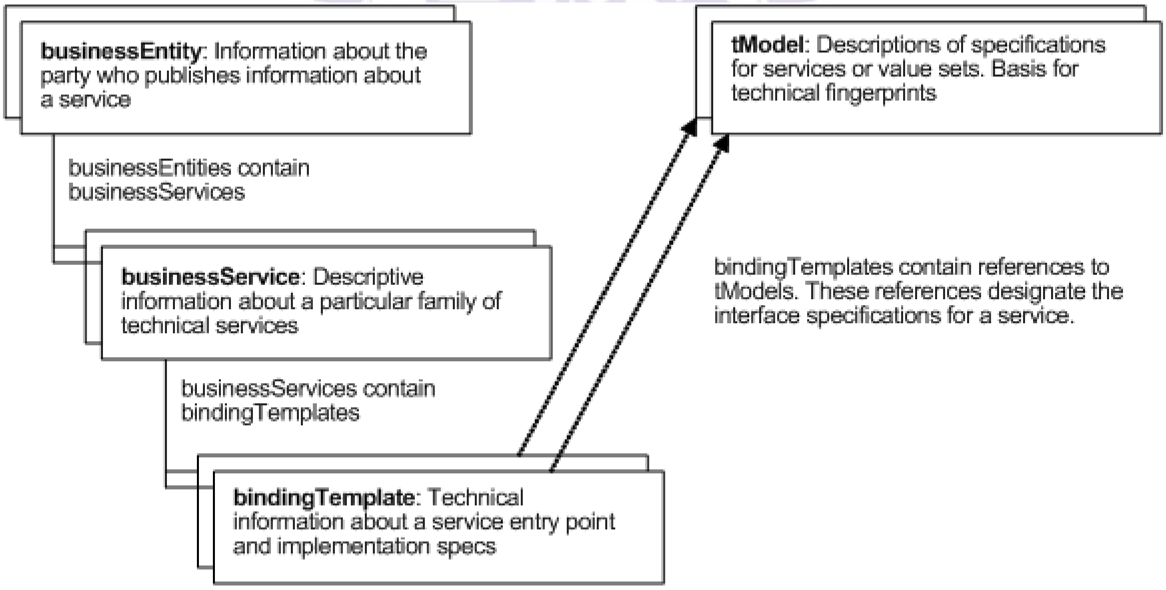

Figure 2-2 The UDDI registry information model [24]

2.1.2 Simple Object Access Protocol (SOAP)

Simple Object Access Protocol (SOAP) [10] is a text-based (specifically XML-based) communication protocol which can be conveyed by other underlying transmission protocols such as HTTP, SMTP, FTP or any other proprietary messaging protocol. SOAP messages increase the interoperability of different executable components of Web services when these components are distributed in the heterogeneous environment.

In other words, SOAP is a tool used to exchange data. SOAP uses XML documents to package and exchange messages among diverse counterparts. A SOAP message is composed of four major elements which are briefly in Table 2-2.

Table 2-2 SOAP elements [10][26]

Element	Descriptions
<Envolope>	The root element of a SOAP message. Specify the encodingStyle and namespace xmlns:soap= http://www.w3.org/2001/12/soap-envelope
<Header>	The application specific information (like authentication, payment and etc…)
<Body>	Actual SOAP message (transmitted data)
<Fault>	Fault handling information and rules when a identified error occurs

Figure 2-3 and Figure 2-4 are two simple examples to illustrate how SOAP messages are used to carry data. The former one shows that a *name-value* pair for a parameter is *Item-Apples* which is a query request to a Web service and the response *name-value* pair is *price-1.90* that indicates that the price for item Apples is 1.90.

```
<?xml version="1.0"?>
  <soap:Envelope
    xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
    soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

  <soap:Body>
    <m:GetPrice xmlns:m="http://www.w3schools.com/prices">
      <m:Item>Apples</m:Item>
    </m:GetPrice>
  </soap:Body>

</soap:Envelope>
```

Figure 2-3 An example for SOAP message to request an inquiry [26]

SOAP was created by Microsoft for the data exchange in Microsoft .Net framework and SOAP 1.1 was proposed to W3C in May 2000 [26] which means that SOAP is widely accepted by the community. The architecture proposed by this dissertation is also powered by SOAP due to its convenient referencing capabilities. In addition to the SOAP specification, some Web service APIs are also available for building Web services [21].

```
<?xml version="1.0"?>
<soap:Envelope
   xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
   soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

   <soap:Body>
      <m:GetPriceResponse xmlns:m="http://www.w3schools.com/prices">
         <m:Price>1.90</m:Price>
      </m:GetPriceResponse>
   </soap:Body>

</soap:Envelope>
```

Figure 2-4 An example for SOAP message to response an inquiry [26]

2.1.3 Web Services Description Language (WSDL)

Web Services Description Language (WSDL) [6] provides a machine-processable interface through which components that contribute to a composite Web service can be executed automatically. WSDL is also based on XML technology (an XML document) and is used to define how to describe the details of a Web service, such as the location of the service and the operations (methods) the service exposes. Those messages themselves are described abstractly and then bound to a concrete network protocol and message format [20].

WSDL service definitions provide documentation for distributed systems and are used as recipes for automating operation invocation [22]. WSDL allows descriptions of Web services and their messages to be represented explicitly in a way that facilitates their communication no matter what message formats or network protocols used. WSDL is now used in conjunction with SOAP 1.1, HTTP GET/POST, and MIME [6].

The major elements with in a WSDL document are shown in Table 2-3. The element <portType> is the most important WSDL element. It describes a Web service and its operation (like subroutine or function) which can be invoked and the messages involved. The element <message> defines the name and type for a specific parameter. The <binding> element defines the message format and protocol details for each <portType>.

Table 2-3 The basic WSDL document structure [25]

Element	Descriptions
<portType>	The operations performed by the Web service
<message>	The messages used by the Web service
<types>	The data types used by the Web service
<binding>	The communication protocols used by the Web service

```
<definition>
   <message name="getTermRequest">
      <part name="term" type="xs:string"/>
   </message>
   <message name="getTermResponse">
      <part name="value" type="xs:string"/>
   </message>

   <portType name="glossaryTerms">
      <operation name="getTerm">
         <input message="getTermRequest"/>
         <output message="getTermResponse"/>
      </operation>
   </portType>

   <binding type="glossaryTerms" name="b1">
      <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
      <operation>
         <soap:operation soapAction="http://example.com/getTerm"/>
            <input>
               <soap:body use="literal"/>
            </input>
            <output>
               <soap:body use="literal"/>
            </output>
      </operation>
   </binding>
</definitions>
```

Figure 2-5 An example for WSDL document [25]

Figure 2-5 is a simple WSDL document. In this example, the Web service is *glossaryTerms* and it has only one operation (function) called *getTerm* which involved one input parameter *getTermRequest* and one output parameter *getTermResponse*. The <binding> element shows that this Web service can use SOAP as communication protocol. The messages are conveyed by HTTP and the operation (function) is located at the following URL: http://example.com/getTerm .

With the detailed abstract information provided by WSDL documents, one Web service can be invoked by the other programs (Web services, agents or applications) regardless of the environment in which the calling party exists.

2.2 Semantic Web

This section briefly reviews the technologies related to the Semantic Web such as Ontology, Web Ontology Language (OWL), Web Ontology Language for Services (OWL-S) and the integration of OWL-S and UDDI.

2.2.1 Semantic Web and ontology

The Web was originally used for sharing information among scientists. At the beginning, this Document Web was used to store and share static information but it has gained increasing attentions due to the advance of Internet technologies and the enhancement of hardware capabilities. However, the large amount of data available on the Web can only be understood by humans or by very specialized programs. The goal of the Semantic Web [1],[13] is to make the data not only understandable by humans but also by machines.

Regarding the Web services, different primitive Web services within a composite Web service derive from diverse service providers and contribution is prone to the semantic misunderstandings. WSDL provides a machine-processable interface in which components that contribute to a composite Web service can be executed automatically. However, machine-processable is not identical to machine-understandable. WSDL provides less support for semantic description of Web services [9]. Even if one output parameter of one primitive Web service has the same name and type as an input parameter of another primitive Web service, it is not necessary that these parameters can be linked up to form a consistent composite service.

The Semantic Web is designed to increase the interoperability among machines by providing a machine-processable format with additional semantic information which can help the primitive Web services be searched and utilized autonomously hence increasing interoperability among Web services.

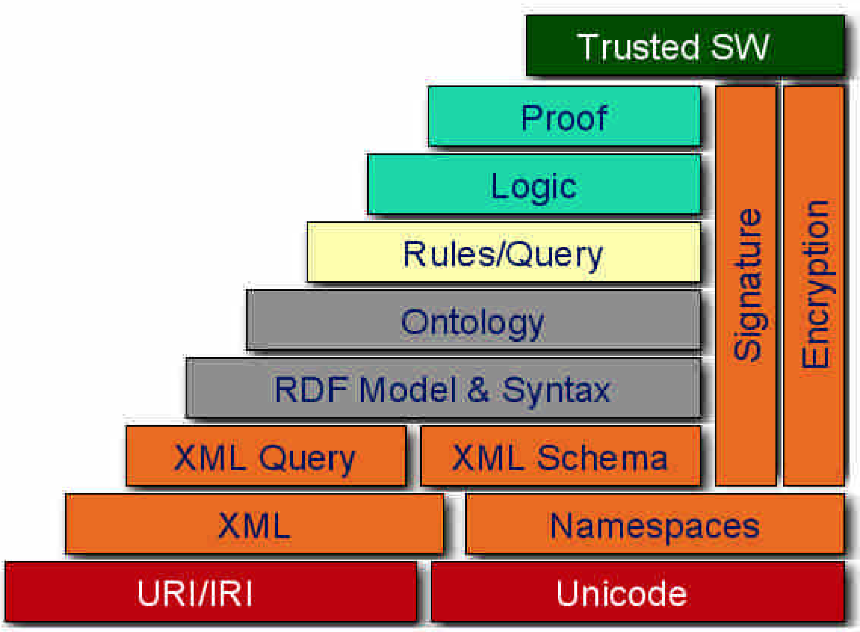

Figure 2-6 Semantic Web Stack [29]

The definition of ontology is: *"Ontology is a formal, explicit specification of a shared conceptualization"* [27]. The author of [28] introduced this concept into the domain of Artificial Intelligent (AI) by extending the meaning of ontology to *"Ontology as Vocabulary"*. The Semantic Web tries to create the *vocabulary* (ontology) for a specific domain and therefore different Web services in a heterogeneous environment can understand each other in this context. The stack for the Semantic Web is shown in Figure 2-6.

Several standards were introduced to achieve this goal such as Web Ontology Language (OWL) [14],[30],[31],[32] and Web Ontology Language for Services (OWL-S or formerly DAML-S) [15]. These standards help to handle the semantic issues occurring in Web service utilization.

2.2.2 Web Ontology Language (OWL)

The Semantic Web tries to build the ontologies for the Web in which information is given explicit meaning, making it easier for machines to automatically process and integrate information. This concept can be realized by the Web Ontology Language (OWL) [14]. It is a W3C recommendation language for defining Web Ontology.

OWL is based on XML, XML Schema, RDF and RDF Schema and it is derived from DARPA Agent Markup Language and the Ontology Inference Layer, in short DAML+OIL. In order to increase interoperability among machines, OWL is used to represent the meaning and semantics of terms on the Web explicitly and to describe explicitly the relationships between these terms.

Figure 2-7 shows three different species (sub-language) of OWL, OWL Full, OWL DL and OWL Lite, whose expressive abilities are decreased in order but the computational abilities are increased in reverse order. OWL Full is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. OWL DL (Description Logics) supports those who want the maximum expressiveness without losing computational completeness. OWL Lite supports users primarily needing a classification hierarchy with simple constraints. [14]

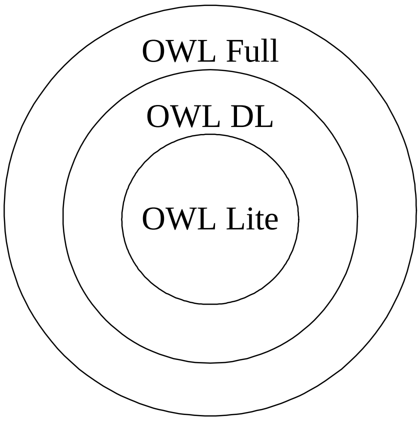

Figure 2-7 Three species for Web Ontology Language (OWL) [14]

In the simple example given in Figure 2-8, OWL is used to describe and define the semantics of terms. In this case, the term *cost* (line 11) is a subclass of *price* (line 13) and the term *value* (line 31) is also a subclass of *price*. Therefore, one communication counterpart can deduce and understand that *cost* can be related to *value* when the others talk about the notion *price* by using these different terms. Certainly, the power of OWL goes beyond this example. OWL can express notions and their complicated relationships, constraints, data types, and other metadata.

The W3C article [30] presents the abstract syntax of the Web Ontology Language (OWL) and the test cases for the OWL approved by the Web Ontology Working Group can be found in the document [32]. Tools for supporting OWL such as DAMLJessKB and OWLJessKB can be found from [33],[34] respectively.

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:p1="http://protege.stanford.edu/plugins/owl/protege#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://localhost:8080/tests/definition.owl#"
  xml:base="http://localhost:8080/tests/definition.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="cost">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="price"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="airperiod">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="airtime"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="arrtime">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="arrivaltime"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="stop">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="stops"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="value">
    <rdfs:subClassOf rdf:resource="#price"/>
  </owl:Class>
  <owl:Class rdf:ID="enroute">
    <rdfs:subClassOf rdf:resource="#stops"/>
  </owl:Class>
</rdf:RDF>
```

Figure 2-8 An example for Web Ontology Language (OWL)

2.2.3 Web Ontology Language for Services (OWL-S)

Web Ontology Language for Services (OWL-S), originally called DAML-S. It is an OWL-based ontology language which is used to describe the capabilities and properties of Web services in unambiguous and machine-understandable form. Figure 2-9 is the top level of the service ontology in OWL-S [15]. In OWL-S, a Web service will be represented by three essential types of knowledge which are as follows:

(1) ServiceProfile: Describe *"what the service does?"* Each Web service should present its own ServiceProfile to describe the capability, functionality and / or the contact information of a service. These messages are normally for human users.

(2) ServiceModel: Elaborate *"how the service works and how to use the service?"* This information contains the structure (workflow) of a service including input, output, precondition and effects, usually referred to as IOPE. If a Web service is composed of several *simple* (primitive) services, OWL-S will use an element called a *controlConstruct* to describe the workflow among *simple* services. These flow control constructs include *Sequence*, *Split*, *Split+Join*, *Unordered*, *Choice*, *If-Then-Else*, *Iterate*, and *Repeat-Until*. These messages are the main body of a OWL-S document.

(3) ServiceGrounding: Describe *"how to access this service?"* This message maps the abstract interface to concrete binding information by specifying the communication protocol, message formats, and other service-specific details such as port numbers.

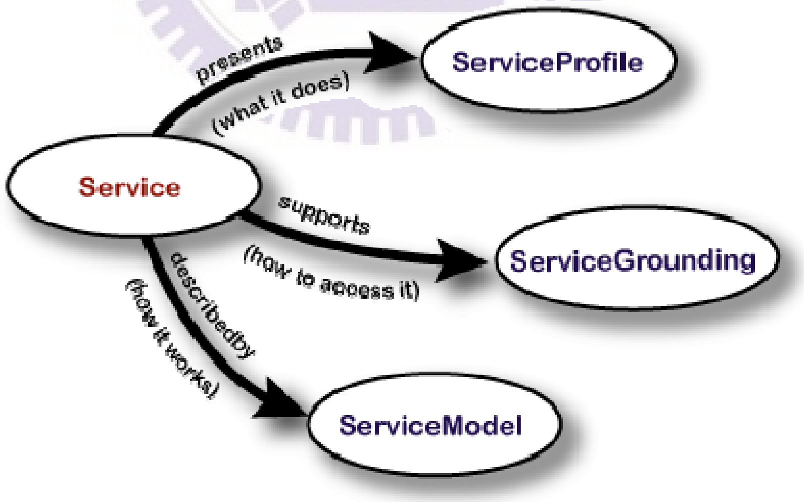

Figure 2-9 The components of OWL-S [15]

OWL-S can be regard as a specialization of OWL which is dedicated for the domain of Web service. An OWL-S document can therefore be understood not only by humans but also by machines. Briefly, ServiceProfile can be read and searched by humans; the ServiceModel explains the workflow that the machines should obey when any process is invoked via the detailed protocol and format provided by ServiceGrounding.

2.2.4 The mapping between UDDI and OWL-S

Web service providers adopt standard UDDI as a tool for advertising their Web services. However, the information represented in UDDI lacks semantic meaning, so it cannot fully support computers and people in cooperation. With the complementary support from Semantic Web technologies, the detail descriptions for a Web service can be modeled by OWL-S which is designed to handle the semantic issues associated with representing a Web service. Retaining a list of semantic meanings in UDDI provides a convenient way to support discovery of Web services, as the ServiceGrounding in OWL-S is able to locate WSDL documents and the associated Web services. With OWL-S, the descriptions of Web services can become machine-understandable concepts. Figure 2-10 shows the mappings between UDDI and OWL-S [35]. This enables UDDI and OWL-S to work seamlessly together for autonomous Web service discovery and execution.

The element *qualityRating* which resides in the ServiceProfile, shown in Figure 2-10, is the parameter used to record the higher level abstractions about the quality provided by a particular Web service. The proposed approach uses this element and its corresponding tModel to evaluate every Web services.

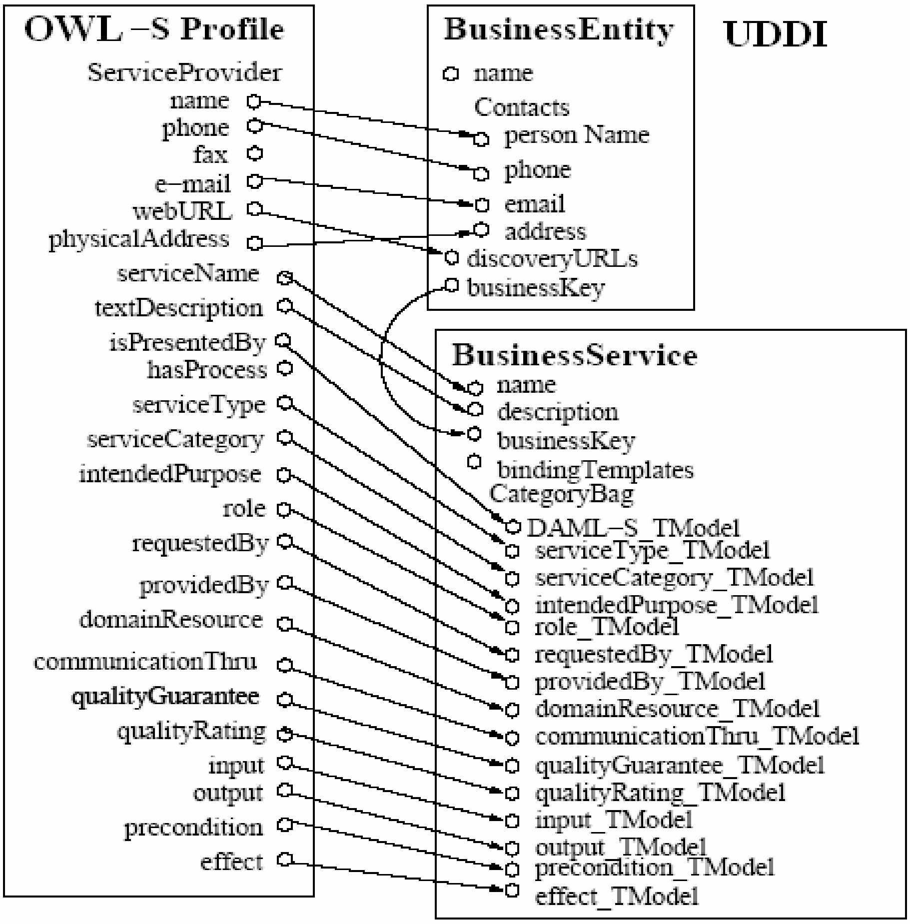

Figure 2-10 The mapping between OWL-S and UDDI specification [35]

Some elements such as *e-mail*, *serviceName* and *textDescription* are mapped directly from ServiceProfile to UDDI, whereas the OWL-S specific attributes such as *input*, *output*, and *qualityRating* are represented by the tModel structure. A detailed illustration of how to import the OWL-S profile file into the UDDI registry is discussed in [35]. This mapping brings the power of the Semantic Web into the UDDI registry. With this feature, information stored in the UDDI registry can not only be understandable by humans but also by machines.

2.3 Service composition

The technologies or standards related to service composition will be reviewed in this section such BPEL4WS, CSSL, BPWS4J, etc… In addition, two different runtime strategies (centralized and decentralized) will be introduced as well.

2.3.1 Service composition

Service composition is the construction process of composite services [8]. A composite service is a *complex* Web service which is composed of several *simple* (primitive) Web services. There are many research programmes endeavour in this field such as BPEL4WS [36],[37],[38], CSSL [9], WSIPL [41], WSCI [40], WSFL [39] and [8],[42],[43], etc. Some of these focus on the mechanisms of how to compose Web services while others are concerned with the semantic issues in composing a *complex* Web service.

The most frequently referenced language for Web service composition is the Business Process Execution Language for Web Services (BPEL4WS) [36],[37],[38] which has been developed jointly by IBM, BEA Systems, Microsoft, et al. BPEL4WS is a high-level distributed XML-based language which is used for assembling Web services to form a composite Web service. BPEL4WS enhances the Web services interaction model by supporting business transactions. The script language is used to describe the interactions among Web services by clarifying the control flow constructs (sequential, concurrent, conditional, etc.), the data structures and the activities (invoke, receive, compute, etc.) of a composite Web service as summarized in Table 2-4. The BPEL4WS runtime engine interprets BPEL4WS scripts to determine the execution process of a composite Web service.

Table 2-4 Summary of BPEL constructs and notation [44]

BPEL construct	Description	Notation
Control Flow Constructs		
sequence	sequential flow	sequence … end-sequence
switch	conditional flow	switch … end-switch
while	iterative flow	while … end-while
pick	non-deterministic flow	pick … end-pick
flow	concurrent flow	flow … end-flow.
link	wait-notify type of synchronization	source(linkId), target(linkId)
Data Structures		
variable	variables include a set of parts analogous to fields	variableName {part1, part2…}
Activities		
invoke[1]	synchronous (blocking) invocation on a partner *P*, sending data from an input variable *in* and receiving the response in the output variable *out*	invoke(*P*, *in*, *out*)
invoke[2]	asynchronous (oneway, nonblocking) invocation on a partner *P*, sending data using an input variable *in* (no response variable)	send(*P*, *in*)
receive	blocking receive of data from a partner *P* into a variable *var*	receive(*P*, *var*)
reply	send response to a partner P from a variable *var*	reply(*P*, var)
assign	assignment. Multiple assignments can be specified in a single assign statement, which executes atomically	var1.p1.g1 = var2.p1.g3
compute	arithmetic or logical operation	

The BPEL4WS flow constructs provide the ability for a developer to model the concurrent tasks within a composite Web service and to invoke them simultaneously. It also allows a service to wait for the response from other Web services. It can be treated as a workflow control language.

In other works [7],[8], the authors try to model semantic service requests for composite Web services by enhancing OWL-S with language construct extensions. These works help to achieve a uniform semantic representation of service requests before service composition. It also enables discovery agents to unambiguously understand the service.

The authors of [9] define a Composite Service Specification Language (CSSL) which is based on XML and is a WSDL-like language for composite services. This language extends WSDL to provide the semantic feature of Web services and defines the specification of the control flow between composite service operations. This work [9] proposed an ontology-based framework to accomplish service composition as shown in Figure 2-11. They model the ontology using directed graphs in which the nodes represent the ontology's concepts, the unfilled nodes refer to WSDL concepts, the gray nodes refer to extended features, and the edges represent relationships between the ontology's concepts labelled with the cardinality of the corresponding relationship.

In the research on service composition, most researchers focus on how to describe a composite Web service or model the process of service composition by assuming that all primitive services are ready-to-use or can be identified via simple UDDI queries. Therefore, service discovery and selection are not the main concerns in this field. The proposed approach of this dissertation, a consensus-based service discovery, tries to touch upon this question to assist the consumer in service discovery and selection during the process of service composition.

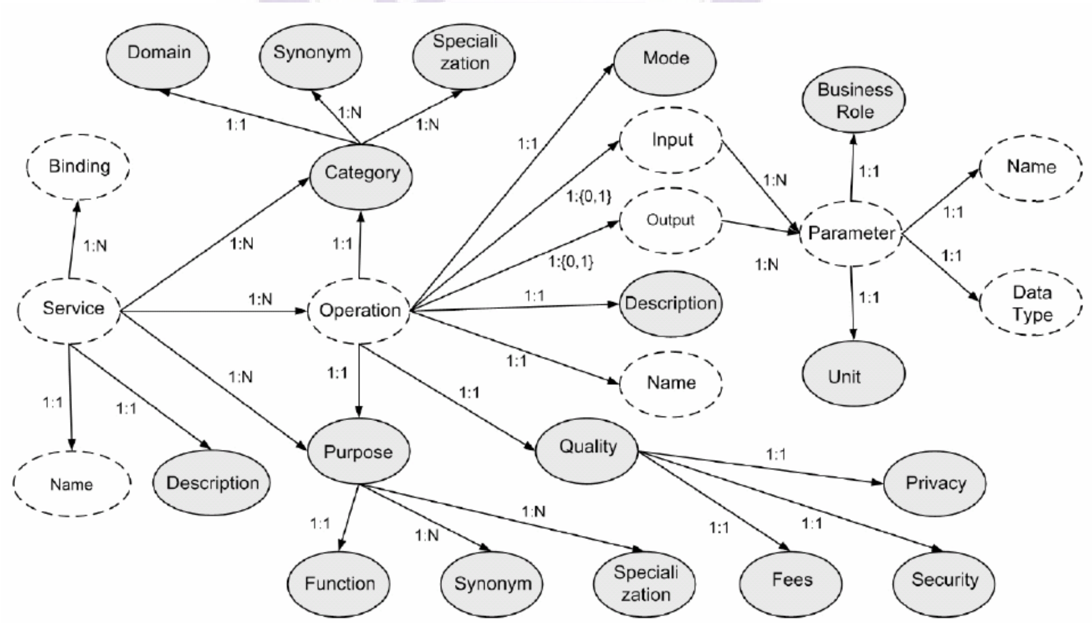

Figure 2-11 Ontology-based description of Web services [9]

The most interesting idea of the CSSL work [9] is that the authors define a composability model for Web services to determine whether two Web services are composable or not. This composability model uses two set of rules, syntactic rules and semantic rules, to estimate the composition possibility for two Web services. The syntactic rules concern (1) mode composability and (2) binding composability. The semantic rules consider (1) message composability, (2) operation semantics composability, (3) qualitative composability, and (4) composition soundness.

2.3.2 Centralized and decentralized orchestration

The script languages introduced in Section 2.3.1 are used to describe the processes of Web services. During runtime, the execution of a composite Web service is governed by a runtime engine such as Business Process Execution Language for Web Services JavaTM Run Time (BPWS4J) [45]. Once a *complex* Web service is composed, it can be executed by a BPWS4J engine as shown in Figure 2-12 [44].

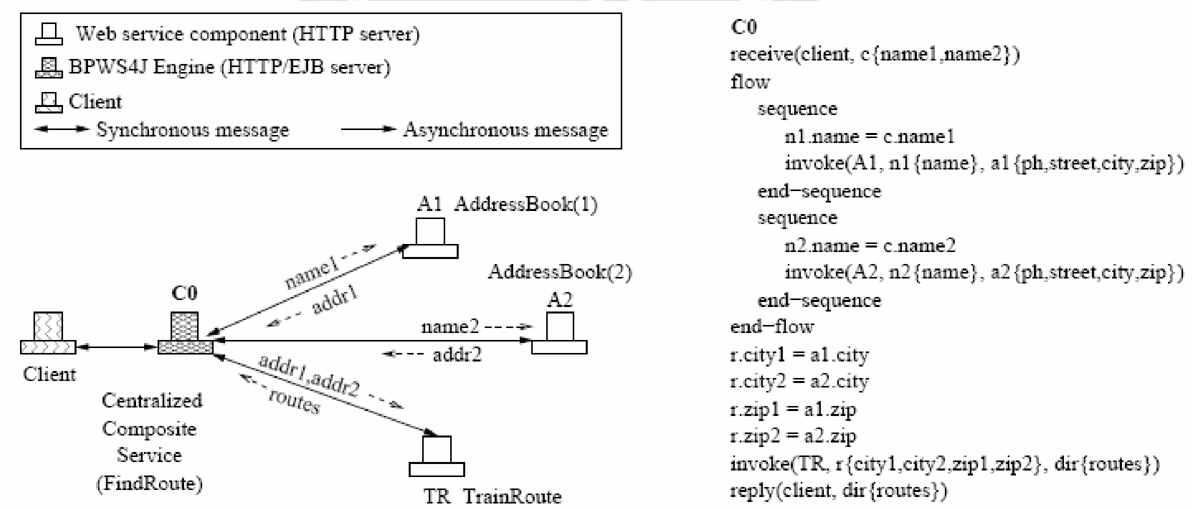

Figure 2-12 Centralized orchestration of a composite Web service [44]

In this example, a client has composed three primitive services *AddressBook(1)*, *AddressBook(2)* and *TrainRoute* into a composite service called *FindRoute*. *FindRoute* needs two names, *name1* and *name2*, from the client, then sends *name1* to *AddressBook(1)* and *name2* to *AddressBook(2)* for acquiring the *addr1* for *name1* and *addr2* for *name2* simultaneously. *FindRoute* extracts only the *city* and *zipcode* from the returned two addresses as input parameters for *TrainRoute*. *TrainRoute* will return the train route from *address1* to *address2*. This is so-called centralized orchestration or centralized execution.

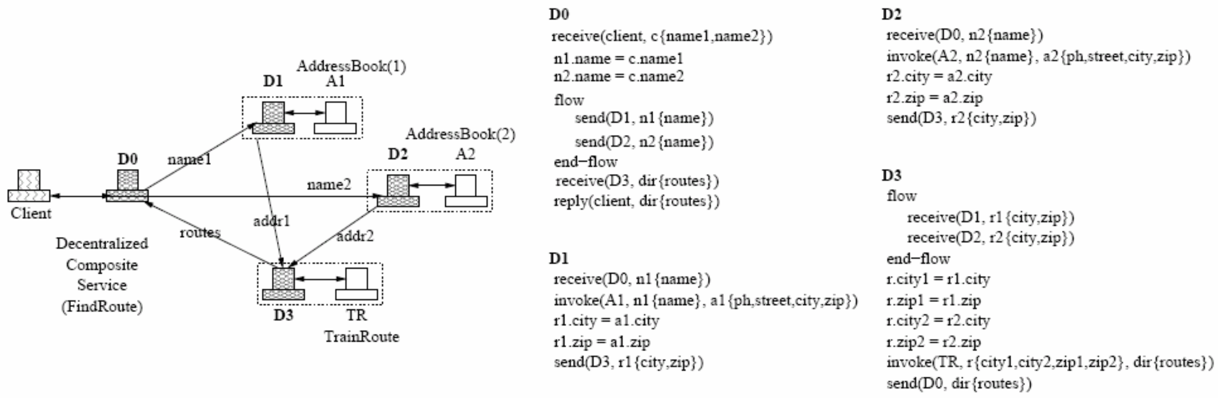

Figure 2-13 Decentralized orchestration of a composite Web service [44]

During the runtime, however, BPEL4WS, WSIPL, WSCI, executed by BPWS4J can be modified for decentralized orchestration as shown in Figure 2-13. A BPWS4J engine [45] is required to be installed in each of the distributed primitive Web services. The code for *FindRoute* will be divided and distributed to each of the corresponding BPWS4J engine (*D0*, *D1*, *D2*, and *D3*). *FindRoute* also receives *name1* and *name2* from client, and then send *name1* to *D1* and *name2* to *D2* in parallel for acquiring the address. However, *addr1* and *addr2* will not be forward back to the *D0*. They will be directly sent to the *D3* for carrying out the *TrainRoute* and only the result of *TrainRoute* will be forward to *D0*. This is known as decentralized orchestration. These research projects, [4],[44], try to simulate and analyze the performances of different orchestration from various points of view such as throughput and response time. They find the performance of decentralized orchestration is somewhat better then centralized orchestration but it raises some questions in relation to fault handling.

Again, researchers in this field focus on how to handle the data flow and pay their attention to fault propagation. They assume that all primitive services are easy to find and locate. Service discovery and selection are not the main concerns in this field.

2.4 Service discovery

This section will briefly review the research and developments related to Web service discovery including three models for Web service discovery and two types of approach to service discovery.

2.4.1 Three basic models for Web service discovery

Service discovery is a crucial process for Web service utilization. Three basic models for Web service discovery are identified: *matchmaking*, *broker* and *peer-to-peer (P2P)* as shown in Figure 2-14 [11],[46],[48]. The job for all discovery mechanisms are (a) take Service Requestors (consumers) description of the required Web service to interact with advertisements of Service Providers, (b) find the Web service(s) that closely fit the description, and (c) get a flexible matching which shows the relation between advertisement and requests [46].

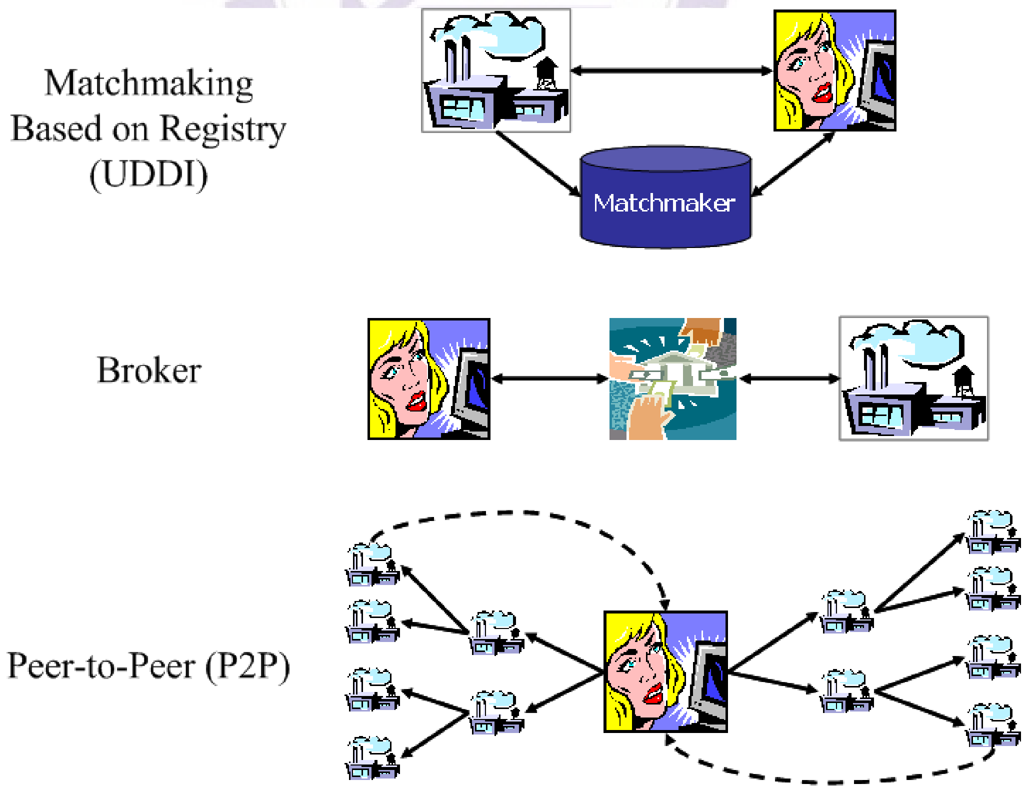

Figure 2-14 Three Models of Service Discovery [46]

The first mode, *matchmaking*, is usually based on a centralized registry in which service advertisements from various Service Providers are stored. When a Service Requestor (consumer) seeks a service, he (she) submits his (her) requirements to the matchmaker. The matchmaker compares the requirements with the descriptions in the advertisements (such as ServiceProfiles) to find the suitable service(s). Then, the registry responds with detailed descriptions of the found services (such as ServiceGroundings) to the Service Requestor. Following reception of an embedded description, Service Requestors can invoke a service. Relevant research can be found in [12],[16],[51].

The second mode, *broker*, is slightly different from the previous one. It performs both discovery and mediation for a client [46]. It also stores the advertisements of Web services submitted by Service Providers and compares requests from Service Requestors with the advertisements. If any suitable service is found, the *broker* acts as a proxy server by relaying the interactions (request-response) between a Service Requestor and a Service Provider. That is, a Service Requestor talks to the Service Provider indirectly. Similar research can be found in [49],[50].

There is no centralized registry in the third mode, *peer-to-peer (P2P)*. In this scenario, Service Requestors themselves find the Service Providers by message passing between peers. There is no matchmaker or broker. When a Service Requestor seeks a service, it broadcasts its requirements via a P2P network. Any Service Provider getting this request will compare its capabilities with the requirements and responds to the Service Requestor when its capabilities match the requirements. This mode is useful for ad-hoc networks and ubiquitous computing due to its dynamical nature. Relevant research can be found in [47],[52],[53].

2.4.2 Functional service discovery – capability discovery

The OASIS research on service discovery [5] is based on technologies of pattern matching and searching techniques that have been applied in the field of Very Large Databases (VLDB). The proposed process searches for the required object in a huge amount of data by the use of classified catalogues. For example, it compares the requested name, address, type of service or region information to the data stored in the registry, and returns the found bindingTemplate to the requestor if there is any match. The bindingTemplate indicates the URL of the found service. Using the URL, clients can download the WSDL description and then starts to interact with the service as shown in Figure 2-15. However, UDDI provides a keyword search of Web services but not of capability [46]. It is hard to find a specific airline booking service through this approach because the service is advertized by its function.

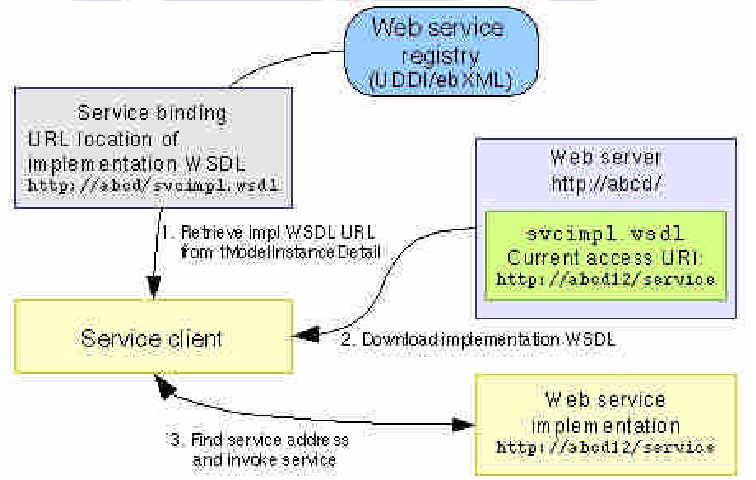

Figure 2-15 Seeking flows for a specific service via UDDI Registry

Functional service discovery means searching by functionalities which are provided by Service Providers. It is not only a simple pattern matching but also semantic searching. For example, Service Requestors may refer to ‘airline booking services’ to book a flight. The discovery service should return booking services for airlines and not include other booking services, for example, for football or concert tickets. It is known as capability discovery.

The work by a team at the Carnegie Mellon University (CMU), M. Paolucci et al. [15],[35],[46],[50], integrates the OWL-S matching engine into UDDI which enables capability search in a semantic context. It compares the Service Requestors' requirements with Service Providers' capabilities based on semantic descriptions. The approach proposed by this dissertation will also leverage their work to enable service discovery using semantic descriptions. This study proposes a consensus-based service discovery approach which attempts to use the underlying data and information about services as a search criterion (quality rating). With this feature, Service Requestors can discovery services by using linguistic terms (vague queries) such as *Cheap* or *Comfortable* during their search for a flight booking service for example.

In [54], the authors argue that search by the information in the ServiceProfile is not enough to find a service properly. The limitations arise due to the logical relationships among the inputs and outputs of a process. Assume that a simple process produces two outputs, *o1* and *o2*. If a request requires both of these outputs, it will result in a positive match when the search is simply based on the ServiceProfile. The authors of [54] develop algorithms which do not search using the ServiceProfile but use the ServiceModel to examine the detailed execution process of a Web service. They declared that analyzing the logical nature within the process would increase accuracy. However, the authors of [11] criticize this idea and claim that the ServiceModel is not primarily provided to express properties to be used for finding matches. Moreover, the use of underlying data referring to services as a search criterion still has not been addressed in [54].

Another important project in the field of service discovery is the Language for Advertisement and Request for Knowledge Sharing (LARKS) [16]. It emphasizes that matching should be based on other elements, NOT ONLY on keyword retrieval. The semantics of requests and advertisements should be taken into consideration. The discovery process of LARKS contains both syntactic and semantic concerns. Table 2-5 shows the frame structure of a LARKS specification used for service discovery.

Table 2-5 The frame structure of a LARKS specification [16]

Context	Context of specification
Types	Declaration of used variable types
Input	Declaration of input variables
Output	Declaration of output variables
InConstraints	Constraints on input variables
OutConstraints	Constraints on output variables
ConcDescriptions	Ontological descriptions of used words
TextDescription	Textual description of specification

An overview for matchmaking using LARKS is shown in Figure 2-16. The LARKS approach offers the option to use application domain knowledge in any advertisement or request using a local ontology. Briefly, LARKS provides discovery based on capability or functionality. This point of view is important to the researches thereafter. As mentioned before, however, the approach proposed by this dissertation is not only interested in the semantic issues but also the vague queries based on the quality rating of the underlying data about Web services.

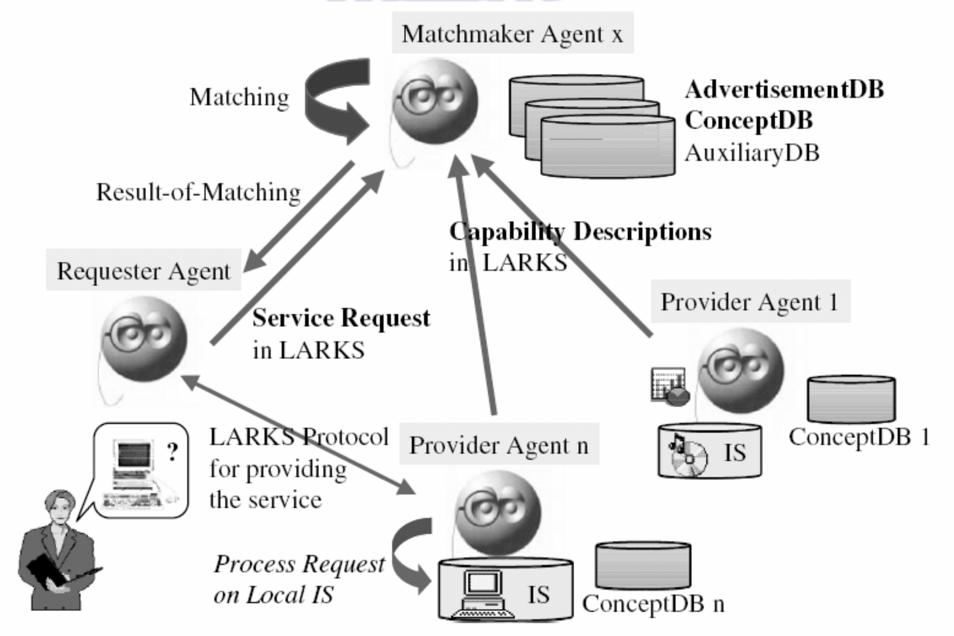

Figure 2-16 An overview for matchmaking using LARKS [16]

In [11] and in earlier work [48], an algorithm to rank the corresponding Web services according to OWL-S descriptions during the discovery process is proposed. In this algorithm, all possible Web services will be analyzed in four stages: (1) the matching of inputs, (2) the matching of outputs, (3) the matching of service category, and (4) the matching of user-defined criteria. Each Web service will be rated during these four stages and the results will be aggregated to become the final assessment as shown in Figure 2-17. According to the assessments, a rank for Web services can be made for further selection. The ranking idea has contributed to service discovery and is somewhat similar to what this dissertation proposes. The concept of quality rating is also adopted by this dissertation. In addition to the concept of quality rating, this dissertation also details a rating procedure, describing how to evaluate a service using consensus based opinion, which has not been fully explored in [11],[48].

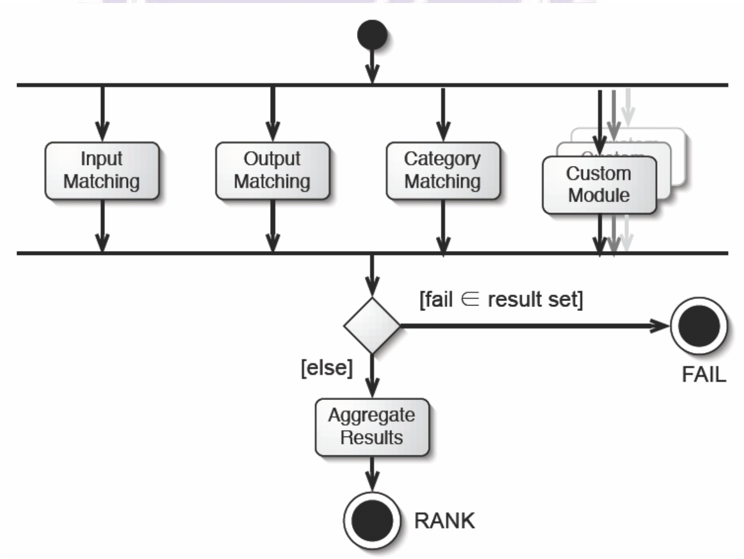

Figure 2-17 The rating procedures and ranking result for algorithm [11],[48]

Other work in functional discovery, such as [55], contribute their efforts in designing the detailed discovery algorithm for functional context reasoning within semantic environments.

However, all studies mentioned in this section are based on functional or capability search, but they have not paid sufficient attention to the use of underlying data and information on services as a search criterion. This study proposed a consensus-based service discovery approach which attempts to use the underlying data on services as a search criterion (quality rating) to refine the search space and to increase the precision rate of discovery.

2.4.3 Non-functional service discovery – QoS-aware discovery

Exactly matching services and those with similar functionalities will be discovered by the capability discovery mentioned in Section 2.4.2, hence identifying a number of possible services. It requires service consumers to include additional aspects (i.e. content of service) to evaluate these services. The purpose of non-functional service discovery is not only to find the services with the correct capabilities but also to find the *best* service which matches the other non-functional concerns such as *fees*, *security* or *availability*. This concept is referred to as QoS-aware discovery.

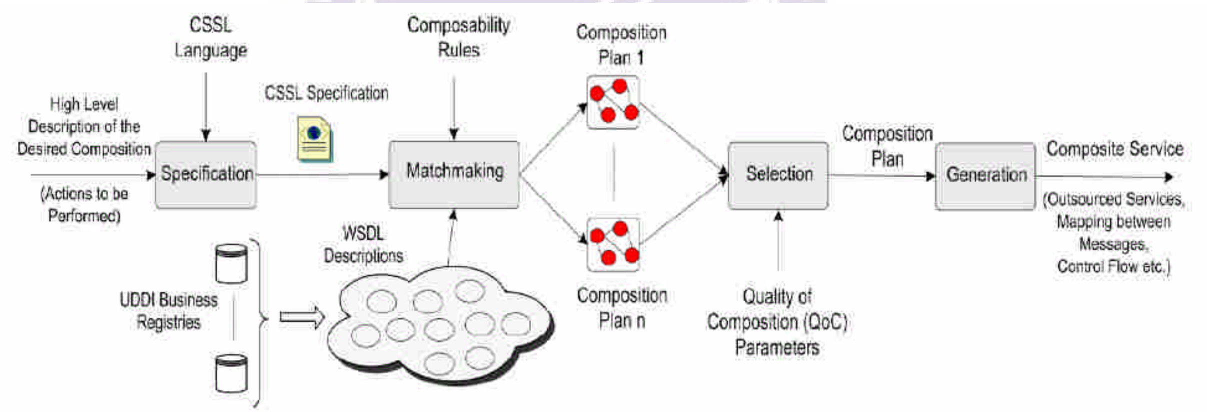

Figure 2-18 Overview of proposed approach by B. Medjhed, et al. [9]

In [9], the authors are not only concerned with whether the composition can be constructed, based on the capabilities, but also they are interested in the quality of the composition. They identify three qualitative properties for the composition: *fees*, *security* and *privacy*. Accompanied by the other common properties such as *time*, *availability* and *latency*, the quality of a composed operation can be calculated to compare with the other composition alternatives. The procedures of their approach are shown in Figure 2-18.

To include the properties such as *fees*, *security*, *privacy*, *time*, *availability* and *latency* as quality of composition for service selection process is an improvement [9]. However, these aspects are technical viewpoints and therefore can be extended by considering the underlying data on services as a selection criterion which is included in the approach proposed by this research.

The authors of [19] talk about the idea of Qos-driven selection for Web service composition. They observed that the selection among services with overlapping or identical capabilities needs consumers to pay additional attentions to the services' qualities. The criteria they found included *price*, *availability*, *reliability* and *reputation*. They provide a QoS-aware middleware for the selection of services which helps to maximize consumers' satisfaction. Table 2-6 defines the quality criteria and their aggregation functions which are used in the QoS-aware middleware.

Table 2-6 QoS criteria and their aggregation functions in [19]

Criteria	**Aggregation function**
Price	$q_{pr}(p) = \sum_{i=1}^{N} q_{pr}(s_i, op(t_i))$
Duration	$q_{du}(p) = CPA(p, q_{du})$
Reputation	$q_{rep}(p) = \frac{1}{N} \sum_{i=1}^{N} q_{rep}(s_i)$
Success rate	$q_{rat}(p) = \Pi_{i=1}^{N} (q_{rat}(s_i)^{z_i})$
Availability	$q_{av}(p) = \Pi_{i=1}^{N} (q_{av}(s_i)^{z_i})$

QoS based on *reputation* is significant and deeply influences the concept of the approach proposed in this dissertation. Nevertheless, a detailed discussion on the moderation of reputation rating is out of the scope of the study [19] but it falls in the scope of the approach proposed by this research. This dissertation details how to rate a Web service based on a group of consumers' opinions and thus achieve better satisfaction.

In the studies [58],[59],[60], the authors also observed that it is hard to find the most useful service based merely on the *explicit* matching of parameters provided by Service

Requestors and Service Providers. They assert that a consumer's preferences or implicit assumptions with respect to common knowledge in a certain domain should be taken into consideration to increase the satisfaction of a service provision. For example, if there is a consumer who lives in California looking for a Chinese restaurant, using only *explicit* capability matching, it is possible to find a restaurant located in Hong Kong which serves Chinese food, and the result is clearly unhelpful. There must be some *implicit* constraints to refine the search range. They call this personalization and incorporate this feature into UDDI / DAML-S registries to allow cooperative discovery and selection of Web services.

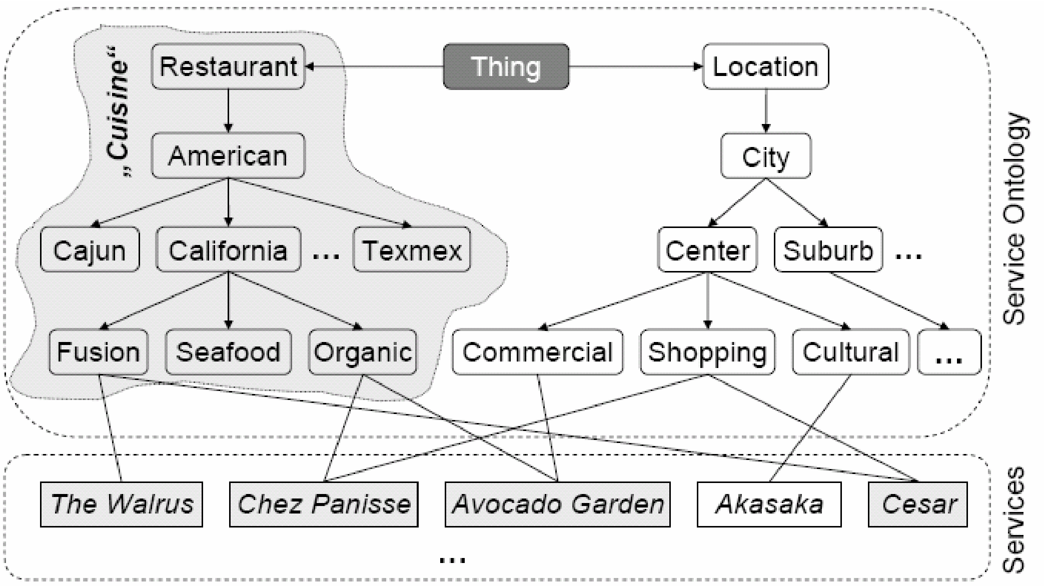

Figure 2-19 Service ontology for restaurant booking [60]

As shown in Figure 2-19 [60], if a consumer looks for a restaurant (*explicit condition*), a lot of services will be recommended. But if the individual's personalization settings (*implicit*) are considered, 'such as lives in the California city center', then only a few restaurants in California will be provided. If further constrained to the commercial district then only Avocado Garden will be a match. Consumers' interests and preferences are important factors in service discovery and selection. Personalization settings could help consumers to find the more suitable services which conform to their preferences. However, full personalization is very time consuming especially when search is based on the content of a service (e.g. the price level or quality of the meal). This dissertation tries to use a group consensus as a basic condition to rank the eligible services and to filter out those services which are not strongly related to the group consensus when search is based on the content of a service. Consumers' preference will be taken into account and opinions will be aggregated to form an objective group opinion for the use of rating services. With this feature services can be rated and classified by an objective opinion to obtain a better match.

Further, [56] reports on the comparison of different algorithms such as the naive algorithm, Fagin's algorithm, and the threshold algorithm. These algorithms aggregate information from various data sources. The aim is to retrieve the overall top-k objects from data resources. [57] presents an approach for answering imprecise queries in web-accessible databases. This approach is claimed to enable databases to support imprecise queries by identifying a set of related precise queries which return the results that are more relevant to the user's queries. This approach is somewhat relevant to our approach in relation to the idea of vague query. Nevertheless, the studies [56],[57] do not consider either the consensus aspects or Web services.

The paper, [17], reports on supporting linguistic search on Web services and applying quality rating to the content of a specific Web service. The authors of [17] tried to classify services by considering the underlying data on services. However, the criterion selected to do the classification has no soundness theory. The data or information on a service has been arbitrarily classified by a provider according to the selected threshold [17]. This may hinder its application since service consumers and providers may have inconsistent classifications for the descriptive terms. For instance, a consumer and a service provider may have different means to evaluate the quality of service content, because they may adopt different criteria or have different expectations. This dissertation proposes a method which iteratively ***moderates*** the inherent classification criterion based on a consensus of opinions.

This section has introduced a number of service discovery mechanisms based on the use of non-functional criteria to select appropriate services from a set of overlapping services which provide similar or identical functions. Some of them proposed useful ideas such as: ranking the services; QoS-driven methods, or, quality of composition. Some of them describe the algorithms for Web service discovery.

However, most of them do not address the issues associated with the impact of diverse preferences and subjective expectations of service consumers and providers which are generally used in searching or in advertising Web services. The service consumers and providers often have different views on the content of services. This study attempts to alleviate these differences by proposing a consensus-based service discovery approach to model subjective fuzzy opinions, and to assist service consumers and providers in reaching a common consensus so that the efficiency of service discovery can be increased. This method is not proposed to replace most of the literature in this section. It is complementary to the literature as it introduces another dimension (quality rating of underlying data) to Web service discovery.

CHAPTER 3 FUZZY AGGREGATION AND FUZZY PREFERENCE FOR GROUP CONSENSUS

This chapter is focused on studies of fuzzy set theory and the methodologies used for reaching a consensus, including fuzzy opinion representation, fuzzy majority, fuzzy similarity measurement, fuzzy aggregation, reaching consensus, resolution methods for group decision problem, and the methodologies used to collect imprecise preference. This literature will be reviewed in the following Sections 3.1 to Section 3.4.

The approach proposed by this research is named "an approach to consensus-based service discovery" which implies that the service discovery is based on a consensus. For reaching a consensus, a Moderated Fuzzy Discovery Method (MFDM) has been proposed which is the most important component in consensus-based service discovery and is based on the aforementioned fuzzy set related methodologies in the following sequence.

The MFDM comprises several parts: (1) Similarity Aggregation Method (SAM) [61],[62],[63],[64]: (2) Resolution Method for Group Decision Problems (RMGDP) [65],[66],[67],[68],[69], and (3) Pseudo-Order Preference Model (POPM) [71],[72]. These methods are processed in a sequence so that SAM is initiated first to gain a consensus on distinct opinions and preferences. RMGDP then obtains the group preference over different selection criteria. Finally, POPM will be introduced to calculate the exact preference relation when consumers' preference is hard to distinguish or the preference relations between two alternatives are imprecise.

3.1 Fuzzy set theory

Ordinary sets, or crisp sets, are defined with an all or nothing membership concept. The fuzzy set is not totally different from a crisp set as it is built upon the concept of the crisp set. However, the value of a membership within a fuzzy set is not just 0 or 1. The values can be smoothly spread between 0 and 1 [22]. It generalizes the notion of membership from the black-and-white binary classification in the crisp set into the one that allows a *partial* membership. When the value of membership is 0, it means complete non-membership. If the value of membership is 1, it represents a complete membership.

A fuzzy set could be defined in two ways: (1) by calculating membership values of those members in the set separately, or (2) by defining membership function mathematically. Generally, the former one is used when the set is composed of discrete members and the later one is used when the domain is a continuous variable. For example, a fuzzy set ($\tilde{A}$) can be defined through enumeration using the following expression [22]. Where the summation and addition operators refer to the union operation and the notation $\mu_A(x_i)/x_i$ refers to a element x_i with a membership degree $\mu_A(x_i)$. Those elements x_i whose membership value is zero are not represented.

$$\tilde{A} = \sum \mu_A(x_i)/x_i = \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \mu_A(x_3)/x_3 + \mu_A(x_{...})/x_{...}$$

A fuzzy set for continuous members is show as follows. Four common types for the membership function are illustrated in Section 3.1.1 to Section 3.3.4.

$$\tilde{A} = \int_x \mu_A(x)/x$$

3.1.1 Membership functions

There are various types of membership functions. In practice, the most commonly used are trapezoidal, Gaussian, S- and Z- membership functions which will be introduced in Section 3.1.1.1 to Section 3.1.1.4.

3.1.1.1 Trapezoidal membership function

A trapezoidal membership function can be specified by four parameters (par1, par2, par3, par4). For example, Figure 3-1 shows a trapezoidal membership function for a fuzzy set ($\tilde{A}$) with parameters (c-b, c, d, d+b).

$$\mu_{\tilde{A}}(a)=\begin{cases} \dfrac{x-(c-b)}{b}, & c-b\le a\le c \\ 1 & c\le a\le d \\ \dfrac{-x+(d+b)}{b} & d\le a\le d+b \\ 0, & a>d+b, a<c-b \end{cases}$$

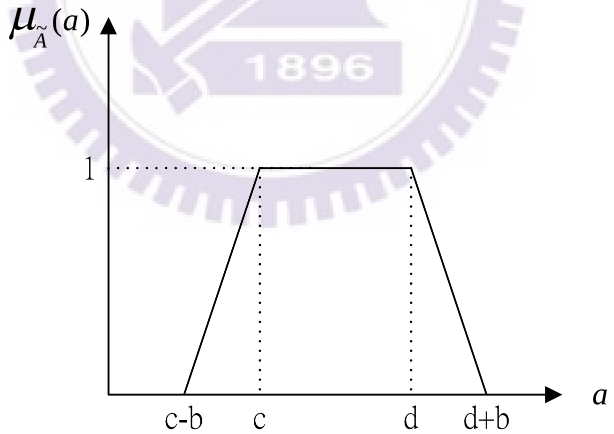

Figure 3-1 Trapezoidal membership function [22]

3.1.1.2 S-membership function

An S-membership function is a smooth membership function with three parameters (par1, par2, par3). For example, Figure 3-2 shows a S-membership function for a fuzzy set ($\tilde{A}$) with parameters (b, c, d).

$$\mu_{\tilde{A}}(a)=\begin{cases} 0, & a\le b \\ 2\left(\dfrac{a-b}{d-b}\right)^2 & b\le a\le c \\ 1-2\left(\dfrac{a-b}{d-b}\right)^2 & c\le a\le d \\ 1, & a\ge d \end{cases}$$

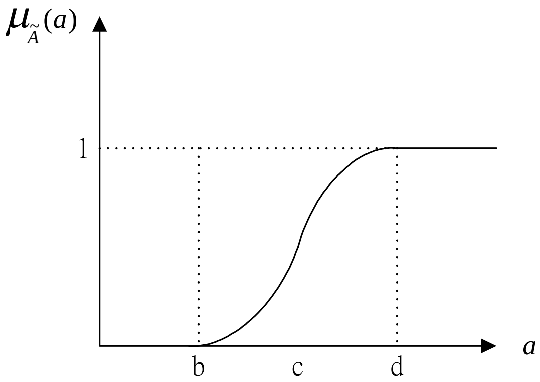

Figure 3-2 S-membership function [22]

3.1.1.3 Z-membership function

A Z-membership function is a smooth membership function with three parameters (par1, par2, par3). Figure 3-3 shows a Z-membership function for a fuzzy set ($\tilde{A}$) with parameters (b, c, d).

$$\mu_{\tilde{A}}(a)=\begin{cases} 0, & a\le b \\ 2\left(\dfrac{a-b}{d-b}\right)^2 & b\le a\le c \\ 1-2\left(\dfrac{a-b}{d-b}\right)^2 & c\le a\le d \\ 1, & a\ge d \end{cases}$$

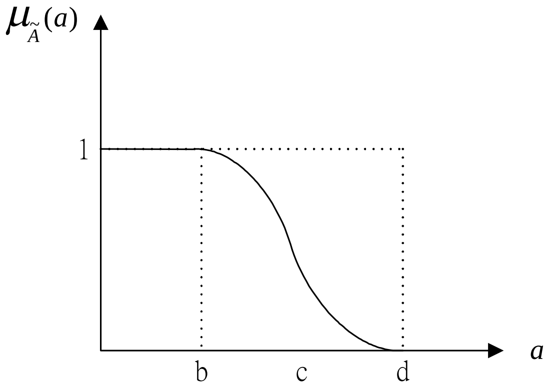

Figure 3-3 Z-membership function [22]

3.1.1.4 Gaussian membership function

A Gaussian membership function is specified by two parameters (m, σ) where m and σ denote the center and width of the function, respectively. The shape of the function can be controlled by adjusting the parameter σ. For example, a Gaussian membership function for a fuzzy set ($\tilde{A}$) with center c and width σ is shown in Figure 3-4.

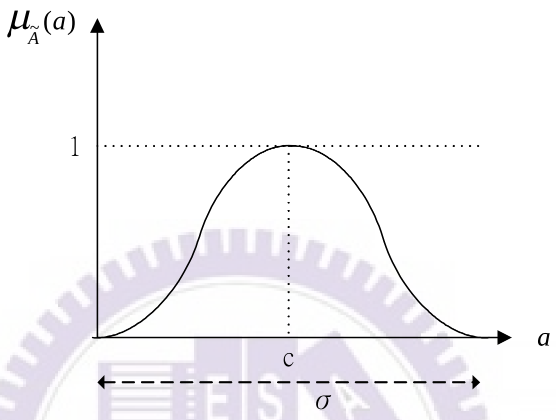

Figure 3-4 Gaussian Membership Function [22]

3.1.2 Basic operation

The fuzzy sets can be operated as crisp sets. Since membership of a fuzzy set is a matter of degree, the operation of fuzzy set should be defined accordingly. There are three basic operations: union, intersection and complement [73],[74].

3.1.2.1 Union

The union operation can be defined in various ways. The following example shows the definition that is used in most cases. The union of two fuzzy sets $\tilde{A}$ and $\tilde{B}$ with the membership functions $\mu_{\tilde{A}}(x)$ and $\mu_{\tilde{B}}(x)$ is a fuzzy set $\tilde{C}$, written as $\tilde{C} = \tilde{A} \cup \tilde{B}$, whose membership function is related to those of $\tilde{A}$ and $\tilde{B}$ as follows:

$$\mu_{\tilde{C}}(x) = \max[\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)], \forall x \in U$$

3.1.2.2 Intersection

Similar to union operation, there is no unique way to define the intersection operation. According to the *min-operator* the intersection of two fuzzy sets $\tilde{A}$ and $\tilde{B}$ with the membership functions $\mu_{\tilde{A}}(x)$ and $\mu_{\tilde{B}}(x)$, respectively, is a fuzzy set $\tilde{C}$, written as $\tilde{C} = \tilde{A} \cap \tilde{B}$, whose membership function is related to those of $\tilde{A}$ and $\tilde{B}$ as follows:

$$\mu_{\tilde{C}}(x) = \min[\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)], \forall x \in U$$

3.1.2.2 Complement

The complement of a fuzzy set ($\tilde{A}$), denoted as $\overline{\tilde{A}}$, is represented as the collection of all elements in the universe which are not included in the fuzzy set ($\tilde{A}$).

$$\mu_{\overline{\tilde{A}}}(x) = 1 - \mu_{\tilde{A}}(x), \forall x \in U$$

3.2 Similarity Aggregation Method (SAM)

The consensus formation technique, SAM [61], [62], [63], [64], is adopted to resolve different opinions about the terms used by service providers and consumers. SAM aggregates different users' fuzzy opinions to form a group's fuzzy consensus opinion. It employs a similarity measure to calculate the differences between individuals within the group in order to obtain an index of consensus. The indexes of consensus for all pairs of individuals can be used to form an agreed group fuzzy opinion. SAM ensures the consistency of the definitions of fuzzy terms for providers and consumers. It involves the following steps:

Step 1: Each user represents his / her subjective fuzzy preference on one specific criterion with a positive trapezoidal fuzzy number. A trapezoidal fuzzy number for a specific *USER*$_i$ can be represented by four parameters, denoted as $\tilde{Q}_i(a_i, b_i, c_i, d_i)$ where $a_i \le b_i \le c_i \le d_i$. $\mu_{\tilde{Q}_i}(x)$ is the membership function for a specific criterion $\tilde{Q}_i$ for *USER*$_i$ and the non-zero values of the user's subjective preference occur between $[a_i, d_i]$. If the value of x falls between $[b_i, c_i]$, *USER*$_i$ subjectively considers the truth value as 1; that is $\mu_{\tilde{Q}_i}(x) = 1$. This is shown in Figure 3-5.

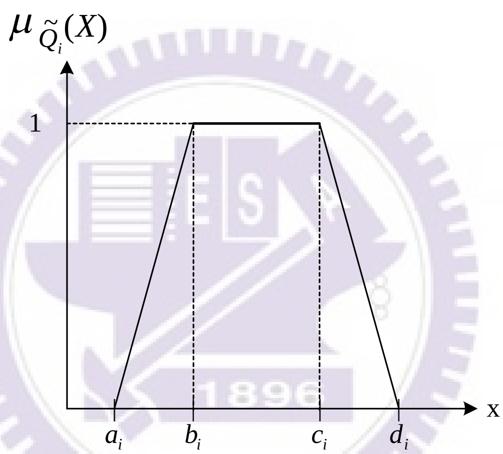

Figure 3-5 A trapezoidal fuzzy number

***Step 2*:** This step obtains opinion similarity between *USER*$_i$ and *USER*$_j$. The divergence between $\tilde{Q}_i(a_i, b_i, c_i, d_i)$ and $\tilde{Q}_j(a_j, b_j, c_j, d_j)$ can be calculated by the similarity measure function denoted as $S_{ij} = S(\tilde{Q}_i, \tilde{Q}_j)$.

$$S(\tilde{Q}_i, \tilde{Q}_j) = \frac{\int_x (\min\{\mu_{\tilde{Q}_i}(x), \mu_{\tilde{Q}_j}(x)\})dx}{\int_x (\max\{\mu_{\tilde{Q}_i}(x), \mu_{\tilde{Q}_j}(x)\})dx} \tag{1}$$

For example, when consider two different opinions on a specific criterion *Cheap*, ($\tilde{Q}$), for *USER*$_i$ and *USER*$_j$. Equation (1) can be transformed as follow:

$$S(\tilde{Q}_i,\tilde{Q}_j)=\frac{\int_x(\min\{\mu_{cheap_i}(x),\mu_{cheap_j}(x)\})dx}{\int_x(\max\{\mu_{cheap_i}(x),\mu_{cheap_j}(x)\})dx}$$

where $\mu_{cheap_i}(x)$ is $USER_i$'s membership function for *Cheap*, and $\mu_{cheap_j}(x)$ is $USER_j$'s membership function for *Cheap*.

Step 3: An agreement matrix, in Equation (2), can be formulated when the similarity between each pair in the group is obtained (where *n* is the number of users).

$$AM=\begin{bmatrix} 1 & S_{12} & \cdots & S_{1j} & \cdots & S_{1n} \\ S_{21} & 1 & \vdots & \vdots & \vdots & \vdots \\ \vdots & \cdots & 1 & \vdots & \vdots & \vdots \\ S_{i1} & \cdots & \cdots & S_{ij} & \vdots & S_{in} \\ \vdots & \cdots & \cdots & \cdots & 1 & \vdots \\ S_{n1} & S_{n2} & \cdots & S_{nj} & \cdots & 1 \end{bmatrix} \tag{2}$$

where $S_{ij}=S_{ji}=S(\tilde{Q}_i,\tilde{Q}_j)=S(\tilde{Q}_j,\tilde{Q}_i)$ and if $i=j$ then $S_{ij}=1$.

Step 4: This step calculates an average agreement degree of one single user.

$$A(USER_i)=\frac{1}{n-1}\sum_{\substack{j=1\\ i\neq j}}^{n}S_{ij} \tag{3}$$

Step 5: Relative Agreement Degree (*RAD*) for each user can be derived from the following formula.

$$RAD_i=\frac{A(USER_i)}{\sum_{i=1}^{n}A(USER_i)} \tag{4}$$

Step 6: This step defines the weightings, $w_i(i=1,2,...,n)$, for all the individuals' opinions. It could be equal weighting when any opinion is considered as important as the others.

Step 7: This step calculates individual Consensus Degree Coefficient (CDC) as follows.

$$CDC_i = \beta * w_i + (1-\beta) * RAD_i \quad \text{where } (0 \le \beta \le 1) \tag{5}$$

β is used for differentiating the importance between individuals' weightings and relative agreement degrees. In general case, $\beta = 0$; that is each individual's feedback is as importance as the others. In such case, it can deduce that Consensus Degree Coefficient (*CDC*) is equivalent to Relative Agreement Degree (*RAD*).

Step 8: According to the results derived from the previous step, each individual's opinion on the criterion can be gathered to form a group consensus opinion and produce $\tilde{Q}$ through the following formula.

$$\tilde{Q} = \sum_{i=1}^{n} (CDC_i \times \tilde{Q}_i) \tag{6}$$

This process, SAM, is applied to get the group consensus opinion on a specific criterion. If more than one criterion is considered, then SAM should be applied repeatedly to obtain a group consensus for each criterion. Once all the consensus opinions on the different criteria are obtained, the Resolution Methods for Group Decision Problem (RMGDP) can be initiated to reach a consensus on their preferences over their different selection criteria (alternatives).

3.3 Resolution Methods for Group Decision Problem (RMGDP)

The objective of RMGDP is to resolve group differences and to reach a group consensus on their preferences over different selection criteria [65],[66],[67],[68],[69],[75]. This method can be divided into the following three phases: (1) transformation phase, i.e., to transform the individuals' opinions on different selection criteria into preference values; (2) aggregation phase, i.e., to aggregate the individual preference values for obtaining the group preference using OWA (Ordered Weighted Averaging) operator [70], and (3) exploitation phase, i.e., to compute the ranking of the alternatives by group preference. These phases are detailed in Section 3.3.1 to Section 3.3.3.

3.3.1 The transformation phase

The first step of this phase is to form a collection of users into a group. Each user has to evaluate a list of alternative criteria, and then assign an ordering preference to the alternatives individually. The users allocate orderings based on their own preferences and subjective judgments. For example, there is a list of alternatives, $A = \{a_1, a_2, a_3\}$, and $User_k$ sorts these three alternatives according to his/her preference such as $A^k = \{a_2, a_3, a_1\}$ which means $User_k$ assigns 1st order to a_2, 2nd order to a_3 and 3rd order to a_1, that is, $User_k$ prefers the criterion a_2 to the other two criteria. This collection is called "Preference Ordering / PO" [68]. For a specific $User_k$, it can be reformulated as $O^k = \{o_1^k, o_2^k, .., o_m^k\}$, where m is the number of alternative and o_m^k means the order assigned to alternative a_m. In this example, $O^k = \{3,1,2\}$ and it denotes the preference ordering for $User_k$ who prefers a_2 to a_3 and a_1. Next, a transfer function is applied to convert those individual ordering of alternatives to a "Preference Relation" [66],[68], p_{ij}^k, which characterizes the ordering preference degree between alternative a_i and a_j expressed by user $User_k$ as follows:

$$p_{ij}^k = f(o_i^k, o_j^k) = \frac{1}{2}\left(1 + \frac{o_j^k}{m-1} - \frac{o_i^k}{m-1}\right) \qquad (7)$$

where p_{ij}^k is a preference relation which denotes that a user $User_k$ has a subjective ordering preference of the alternative a_i over alternative a_j and m is the number of alternatives. The transformation function, *f*, will satisfy that increase in o_j^k and decrease in

o_i^k increases the value of p_{ij}^k. This is due to the fact that the lower ordering number represents that the user prefers the alternative, and vice versa.

3.3.2 The aggregation phase

This phase computes the collective preference, p_{ij}^c. The value of p_{ij}^c is an aggregation of *n* users' ordering preferences, $\{p_{ij}^1, \ldots, p_{ij}^n\}$, by means of a fuzzy majority [67]. A fuzzy majority is obtained by combining the OWA (Ordered Weighted Averaging) operator [70] with a fuzzy quantifier. The merging of the OWA operator and the fuzzy quantifier *Q* specifies the collective ordering preference on each alternative as follow:

$$p_{ij}^c = F_Q(p_{ij}^1, \ldots, p_{ij}^n) = \sum_{i=1}^{n} w_i . b_i \tag{8}$$

where $w_i = Q(i/n) - Q((i-1)/n)$, and b_i is the *i*-th largest value in the collection $(\{p_{ij}^1, \ldots, p_{ij}^n\})$. F_Q is the OWA operator combining the fuzzy quantifier Q to aggregate the individual preference values and to obtain the collective ordering preference of all users.

3.3.3 The exploitation phase

The exploitation phase has as a consequence the identification of the priorities of alternatives of group preference. Two well-known fuzzy ranking methods are used in this phase which are: Quantifier Guided Non-Dominance Degree (*QGNDD*) and Quantifier Guided Dominance Degree (*QGDD*) [65].

3.3.3.1 Quantifier Guided Non-Dominance Degree (QGNDD)

The authors of [75] developed a method for fuzzy ranking by means of fuzzy preference relations. The method determines the relative preference degree of alternatives. The Non-Dominance Degree (*NDD*) of fuzzy ranking can be calculated for an individual preference relation, and is formulated as follows:

$$u_{NDD} = 1 - \max\{\, p_{ji}^{c} - p_{ij}^{c}, 0\} \tag{9}$$

From Equation (9), the membership function $\mu_{NDD}(a_i)$ can be interpreted as the degree to which a_i is not dominated by any other $a_j (j = 1,...,m, j \neq i)$, where *m* is the number of alternatives. The function $\mu_{NDD}(a_i)$ is able to find the highest ranking of alternatives. One criterion with highest value of *NDD* indicates that it is not dominated by the remaining criteria. For a linguistic quantifier Q (e.g. “most”), the *NDD* of the linguistic quantifier is denoted as Quantifier Guided Non-Dominance Degree (*QGNDD*) as:

$$QGNDD(a_i) = F_Q(1 - d_{ji}^{s}, j = 1...m, j \neq i) = \sum_{i=1}^{m} w_i . b_i \tag{10}$$

where $d_{ji}^{s} = \max\{p_{ji}^{c} - p_{ij}^{c}, 0\}$, $w_i = Q(i/m) - Q((i-1)/m)$, and b_i is the *i*-th largest value in the collection ($1 - d_{ji}^{s}, j = 1..m, j \neq i$). $QGNDD(a_i)$ specifies the degree which a_i is not dominated by a fuzzy majority of the remaining criteria [68].

It is recognized that the solution offered by Equation (10) is that the fuzzy majority of the remaining alternatives $a_j (j = 1,...,m)$ does not dominate the alternative a_i. All the ordering preferences on the alternatives can be calculated by the application of Equation (10) to prioritise their order.

3.3.3.2 Quantifier Guided Dominance Degree (QGDD)

QGNDD cannot discriminate between the ordering of preferences, when μ_{NDD} of numerous alternatives are Unfuzzy Nondominated (UND) solutions [75], i.e. $\mu_{NDD}(a_j) = 1$. For instance, UND occurs when $\mu(a_i) \geq 0.8$, which represents the "most" quantifier. In order to avoid simultaneous existences of UND solutions, the resulting fuzzy ordering needs to be validated by other fuzzy ranking methods, i.e. Quantifier Guided Dominance Degree (QGDD). According to [65], the Quantifier Guided Dominance Degree (QGDD) which is defined in Equation (11) can quantify the ordering preference dominance that a_i has over all the others where a_j (j=1,…,m) using the fuzzy majority concept. As a result, it is able to prioritize the final collective ordering preference. Therefore, QGDD is used to validate the fuzzy preference ordering of alternatives derived from Equation (11) as follows:

$$QGDD(a_i) = F_Q(p_{ij}^c, j = 1...m, i \neq j) \qquad (11)$$

where $F_Q(a_1, a_2, ..., a_m) = \sum_{i=1}^{m} w_i . b_i$, $w_i = Q(i/m) - Q((i-1)/m)$, and b_i is the *i*-th largest value in the collection ($a_1, a_2, ..., a_m$). If the "*UND*" solutions have occurred (more than one alternative has the *UDD* value is 1), then it will be better to make the final preference ranking of each alternative by applying the results of *QGDD.*

3.4 Pseudo-Order Preference Model (POPM)

In Section 3.3, it is assumed that user preferences between various criteria are collected by a popular method – "Preference Ordering / PO" [66]. PO can be used to gather the ordering between different criteria but PO cannot distinguish the imprecise favourite degree (or distance) between two adjacent criteria. It is assumed that preference ordering over the criteria is precise and the order for alternatives based on group preferences is complete. In other words, PO requires users to provide their preference over different criteria in precise sequences (complete order).

In some case, however, a user might depict that "*These two criteria are almost identical (indifferent) to me*" or "*I can not distinguish the importance between these two criteria*". In such case, PO is not applicable because the users do not have enough knowledge or information on unfamiliar criteria. So it is difficult to provide the complete orders for indifferent or indistinguishable alternatives. For example, some users find it difficult to distinguish the relative importance of *Cheap* and *Comfortable*. In this situation, another collection method, Pseudo-Order Preference Model (POPM) [71],[72], is introduced to collect users' preferences. This does not require the users to express their preference for alternatives in complete order. The importance of alternatives is collected pair by pair in the form of fuzzy preference relations.

Further, the POPM is also helpful in facilitating a group of users in finding the most important (top-N) alternatives, when numerous criteria exist. System complexity can be reduced by limiting the number of alternatives and the performance can then be increased. Once the top-N alternatives are produced, the RMGDP process, described in Section 3.3, can be adopted to resolve the quantified consensus weightings for these important alternatives. The steps for the POPM are detailed as following.

First, a number of users have to be formed as a group: $User_k$, ($k=1,\ldots,m$). Each user has to evaluate a set of alternatives $A = \{ a_i \mid i=1,\ldots, n\}$, and assign the relative importance to each pair of alternatives, $P(o_i^k, o_j^k)$, which denotes the value that the $User_k$ allocates to the ordering preference of alternative a_i over alternative a_j based on their own preferences and subjective judgments. $P(o_i^k, o_j^k)$ can be used in place of the p_{ij}^k used in Section 3.3.

There are three fundamental preference relations existing in the classical preference structure. These relations are: (1) Strict preference (**P**), (2) Weak preference (**Q**) and (3) Indifference (**I**), and they are applied to represent an imprecise preference relation, based on the richness of service information. **P**, **Q**, and **I** describe the imprecise ordering preference degree between alternative a_i and a_j expressed by $User_k$ as follows [71],[76]:

Strict preference relation: $$P(o_i^k, o_j^k) - P(o_j^k, o_i^k) > p \tag{12}$$

Weak preference relation: $$q < P(o_i^k, o_j^k) - P(o_j^k, o_i^k) \leq p \tag{13}$$

Indifference relation: $$|P(o_i^k, o_j^k) - P(o_j^k, o_i^k)| \leq q \tag{14}$$

where the preference threshold *p* and indifference threshold *q* are defined to distinguish strict preference, weak preference, and indifference relations. When the difference between o_i^k and o_j^k exceed *p*, it indicates that $User_k$ strictly prefers o_i^k to o_j^k. Furthermore, if the difference between o_i^k and o_j^k is smaller than *q*, it means that o_i^k and o_j^k are regarded as no major difference between them.

The POPM follows the classical preference structure, which is described above, and has a special case called Semi-Order Preference Model (SOPM) that is adopted to reach the consensus among a group of users. The SOPM is applied only when user' preferences remain imprecise, uncertain or ambiguous or there are too many alternatives. The SOPM could help to identify the most important alternative (top-N) in numerous criteria and filter out those that are insignificant.

The SOPM, the special case when $p = 0, q \neq 0$, is applied to gain the nondominance set of alternatives when the relative importance of each user is predictable [76]. In such cases, the weak preference relation is neglected, and only the indifference threshold is employed to discriminate between the preference and the indifference relation. The relations between two alternatives for a specific $User_k$ are shown as follows:

$\forall a_i$ and $a_j \in A$,

Preference relation: $$P(o_i^k, o_j^k) - P(o_j^k, o_i^k) > q \quad (15)$$

Indifference relation: $$|P(o_i^k, o_j^k) - P(o_j^k, o_i^k)| \leq q \quad (16)$$

where indifference threshold q are defined to distinguish the preference degree between a_i and a_j.

According to the results derived from Equations (15) and (16), the collective preference (p_{ij}^c) for the group of user, $User_k$, (*k=1,..,m*), can be aggregated by the weighted sum of $P(o_i^k, o_j^k)$, as shown in Equation (17).

$$P_{ij}^c = \sum_{k=1}^{m} w_k \cdot P(o_i^k, o_j^k), \quad \text{where} \quad \sum_{k=1}^{m} w_k = 1, \quad (17)$$

Since it is difficult to reach a full consensus while aggregating, soft-consensus [65] is adopted for determining the group preference. The weighting vector w_k can be computed by the OWA operator [70].

After the computation of p_{ij}^c, the Outranking and Incomparability relations for the group of users, $User_k$, (*k=1,..,m*), can be determined by the following Equations [72]:

Outranking relation: $$P_{ij}^{c} - P_{ji}^{c} > q \tag{18}$$

Incomparability relation: $$| P_{ij}^{c} - P_{ji}^{c} | \leq q \tag{19}$$

According to Outranking relation and Incomparability relation, the consensual preference order of alternatives can be identified. It is based on the relative importance of criteria for service discovery, which can assist in screening insignificant criteria in the evaluation process.

Once the top-N alternatives are obtained, the RMGDP process, described in Section 3.3, can be adopted in turn to gather the quantified consensus weightings for these important alternatives. It is important that the steps for POPM / SOPM can be skipped when the users are confident with their preferences and the number of alternatives is appropriate.

CHAPTER 4 THE PROPOSED ARCHITECTURE – AN APPROACH TO CONSENSUS-BASED SERVICE DISCOVERY

4.1 Elaboration for the architecture and the key components

Web service technologies contain a set of standardized languages for describing interfaces and communication protocols which increase software interoperability. Service consumers or software developers can construct new services by composing existing services over the Internet. The provision of service discovery is a step towards semi-automatic or automatic Web service composition. Traditional information systems that may include database systems (data repository) can be wrapped by Web service technologies to become a service. In this case, the information associated with the interfaces and capabilities of the service may not be sufficient for consumers to locate their required services, as they have more interest in the contents within data repository. In addition, the users or systems may use vague requests, so fuzzy terms may be included in the query.

Such situations require extra descriptions for the data. Having a higher level of abstraction to describe the resident information or data could facilitate the service consumers in identifying their required services. The proposed architecture aims to represent the underlying data of Web services abstractly using fuzzy logic and semantic web technologies in order to optimize the discovery process. It also allows service consumers to employ imprecise terms in queries used to discover appropriate services. The architecture and key components will be detailed in the following sections.

4.1.1 General description and basic scenario

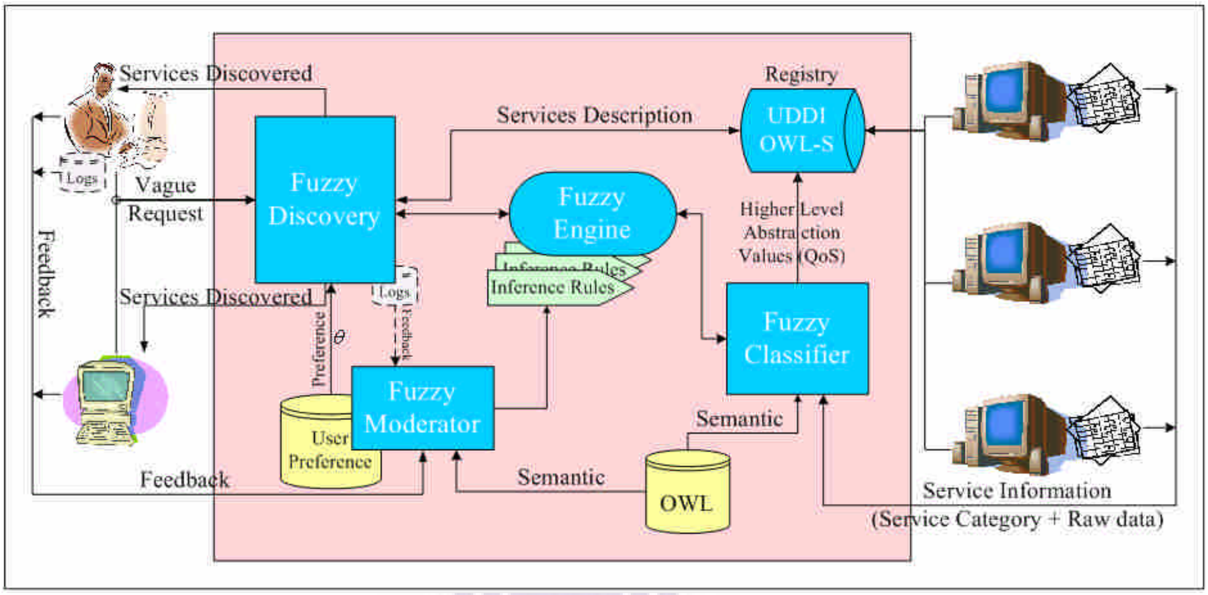

Figure 4-1 The proposed architecture for a consensus-based service discovery

The proposed architecture, as shown in Figure 4-1, it comprises a number of components, including *Fuzzy Classifier*, *Fuzzy Engine*, *UDDI / OWL-S Registry*, a *Fuzzy Discovery*, and a *Fuzzy Moderator*. Furthermore, two behaviours are identified namely the Service Providers (the right side) and Service Consumers (the left side). The basic scenario for this architecture is divided into 6 steps which are illustrated as follows:

Step 1: Various Service Providers prepare the advertisements for their services and publish these advertisements to an *UDDI / OWL-S Registry* for further inquiry.

Step 2: For each service, *Fuzzy Classifier* will examine its service category and the raw data for the purpose of forming a higher level abstraction of the underlying data provided by a service. A higher level abstraction is “one” kind of quality rating for a specific service (or QoS), such as quality rating for *Cheap*, and will become part of the information advertised in the *Registry*. This step is called pre-classification and is based on the inference rules preset in the *Fuzzy Engine*.

Step 3: Any Service Consumer (e.g. a consumer in Taiwan) expresses his / her needs to *Fuzzy Discovery* for finding services. For example, a consumer may state *"I need a cheap flight to UK"*. This request is vague and contains a quality rating (QoS) description for the desired services. *Fuzzy Discovery* will search the advertisements stored in the *Registry* based on the higher level abstractions – *Cheap* and the basic capability – a flight from Taiwan to UK. All requests from Service Consumers are processed individually.

Step 4: After filtering, *Fuzzy Discovery* returns the appropriate services which fulfil both the capability and quality requirements, where the capability information is provided by Service Providers in Step 1 and quality ratings of the service are pre-classified in Step 2.

Step 5: Service Consumers or their agents provide their feedback about the discovery and / or express their expectation of the specific term for quality rating, e.g. *Cheap*. Although requests are processed individually, these user preferences will be accumulated in *Fuzzy Moderator*.

Step 6: *Fuzzy Moderator* will be activated to reach a consensus based on the accumulated user preferences, for example, forming a consensus on the term *Cheap*. This would lead to the modification to the inference rules and trigger another classification in step 2 with the new consensus-based criterion *Cheap*.

The reasons to activate the *Fuzzy Moderator* will be discussed in Section 4.3.3. The detail operations of *UDDI / OWL-S Registry*, *Fuzzy Classifier*, *Fuzzy Engine*, *Fuzzy Discovery*, and *Fuzzy Moderator* will be elaborated in the following sections.

4.1.2 UDDI / OWL-S Registry

The proposed framework adopts standard UDDI as a tool for advertising Web services. However, the information represented in UDDI lacks well-defined meaning, so it cannot fully support computers and people to work in cooperation. With the complementary support from Semantic Web technologies, the descriptions in UDDI can be modelled in OWL-S.

Service Providers use OWL-S descriptions such as ServiceProfile, ServiceModel and ServiceGrounding to describe their services (but not including the values of quality ratings). Therefore, these become parts of the ontology in the OWL database. The OWL database is regarded as a data dictionary which resolves the different representations of one concept. The OWL database provides the ability to handle the semantic issues for several components, e.g. *Fuzzy Moderator*, *Fuzzy Classifier* and *Fuzzy Discovery*.

Retaining a list of Semantic Webs in UDDI provides a convenient way to discover Web services, as the ServiceGrounding in OWL-S is able to locate WSDL documents and the associated Web services. The description of services can be machine-understandable concepts. Figure 2-10 shows the mappings between UDDI and OWL-S [15]. This enables UDDI and OWL-S to work seamlessly together for the autonomous Web service discovery and execution.

This work adopts the general UDDI and OWL-S standards as tools to solve the semantic issues. The performance and deployment issues for a UDDI Registry, however, are out of the scope of this work.

4.1.3 Fuzzy Classifier and Fuzzy Engine

Fuzzy Classifier contains essential predefined knowledge for interpreting and classifying (rating) the information residing in Web services. It consists of primitive and composite fuzzy terms, modifier and quantification fuzzy terms, and fuzzy rules (i.e., inference rules for the *Fuzzy Classifier*). Primitive terms are a set of atomic terms that represent a collection of the raw data. A primitive term is derived from a fuzzy set $\tilde{A}$ which is defined as

$\tilde{A} = \int_x \mu_A(x)/x$, where x is the actual value, $\mu_A(x)$ is a member function and the value domain is located between 0 and 1.

Composite terms are generated through the combination of primitive terms and fuzzy rules. Because there is no specific QoS model provided in this case, it is assumed that the ingredients of a composite term are independent to each other. Composite terms can also be represented in fuzzy rules, whenever heuristic associations between terms are required. The quantification terms are also used to model the probabilities of occurrences. A statement can be altered by a modifier thereby making the statement a little more imprecise. In other words, the statements associated with quantification and modifier terms are represented in fuzzy rules for the purpose of reasoning.

The *Fuzzy Classifier* extends the aforementioned rules and their combinations to provide powerful classifications on the data resident in services in order to produce higher level informative declarations (quality rating or QoS value for Web services). After classification, each service will be rated by a QoS value as a higher level abstraction of one specific concept. This value will be inserted into the *Registry* as a part of the advertisement which was not available in the original OWL-S descriptions. If it is required, each service stored in the *Registry* can be rated from various perspectives, which means different QoS values will be produced and inserted into the registry.

The fuzzy rules (inference rules) are stored in the *Fuzzy Engine* which drives the *Fuzzy Classifier* to carry out the classification and evaluate the values of QoS for Web services. The example shown in Figure 4-2 illustrates that a service, SS, providing only a widget with price $100 will be rated to 0.8 when the specific inference rule, *Cheap*, is applied.

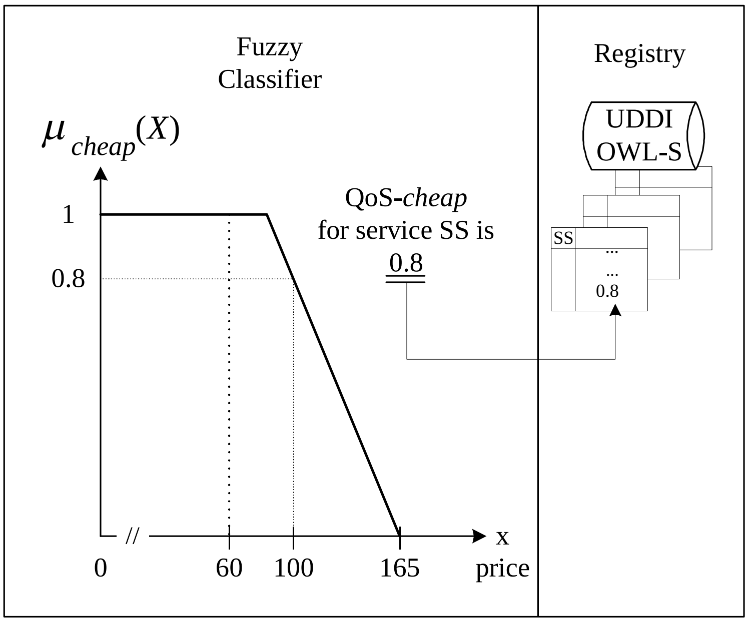

Figure 4-2　An example for the classification process

4.1.4 Fuzzy Discovery

Service Consumers express their needs to *Fuzzy Discovery* for finding services. For example, one consumer (in Taiwan) may state *"I need a cheap flight to UK"*. This request is vague and contains a quality rating (QoS) description for the desired services. *Fuzzy Discovery* will search the advertisements stored in the *Registry* based on a higher level abstraction – *Cheap*, the basic capability – a flight to UK and the context – departing from Taiwan. All Web services compliant to these capability requirements will be selected from the *Registry* and held in the *Fuzzy Discovery* for further filtering. Owing to the constraint, *Cheap*, those Web services whose quality rating value for *Cheap* is considered not cheap will be ruled out. The filtering criterion is based on the threshold θ. The value of θ is dynamic and is determined by the default setting in *Fuzzy Discovery* or by consumers' personalization settings.

It is not mandatory to have vague inquiries. Sometimes, Service Consumers might place the standing orders precisely, such as *"I need a flight to UK. Budget is 100"*. In such case, *Fuzzy Discovery* provides a function that can convert crisp requests from Service

Consumers into fuzzy requests. In this situation, this request, *"Budget is 100"*, will be transformed into fuzzy term – *"QoS-Cheap exceeds 0.8"* according to the inference rules stored in *Fuzzy Engine* as shown in Figure 4-2. It is important to have crisp terms transformed into fuzzy terms for the use of approximate reasoning, as the higher level informative declarations (quality rating descriptions) for services have been represented in fuzzy terms.

The use of *Fuzzy Discovery* enables the architecture to discover the required Web services in a way that allows Service Consumers to use vague queries and filter out those that are insignificant as based on the quality rating about the underlying data about Web services. Other linguistic vague inquiry methodologies could be applied into *Fuzzy Discovery*, such as finding a *'most' Cheap* or a *'very' Cheap* flight, but these enhancements go beyond the scope of this dissertation. The further descriptions on the linguistic search method, such as Possibility Relational Universal Fuzzy (PRUF), can be found in articles [17],[22].

4.1.5 Fuzzy Moderator

At the initial stage, the arbitrary fuzzy rules are applied to classify each of the Web services and produce a higher level abstraction about the underlying data provided by the service. However, the initial fuzzy rules may not be objective so the query results might not conform to consumers' opinions. It is important to moderate the rules according to service consumers' feedback. *Fuzzy Moderator* implements a moderation method called Moderated Fuzzy Discovery Method (MFDM) which bridges the gap between the expectations and preferences of Service Providers and Service Consumers. This is the key feature of the proposed architecture.

Fuzzy Moderator has the ability to keep track of the service consumers' feedback after they evaluate the result of each vague query request and it is also capable of reaching one common consensus opinion from those subjective feedback opinions. *Fuzzy Moderator* is able to incorporate iteratively users' subjective opinions and preferences, and transform them to less subjective ones. In principle, the more feedback gathered from users, the less subjective the consensus is. This is due to the generalization of their opinions and expectations. The feedback will be accumulated and calculated in the *Fuzzy Moderator* for further *'moderation'* use. It is assumed that the feedback collected in this study is gathered by questionnaires but questionnaire is not the only way (or the most efficient way) to collect user feedback. There are some manners applied in the field of data mining can be used to automatically collect feedback from the Web logs. However, these studies go beyond the scope of this dissertation.

This mechanism assists the service consumers and providers in reaching consensus on using the fuzzy terms and the preferences over the selection criteria. It is assumed that Web services consumers and providers possess different opinions and preferences on the required services. The moderation mechanism ensures consensus by taking into account those opinions and preferences which are accepted by the majority of service providers and consumers.

The proposed of moderation is to modify the fuzzy rules. After the consensus has been reached, the initial inference rules (fuzzy rules) can be moderated with less subjective opinions. Therefore *Fuzzy Classifier* will be triggered to do another classification with new consensus-based rules and new quality ratings for each service will be produced. Consequently consumers are expected to have a greater level of satisfaction with the discovery results, as the gaps between the consumers' and providers' expectations have been reduced.

The way of reaching consensus over their expected services among service providers and consumers can be considered as a problem of aggregation of a number of opinions for group decision making. In this case, the problem is complicated by the introduction of the fuzzy opinions by the consumers and providers. The detailed explanation about how to reach a consensus was provided in Chapter 3 and will be illustrated by case studies in the future.

4.2 System range and constraints

The proposed architecture is illustrated in the context of Semantic Web services throughout the dissertation. It is to serve the purpose of fully demonstrating the procedures of the proposed approach by case studies. But the proposed approaches, a consensus-based service discovery, and the approach for reaching a consensus, Moderated Fuzzy Discovery Method (MFDM), are not limited in the field of Web service discovery. It can be applied in any specific domain where service discovery is made based on the independent feedback which represent the quality rating of the underlying content (QoS or reputation), and gaps exist between the expectations and preferences of service providers and consumers. It is not suitable to have providers setting their own QoS or reputation values subjectively. The proposed approaches are helpful to form the values objectively based on the consensus, and it can be iteratively applied for reaching a consensus to mitigate these gaps.

However, this architecture has its own constraints. Firstly, the proposed method is based on pre-classification to evaluate a service. Any service that is initially entering into the registry can only be searched by its capability and cannot be discovered by the value of the quality rating. The value of quality rating may not be up-to-date but this problem can be alleviated by shortening the pre-classification interval. However, dynamical classification for each service is not supported in order to get better performance and avoid the superfluous classification. Secondly, this architecture is based a conceptually centralized registry as used in the *matchmaking* or *broker* systems described in Section 2.4.1. P2P is not supported because it is unreasonable to have a provider evaluating his own service objectively.

4.3 Implementation considerations

This section will describe the considerations for implementation such as implementation suggestions, handling the outliers and the system sensitivity. The tools used to implement the prototype will be listed for reference and the OWL ontology designed in the prototype system will be described.

4.3.1 Implementation suggestions

The hardware and software used to implement the prototype system are listed in Table 4-1.

Table 4-1 Implementation suggestions

Hardware	CPU	AMD 2500+
	L2 cache	512K
	RAM	768MB DDR-II
Operation System		Windows XP Professional with Service Pack 2
Expert System		Java Expert Shell System (JESS) [78],[79]
OWL Editor		Protégé 3-1-Beta [77]
OWL Parser		OWLJessKB [34]
UDDI Server		jUDDI v0.9rc4 [80]
HTTP Server		Tomcat 5.5 [81]
Database System		MySQL 5.0 [82] or textfile
Main Program Language		J2SDK 1.5 [83]
Group Decision Making		Mathematica 5.0 [84], Matlab [85]

Figure 4-3 shows an ontology example, OWL expressed in which will be used in the case studies. Different service providers may use terms, such as *Stop*, *stops* or *enroute*, to represent the number of landings (or departures) during a given flight in their proprietary systems. However, these terms are considered to be the same within the ontology. Moreover, *cost*, *fare* and *value* are considered as *Price*.

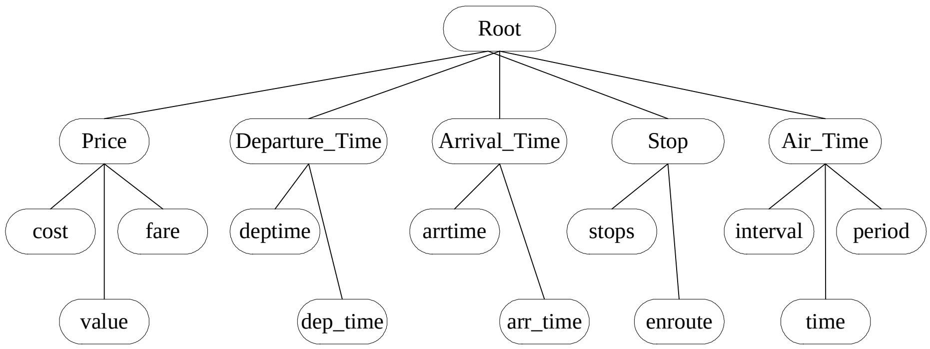

Figure 4-3 An example of OWL definition

4.3.2 Handling the outliers

The outliers are the feedback messages which are provided by malicious users during the phase of feedback collection. It is not difficult to recognize them. By the use of SAM, it is easy to calculate the similarity between the new feedback and the activated one. If the difference is greater than a threshold, it will be regarded as the outlier and it will not be taken into consideration. However, how to determine an adequate threshold is not the issue of this dissertation. All feedbacks are not outliers in the case studies.

4.3.3 System sensitivity

The issue about how frequently the *Fuzzy Moderator* should be activated to calculate or trigger the *Fuzzy Classification* to classify the services depends on the required system sensitivity. If the system is to be sensitive, the processes can be executed automatically when some thresholds are exceeded or at fixed intervals (e.g. every day, every week or every 3 months). Otherwise, the procedures can be manually activated when system is considered to be insensitive.

The sensitivity depends on numerous reasons such as the number of feedback messages, the variation of the service content, etc. It depends on what kind of environment this approach is applied to. It is not appropriate to have a conclusion here. In the case studies, the system will be set in insensitive mode and the moderation process is triggered manually.

CHAPTER 5 CASE STUDIES AND PERFORMANCE EVALUATIONS

This study proposed a consensus-based service discovery approach which attempts to use the underlying data and information about services as a searching criterion (quality rating). With the help of classification, the proposed method could refine the search space and increase the precision rate of service discovery. *Fuzzy Classifier*, which is used to do the classification, contains essential predefined knowledge (criteria) for interpreting and classifying (rating) the information resident in Web services. These criteria can be grouped into two types: primitive terms and composite terms. Primitive terms are a set of atomic terms that represent a collection of the raw data. A primitive term is derived from a fuzzy set $\tilde{A}$ which is defined as $\tilde{A} = \int_x \mu_A(x)/x$, where x is actual value, $\mu_A(x)$ is a member function and the value domain is located between 0 and 1. A Composite term is generated through the combination of primitive terms (detailed in section 4.1.3).

Section 5.1 illustrates a case study with primitive terms and shows how the SAM process (section 3.2) is applied to assist in reaching the consensus on a specific primitive term. Section 5.2 presents a case study with a composite term and demonstrates how the RMGDP process (section 3.3) is triggered to assist in reaching consensus weightings for the specific composite term. Sometimes, when users are not confident with their preferences or the number of classification terms is inappropriate, POPM (section 3.4) could be applied to refine the terms. Such a case will be depicted in the section 5.3. The performance evaluation for each of these three cases will be presented at the end of each case study respectively.

5.1 Case I - Flight booking case study with primitive term

5.1.1 Scenario and the moderation process for Case I

In Case I, the basic steps for Web services discovery are based on the scenario addressed in Section 4.1.1 and the environment for this case is built upon the flight booking services. Ten service providers prepare the advertisements for their flight booking services and publish these advertisements to the *UDDI / OWL-S Registry*. The raw data (price of flight tickets from Taipei to Shanghai) for these ten services were obtained from the Web site (source as [86]) at June 2005.

In Case I, it is assumed that the search criterion, *Cheap*, is a term for quality rating which is used to represent the cost of a flight ticket. *Cheap* is a primitive term and defined as a fuzzy rule. This can be formulated as *Cheap*(Q), or $\tilde{C}$ for brief, where Q represents the actual cost (underlying data) for a specific flight. It is assumed that $\tilde{C}_{init}$ (see Figure 5-1) is populated with an initial value and denoted as $\tilde{C}_{init} = (0,0,14500,16500)$, where $a \le b \le c \le d$.

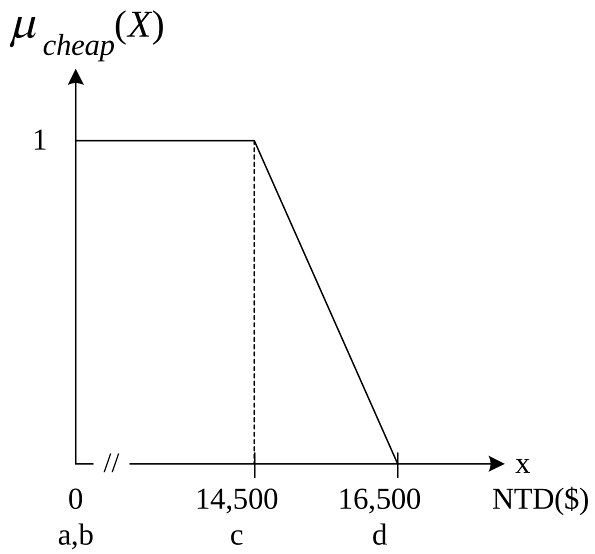

Figure 5-1 $\tilde{C}_{init} = (0,0,14500,16500)$

Given the initial values for the fuzzy rule, the inference rule can be applied to derive the classification result. For instance, if the ticket price is 14500 (NTD), then *Cheap*(Q) = 1, according to above fuzzy rule. However, if the price is 15500 (NTD), then the *Cheap*(Q) = 0.5. The values 1 and 0.5 represent the quality for two different flights under the primitive term *Cheap*. This is the way in which *Fuzzy Classifier* is used to classify each of the flights stored in one specific service and the average value of all the classification results forms the quality rating of one specific service. Initially, each of the ten services is rated by $\tilde{C}_{init}$ and thus each service gets a value which represents its higher level informative declaration (quality rating or QoS).

Service consumers express their needs, *"I need a cheap flight to Shanghai",* to *Fuzzy Discovery* in order to find appropriate flight booking services. This request is vague and contains a quality rating (QoS) description for the desired services. *Fuzzy Discovery* will search the advertisements stored in the *Registry* based on the primitive term – *Cheap*, and satisfying the basic capability – a flight to Shanghai and the context – departing from Taiwan. All flight booking services compliant with these capability requirements will be selected from the *Registry* and held in *Fuzzy Discovery* for further filtering. Owing to the primitive term, *Cheap*, those flight booking services whose quality rating value for *Cheap* is considered not cheap will be ruled out. The filtering criterion is based on the threshold θ. The value of θ is adaptable and is determined by the default setting in *Fuzzy Discovery* or by consumers personalized settings.

Each of the ten services is rated by $\tilde{C}_{init}$, which is arbitrary initialized or preset under the agreement of service providers. However, this initial $\tilde{C}_{init}$ may not be objective so the query results might not conform to consumers' opinions and this gap decreases the precision rate of service discovery. It is important to mitigate the gap between the expectations and preferences of service providers and consumers by moderating the $\tilde{C}_{init}$ according to consumers' feedbacks.

Before the moderation process starts, consumers or users feedbacks should be gathered and it is assumed that there are a group of consumers, denoted as $User_i\,(i = 1,2,3,...,n)$, with their different subjective opinions on the definition of the primitive term *Cheap*. These feedbacks on the term *Cheap* can be denoted as $\tilde{C}_i(a_i, b_i, c_i, d_i)$, where *i* indicates the *i*-th user, and formulated as following fuzzy sets (see Figure 5-2).

$$\tilde{C}_1(a_1, b_1, c_1, d_1) = (0,\ 0,\ 13500,\ 16500)$$
$$\tilde{C}_2(a_2, b_2, c_2, d_2) = (0,\ 0,\ 14500,\ 14500)$$
$$\tilde{C}_3(a_3, b_3, c_3, d_3) = (0,\ 0,\ 14000,\ 15500)$$
$$\tilde{C}_4(a_4, b_4, c_4, d_4) = (0,\ 0,\ 11000,\ 13000)$$

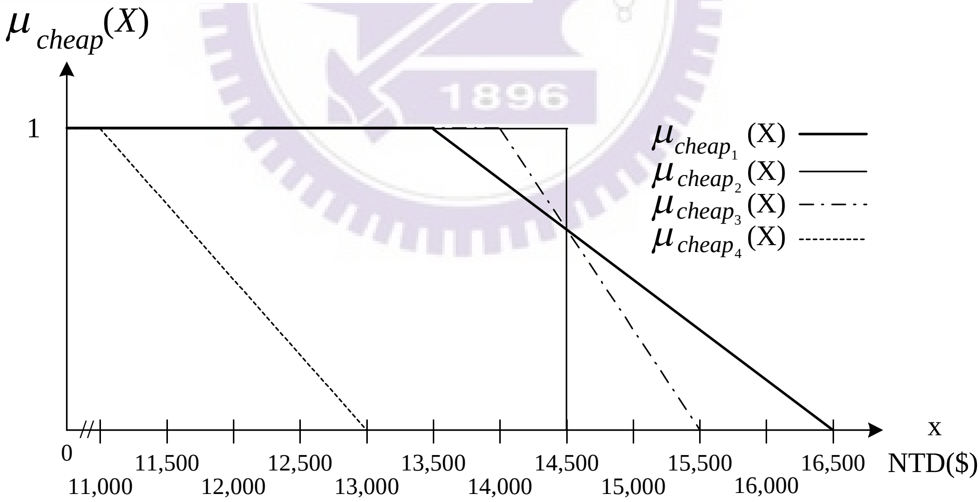

Figure 5-2 Four different fuzzy sets for *Cheap*

During the moderation period, the SAM process is applied for gaining the consensus on the primitive term *Cheap*. From the application of Equation (1), $S_{ij} = S(\tilde{C}_i, \tilde{C}_j)$, the degree of similarity for each pair's opinions, $User_i$ and $User_j$, on the term (or criterion) *Cheap* can be derived. It shows as follows.

$$S(\tilde{C}_1,\tilde{C}_2)=S(\tilde{C}_2,\tilde{C}_1)=\frac{86}{91}, S(\tilde{C}_2,\tilde{C}_3)=S(\tilde{C}_3,\tilde{C}_2)=\frac{173}{178},$$
$$S(\tilde{C}_1,\tilde{C}_3)=S(\tilde{C}_3,\tilde{C}_1)=\frac{176}{181}, S(\tilde{C}_2,\tilde{C}_4)=S(\tilde{C}_4,\tilde{C}_2)=\frac{24}{29},$$
$$S(\tilde{C}_1,\tilde{C}_4)=S(\tilde{C}_4,\tilde{C}_1)=\frac{4}{5}, S(\tilde{C}_3,\tilde{C}_4)=S(\tilde{C}_4,\tilde{C}_3)=\frac{48}{59}$$

Once the similarities of their opinions between all the pairs are obtained, an AM (Agreement Matrix), Equation (2), can be formed. The result is shown as follows:

$$AM = \begin{pmatrix} 1 & \frac{86}{91} & \frac{176}{181} & \frac{4}{5} \\ \frac{86}{91} & 1 & \frac{173}{178} & \frac{24}{29} \\ \frac{176}{181} & \frac{173}{178} & 1 & \frac{48}{59} \\ \frac{4}{5} & \frac{24}{29} & \frac{48}{59} & 1 \end{pmatrix}$$

Once the AM is available, the average agreement degree can be obtained after the use of Equation (3).

$$A(USER_1)=\frac{74598}{82355}$$
$$A(USER_2)=\frac{1289231}{1409226}$$
$$A(USER_3)=\frac{5242283}{5702586}$$
$$A(USER_4)=\frac{20884}{25665}$$

Through Equation (4), each individual *RAD* can be calculated and shown as follows:

$$RAD_1=\frac{74598}{82355}\div\left(\frac{74598}{82355}+\frac{1289231}{1409226}+\frac{5242283}{5702586}+\frac{20884}{25665}\right)=\frac{34079126526}{133698585637}$$
$$RAD_2=\frac{1289231}{1409226}\div\left(\frac{74598}{82355}+\frac{1289231}{1409226}+\frac{5242283}{5702586}+\frac{20884}{25665}\right)=\frac{68838489245}{267397171274}$$
$$RAD_3=\frac{5242283}{5702586}\div\left(\frac{74598}{82355}+\frac{1289231}{1409226}+\frac{5242283}{5702586}+\frac{20884}{25665}\right)=\frac{69171924185}{267397171274}$$
$$RAD_4=\frac{20884}{25665}\div\left(\frac{74598}{82355}+\frac{1289231}{1409226}+\frac{5242283}{5702586}+\frac{20884}{25665}\right)=\frac{30614252396}{133698585637}$$

As mentioned previously, each individual opinion (feedback) is treated with equal importance in Equation (5), so that $\beta = 0$, $CDC_i = RAD_i$, and

$$CDC_1 = RAD_1 = \frac{34079126526}{133698585637}$$
$$CDC_2 = RAD_2 = \frac{68838489245}{267397171274}$$
$$CDC_3 = RAD_3 = \frac{69171924185}{267397171274}$$
$$CDC_4 = RAD_4 = \frac{30614252396}{133698585637}$$

With the application of Equation (6), the consensus on the term *Cheap*(Q) for four different users can be aggregated from individual's feedbacks, $\tilde{C}_i(a_i, b_i, c_i, d_i)$ where *i* indicates the *i*-th user.

$$\begin{aligned}
\tilde{C} = {} & \frac{34079126526}{133698585637} \times \tilde{C}_1(0,0,13500,16500) + \\
& \frac{68838489245}{267397171274} \times \tilde{C}_2(0,0,14500,14500) + \\
& \frac{69171924185}{267397171274} \times \tilde{C}_3(0,0,14000,16000) + \\
& \frac{30614252396}{133698585637} \times \tilde{C}_4(0,0,11000,14500)
\end{aligned}$$
$$\tilde{C} = (0, 0, \frac{1780107500778250}{133698585637}, \frac{1995452328287000}{133698585637})$$
$$\tilde{C} = (0, 0, 13314.333, 14925.007)$$

Initially, a subjectively value, $\tilde{C}_{init} = (0,0,14500,16500)$, was given for the *Fuzzy Classifier* to carry out reasoning. Before the moderation starts, consumers provide their feedback and their opinions on the term *Cheap*, then a number of steps for reaching a consensus have been taken. Finally, a moderated consensus value for the primitive term *Cheap* is derived, namely $\tilde{C} = (0, 0, 13314.333, 14925.007)$, to replace the existing one

($\tilde{C}_{init}$). *Fuzzy Classifier* could allow the less subjective value to evolve in order to achieve better quality of service after more consumers' feedback has been collected.

5.1.2 Performance evaluation for Case I

A case study with four different service consumers and ten different airlines was adopted to evaluate the comparative performance of three different approaches. The proposed approach Moderated Fuzzy Discovery Method (MFDM) is evaluated in comparison to the Capability Discovery Method (CDM) and the Fuzzy Discovery Method (FDM) [17].

5.1.2.1 Capability Discovery Method (CDM)

In the first experiment, service discovery approach is based on the use of UDDI registry and the capability search mechanism without involving any fuzzy discovery and higher level abstraction mechanisms (quality rating). This is called the Capability Discovery Method.

The capability matchmaker suggests all the ten Web services to the consumers, since they satisfy the capability constraints (flight booking service). Thus, each Web service consumer starts to check whether the actual contents of the Web services can meet their requirements or not. Figure 5-2 illustrates the fuzzy sets for service consumers that appear in this case and Table 5-1 shows the results related to the precision rate.

In Table 5-1, service Consumer 1's fuzzy set for *Cheap* is $\tilde{C}_1(a_1, b_1, c_1, d_1)$ =(0, 0, 13500, 16500). It means that Consumer 1 has a subjective opinion on cheap flight price which is between 0 and 16500. As a result, there are only seven airline Web services can meet Consumer 1's requirement. So the precision rate is 70% (7 / 10 = 0.7). Use the same principle and apply it to service Consumer 2 , 3 and 4, then different precision rates can be obtain at 0.5, 0.7, and 0.1 respectively.

Table 5-1 CDM precision rates for service Consumer 1 to 4

CDM Suggestions (No filtering)	C1	C2	C3	C4
ChinaEasternAir	V		V	
DragonAir				
FarEasternAir	V	V	V	V
MacauAir	V		V	
TransasiaAir	V	V	V	
JapanAsiaAir				
ChinaAir	V	V	V	
CathayAir				
EvaAir	V	V	V	
ShanghaiAir	V	V	V	
Precision Rate	7 / 10 = 0.7	5 / 10 = 0.5	7 / 10 = 0.7	1 / 10 = 0.1

5.1.2.2 Fuzzy Discovery Method (FDM)

The second set of experiments in this case is carried out to test the Fuzzy Discovery Method (FDM) [17]. FDM was deployed after the fuzzy classification had been conducted on the underlying data about each service. In this experiment, *Fuzzy Classifier* adopts the arbitrary $\tilde{C}_{init} = (0,0,14500,16500)$, where $a \le b \le c \le d$ (see Figure 5-1), as the fuzzy rule for classification according to the actual cost of a specific flight. Before the FDM is applied for service discovery, each of the ten services will be rated by $\tilde{C}_{init}$ and therefore each service gets a value representing its higher level informative declaration (quality rating or QoS) on the primitive term *Cheap*.

Before the FDM can be deployed, the *Fuzzy Classifier* have to conduct fuzzy classification on the data provided by each service provider. The initial fuzzy set, $\tilde{C}_{init} = (0,0,14500,16500)$, is introduced to calculate primitive term *Cheap* for each service provider. The classification results are shown in Table 5-2.

Table 5-2 Classification results for each service with $\tilde{C}_{init} = (0,0,14500,16500)$

Service	QoS Value for *Cheap*	Service	QoS Value for *Cheap*
ChinaEasternAir	0.4	JapanAsiaAir	0
DragonAir	0	ChinaAir	0.16
FarEasternAir	0.5	CathayAir	0
MacauAir	0.47	EvaAir	0.23
TransasiaAir	0.52	ShanghaiAir	0.16

Suppose that the threshold θ = 0.25 is adopted for all web consumers. θ, the threshold, is used in the *Fuzzy Discovery* to filter out those services that are less likely to meet the requirement. In this experiment, *Fuzzy Discovery* only recommends four possible satisfactory Web services, that is, ChinaEasternAir, FareasternAir, MacauAir and TransasiaAir. Consumer 2 with fuzzy set $\tilde{C}_2(a_2, b_2, c_2, d_2)$=(0, 0, 14500, 14500) indicates that his / her subjective cheap price sits between 0 and 14500. From the evaluation result shown in Table 5-3, it can be observed that only two flight booking services can satisfy his / her requirement. For service Consumer 2, the precision rate is 50% (2 / 4 = 0.5). In addition, the same principle can be also applied to Consumer 1, 3, and 4 and the results are 100%, 100%, and 25% respectively for the precision rates.

Table 5-3 FDM precision rates for Consumer 1 to 4 with θ = 0.25

θ = 0.25 FDM Suggestions	C1	C2	C3	C4
ChinaEasternAir	V		V	
FarEasternAir	V	V	V	V
MacauAir	V		V	
TransasiaAir	V	V	V	
Precision Rate for Specific Consumer	4 / 4 = 1	2 / 4 = 0.5	4 / 4 = 1	1 / 4 = 0.25

If θ is 0.5, only FareasternAir and TransasiaAir will be recommended and the precision rates for FDM are revealed in Table 5-4.

Table 5-4 FDM precision rates for Consumer 1 to 4 with θ = 0.5

θ= 0.5 / FDM Suggestions	C1	C2	C3	C4
FarEasternAir	V	V	V	V
TransasiaAir	V	V	V	
Precision Rate for Specific Consumer	2 / 2 = 1	2 / 2 = 1	2 / 2 = 1	1 / 2 = 0.5

5.1.2.3 Moderated Fuzzy Discovery Method (MFDM)

The third set of experiments is conducted to test the Moderated Fuzzy Discovery Method (MFDM). After four service consumers have made the requests via the *Fuzzy Discovery* and give their feedbacks or opinions on the primitive term *Cheap*. The SAM method will be conducted by *Fuzzy Moderator* to aggregate the group consensus on primitive term *Cheap* in order to produce a more objective inference rule. This process has been detailed in section 5.1.1 and a moderated consensus value for primitive term *Cheap* is derived as $\tilde{C} = (0, 0, 13314.333, 14925.007)$. This consensual value will replace the existing one ($\tilde{C}_{init} = (0,0,14500,16500)$). With the new derived fuzzy set, *Fuzzy Classifier* will be triggered again in order to obtain new classification result for the term *Cheap*. This is illustrated in Table 5-5.

Table 5-5 Classification results for each service with moderated $\tilde{C} = (0, 0, 13314.333, 14925.007)$

Service	QoS Value for *Cheap*	Service	QoS Value for *Cheap*
ChinaEasternAir	0.01	JapanAsiaAir	0
DragonAir	0	ChinaAir	0.14
FarEasternAir	0.5	CathayAir	0
MacauAir	0.1	EvaAir	0.08
TransasiaAir	0.26	ShanghaiAir	0.13

In this experiment, only two flight booking services are above the threshold θ = 0.25, that is, only two possible Web service, FarEasternAir and TansasiaAir, will be recommended by *Fuzzy Discovery*. Consumer 3 with fuzzy set $\tilde{C}_3(a_3, b_3, c_3, d_3)$=(0, 0, 14000, 15500) indicates that his / her subjective cheap price sits between 0 and 15500. From the result shown in Table 5-6, two of the recommended flight booking services can satisfy service Consumer 3's subjective opinion. The precision rate has increased to 100% (2 / 2 = 1), due to the contribution of the proposed moderation. By applying the same steps to the other service Consumers 1, 2, and 4, their precision rates would therefore be 100%, 100%, and 50% respectively.

Table 5-6 MFDM precision rates for Consumer 1 to 4 with θ = 0.25

θ= 0.25 MFDM Suggestions	C1	C2	C3	C4
FarEasternAir	∨	∨	∨	∨
TransasiaAir	∨	∨	∨	
Precision Rate for Specific Consumer	2 / 2 = 1	2 / 2 = 1	2 / 2 = 1	1 / 2 = 0.5

If θ is 0.5, only FarEasternAir will be recommended and the precision rates for MFDM are revealed Table 5-7.

Table 5-7 MFDM precision rates for Consumer 1 to 4 with θ = 0.5

θ= 0.5 MFDM Suggestions	C1	C2	C3	C4
FarEasternAir	∨	∨	∨	∨
Precision Rate for Specific Consumer	1 / 1 = 1	1 / 1 = 1	1 / 1 = 1	1 / 1 = 1

5.1.2.4 Summary of Case I

Table 5-8 shows an integrated view of Table 5-1,Table 5-3,Table 5-4,Table 5-6 and Table 5-7. The average precision rates for CDM, FDM and MFDM are indicated in Table 5-8 with different thresholds.

From Table 5-8, it can be concluded that the proposed Moderated Fuzzy Discovery Method (MFDM) has outperformed the Fuzzy Discovery Method (FDM) and the FDM has produced better precision rate than the Capability Discovery Method (CDM). In addition, MFDM has performed twice as well as the CDM in terms of precision rate.

Table 5-8 Precision rates for CDM, FDM and MFDM with different thresholds

Precision Rates for Specific Consumer	C1		C2		C3		C4		Average Precision Rate	
θ	0.25	0.5	0.25	0.5	0.25	0.5	0.25	0.5	0.25	0.5
CDM	0.7		0.5		0.7		0.1		0.5	
FDM	1	1	0.5	1	1	1	0.25	0.5	0.68	0.87
MFDM	1	1	1	1	1	1	0.5	1	0.87	1

Through the consideration of quality rating on the perspective *Cheap*, and the use of the proposed moderation process, the precision rate of service discovery can be improved by pre-classifying services and filtering out those services whose quality of underlying content is not considered as a recommended service. This will save the consumers' time while selecting the suitable services.

The results show that CDM is the most imprecise way for service discovery. Nevertheless, CDM uses general UDDI inquiries where no additional pre-classification is needed before service discovery. Both of FDM and MFDM need the additional computation cost for classification (time for evaluating the QoS terms of all services). In this experiment, the time for pre-classification process is less than 1 second. MFDM consumes extra 0.921875 second for SAM processing time. Briefly, if CDM is treated as a basis, then FDM consumes less than 1 additional second and MFDM requires an extra 1.732875(+-0.5) seconds. The additional time is trivial but it does increase the computational cost when FDM and MFDM are applied. The cost might vary according to the amount of data and the number of feedback classifications. Considering the time gained from the increase of precision rate and the time saved by filtering out the less significant services, MFDM is a better solution for service discovery.

In Case I, however, only one perspective, *Cheap*, is used. Different weightings from different service consumers for the ingredients of a composite term are not considered. In addition, the number of consumers is small in Case I. Therefore, in the next case study, a larger scale of exercises with multiple criteria will be conducted in order to examine the issues associated with scalability.

5.2 Case II - Flight booking case study with composite term

5.2.1 Scenario and the moderation process for Case II

In the previous case, only one perspective on quality, *Cheap*, is considered and the number of consumers is relatively small. For this reason, a larger scale case with multiple criteria will be considered in this section to examine the performance of the proposed method. This section presents a case study with a composite term and demonstrates how the RMGDP process is triggered to assist in reaching the consensus over the weightings for the ingredients of a composite term. The MFDM applied in Case II comprises two parts: SAM and RMGDP. These parts are processed in a sequence so that SAM is initiated first to gain a consensus from the distinct opinions on the specific primitive terms. RMGDP then obtains the group preferences on the different selection criteria which are the ingredients of a composite term.

The context of Case II is also based on the flight booking services. Advertisements of nine service providers are included in the *UDDI / OWL-S Registry*. The raw data (price of flight tickets from Taipei to London) for these nine services were obtained from the Web site (source as [87],[88],[89]) at August 2005. The feedback in this case is gathered from thirty practical consumers by questionnaires.

Service consumers may express their needs, *"I need a satisfactory flight to London",* to *Fuzzy Discovery* for finding flight booking services. *Fuzzy Discovery* will search the advertisements stored in the *Registry* based on the QoS term – *Satisfaction*, and satisfying the basic capability – a flight to London and the context – departing from Taiwan. All flight booking services compliant with these capability requirements will be selected and held in the *Fuzzy Discovery* for further filtering. Owing to the QoS term, *Satisfaction*, those services whose quality rating value for *Satisfaction* is considered not good enough will be ruled out. The filtering criterion is based on the threshold θ. The value of θ is adaptable and is determined by the default setting in *Fuzzy Discovery* or by consumers' settings.

In Case II, *Satisfaction* is a composite term defined as a fuzzy rule which represents the overall quality of a flight ticket. This term is denoted as *Satisfaction*(Q), or $\tilde{Q}$ for brief, where Q represents the underlying data for a specific flight. *Satisfaction*(Q) can be rated from five different independent perspectives on a flight ticket and it is derived from the following primitive terms:

1. *Cheap*: It is a measurement of the cost of a flight ticket. It is denoted as *Cheap*(Q) or $\tilde{C}$. It is the same as the primitive term used in the previous case.
2. *DepartureTime*: It indicates the desirable (ideal) flight departure time (in minutes). It is denoted as *DepartureTime*(Q) or $\tilde{D}$.
3. *ArrivalTime*: It indicates the desirable flight arrival time (in minutes). It is denoted as *ArrivalTime*(Q) or $\tilde{A}$.
4. *TravelTime*: It represents the desirable duration of total travelling time. It is denoted as *TravelTime*(Q) or $\tilde{T}$. Notice: $\tilde{T}$ is not the difference between $\tilde{A}$ and $\tilde{D}$.
5. *Stops*: It represents the number of stops a flight has to make before reaching destination. It is denoted as *Stops*(Q) or $\tilde{S}$.

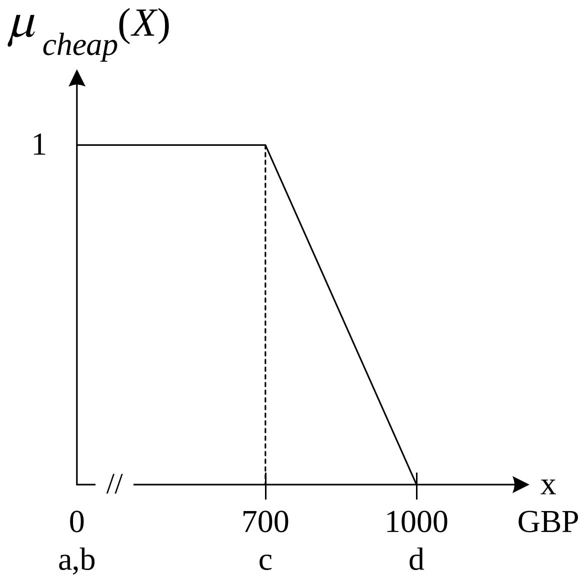

Figure 5-3 $\tilde{C}_{init} = (0,0,700,1000)$

It is assumed that $\tilde{C}_{init}$ (as shown in Figure 5-3) is populated with an initial value and denoted as $\tilde{C}_{init} = (0,0,700,1000)$. Similarly, $\tilde{D}_{init} = (600,660,1080,1260)$, $\tilde{T}_{init} = (0,0,1700,2200)$, $\tilde{A}_{init} = (720,780,1140,1260)$, and $\tilde{S}_{init} = (0,0,2,2)$ where $a \le b \le c \le d$. Thus, the initial degree of the composite term *Satisfaction*(Q), or $\tilde{Q}_{init}$, can be obtained by assigning them with equal weighting and adding them up:

$$\tilde{Q}_{init} = 1/5 \times \tilde{C}_{init} + 1/5 \times \tilde{D}_{init} + 1/5 \times \tilde{T}_{init} + 1/5 \times \tilde{A}_{init} + 1/5 \times \tilde{S}_{init}$$

Given the initial values for the fuzzy rule, the inference rules can be used to derive the classification result. For instance, if the ticket price is 700 (GBP), then $\tilde{C}_{init}$ = *Cheap*(Q) = 1, according to the fuzzy rule shown in Figure 5-3. However, if the price is 850 (GBP), then $\tilde{C}_{init}$ = *Cheap*(Q) = 0.5. The value of 1 and 0.5 represents the quality for two different flights according to the definition of the primitive term *Cheap*. The same way can be applied on $\tilde{D}_{init}$, $\tilde{T}_{init}$, $\tilde{A}_{init}$, and $\tilde{S}_{init}$ to derive the composite term $\tilde{Q}_{init}$. This is the way which *Fuzzy Classifier* is used to classify each of the flights stored in one specific service and the average value of all classification results forms the quality rating of one specific service in related to the composite term *Satisfaction*.

Initially, the arbitrary initialized values, $\tilde{C}_{init}$, $\tilde{D}_{init}$, $\tilde{T}_{init}$, $\tilde{A}_{init}$, and $\tilde{S}_{init}$, with equal weighting will be used as inputs to derive $\tilde{Q}_{init}$, and each of the nine services will be rated by $\tilde{Q}_{init}$ and thus each service gets a value which represents its higher level informative declaration (quality rating or QoS). However, this initial $\tilde{Q}_{init}$ may not be objective so the query results might not conform to consumers' opinions and this gap decreases the precision rate of the service discovery. It is important to mitigate the gap by moderating the $\tilde{Q}_{init}$ according to consumer feedback. Later, these arbitrary initialized values will be replaced by the consensus values derived from the SAM resolution process. After the RMGDP process, the initial equal weighting will also be modified to reflect the situation based on consumer feedback.

The SAM Process:

Consider a group of service consumers, $User_i (i = 1,2,3,...,30)$, having different subjective opinions on the definition of the primitive term *Cheap*. These preferences for the term *Cheap* are listed in Table 5-9 and it can be denoted as $\tilde{C}_i(a_i, b_i, c_i, d_i)$, where i indicates the i-th user, and formulated as fuzzy sets. For example:

$\tilde{C}_1(a_1, b_1, c_1, d_1)$ = (0,0,450,600),
$\tilde{C}_2(a_2, b_2, c_2, d_2)$ = (0,0,500,650),
$\tilde{C}_3(a_3, b_3, c_3, d_3)$ = (0,0,500,700),
$\tilde{C}_4(a_4, b_4, c_4, d_4)$ = (0,0,600,800), … , and
$\tilde{C}_{30}(a_{30}, b_{30}, c_{30}, d_{30})$ = (0,0,600,700)

Consumer feedback or preferences for the remaining terms, *DepartureTime*, *TravelTime*, *ArrivalTime*, and *Stops*, were also collected from the same 30 consumers. These are denoted as $\tilde{D}_i(a_i, b_i, c_i, d_i)$, $\tilde{T}_i(a_i, b_i, c_i, d_i)$, $\tilde{A}_i(a_i, b_i, c_i, d_i)$, and $\tilde{S}_i(a_i, b_i, c_i, d_i)$, and are recorded in Table 5-10 to Table 5-13.

Table 5-9 Consumers' preferences for *Cheap*

$\tilde{C}_i(a_i, b_i, c_i, d_i)$			
i = 1	(0,0,450,600)	i = 16	(0,0,500,700)
i = 2	(0,0,500,650)	i = 17	(0,0,600,700)
i = 3	(0,0,500,700)	i = 18	(0,0,700,900)
i = 4	(0,0,600,800)	i = 19	(0,0,600,900)
i = 5	(0,0,700,900)	i = 20	(0,0,700,1000)
i = 6	(0,0,400,500)	i = 21	(0,0,800,1100)
i = 7	(0,0,500,700)	i = 22	(0,0,500,700)
i = 8	(0,0,800,900)	i = 23	(0,0,700,900)
i = 9	(0,0,550,700)	i = 24	(0,0,800,1000)
i = 10	(0,0,500,800)	i = 25	(0,0,600,800)
i = 11	(0,0,400,500)	i = 26	(0,0,700,900)
i = 12	(0,0,450,650)	i = 27	(0,0,600,700)
i = 13	(0,0,600,800)	i = 28	(0,0,750,850)
i = 14	(0,0,650,900)	i = 29	(0,0,700,800)
i = 15	(0,0,350,500)	i = 30	(0,0,600,700)

Table 5-10 Consumers' preferences for *DepartureTime*

$\tilde{D}_i(a_i, b_i, c_i, d_i)$			
i = 1	(540,660,960,1080)	i = 16	(420,540,660,780)
i = 2	(420,540,900,1020)	i = 17	(360,480,600,720)
i = 3	(360,480,600,720)	i = 18	(600,720,840,960)
i = 4	(420,540,600,720)	i = 19	(600,720,840,960)
i = 5	(600,720,840,960)	i = 20	(480,600,660,780)
i = 6	(720,840,1020,1140)	i = 21	(480,660,780,900)
i = 7	(660,780,900,1020)	i = 22	(420,540,720,840)
i = 8	(480,600,840,960)	i = 23	(600,720,840,960)
i = 9	(1260,1320,1380,1440)	i = 24	(420,540,600,720)
i = 10	(840,960,1140,1260)	i = 25	(420,540,660,780)
i = 11	(420,540,660,780)	i = 26	(480,600,720,840)
i = 12	(360,480,660,780)	i = 27	(720,840,900,1020)
i = 13	(420,540,660,780)	i = 28	(600,720,840,960)
i = 14	(480,600,840,960)	i = 29	(540,660,720,840)
i = 15	(600,720,900,1020)	i = 30	(900,1020,1140,1260)

Table 5-11 Consumers' preferences for *TravelTime*

$\tilde{T}_i(a_i, b_i, c_i, d_i)$			
i = 1	(0,0,1350,1590)	i = 16	(0,0,1410,1530)
i = 2	(0,0,1170,1470)	i = 17	(0,0,1350,1710)
i = 3	(0,0,1350,1650)	i = 18	(0,0,1350,1410)
i = 4	(0,0,1410,1590)	i = 19	(0,0,1410,1470)
i = 5	(0,0,1290,1530)	i = 20	(0,0,1350,1650)
i = 6	(0,0,1230,1470)	i = 21	(0,0,1410,1470)
i = 7	(0,0,1470,1590)	i = 22	(0,0,1470,1650)
i = 8	(0,0,1350,1470)	i = 23	(0,0,1530,1710)
i = 9	(0,0,1470,1530)	i = 24	(0,0,1470,1710)
i = 10	(0,0,1410,1650)	i = 25	(0,0,1350,1530)
i = 11	(0,0,1470,1590)	i = 26	(0,0,1350,1650)
i = 12	(0,0,1350,1650)	i = 27	(0,0,1470,1710)
i = 13	(0,0,1350,1410)	i = 28	(0,0,1410,1530)
i = 14	(0,0,1290,1410)	i = 29	(0,0,1470,1710)
i = 15	(0,0,1470,1710)	i = 30	(0,0,1410,1650)

Table 5-12 Consumers' preferences for *ArrivalTime*

$\tilde{A}_i(a_i, b_i, c_i, d_i)$			
i = 1	(540,660,960,1080)	i = 16	(420,540,720,840)
i = 2	(480,600,1080,1200)	i = 17	(660,780,840,960)
i = 3	(540,660,960,1080)	i = 18	(660,780,960,1080)
i = 4	(480,600,660,780)	i = 19	(840,960,1140,1260)
i = 5	(780,900,1020,1140)	i = 20	(780,900,1080,1200)
i = 6	(480,600,840,960)	i = 21	(720,840,1140,1260)
i = 7	(480,600,900,1020)	i = 22	(420,540,600,720)
i = 8	(480,600,840,960)	i = 23	(540,660,780,900)
i = 9	(480,600,840,960)	i = 24	(570,690,840,960)
i = 10	(960,1080,1200,1320)	i = 25	(600,720,840,960)
i = 11	(720,840,1020,1140)	i = 26	(900,1020,1140,1260)
i = 12	(780,900,1080,1200)	i = 27	(900,1020,1200,1320)
i = 13	(660,780,900,1020)	i = 28	(780,900,960,1080)
i = 14	(600,720,1140,1260)	i = 29	(420,540,600,720)
i = 15	(420,540,660,780)	i = 30	(900,1020,1140,1260)

Table 5-13 Consumers' preferences for *Stops*

$\tilde{S}_i(a_i, b_i, c_i, d_i)$					
i = 1	(0,0,1,2)	i = 11	(0,0,0,1)	i = 21	(0,0,0,1)
i = 2	(0,0,0,1)	i = 12	(0,0,1,2)	i = 22	(0,0,1,2)
i = 3	(0,0,1,2)	i = 13	(0,0,1,2)	i = 23	(0,0,1,2)
i = 4	(0,0,0,1)	i = 14	(0,0,0,1)	i = 24	(0,0,1,2)
i = 5	(0,0,1,2)	i = 15	(0,0,1,2)	i = 25	(0,0,1,2)
i = 6	(0,0,1,2)	i = 16	(0,0,1,2)	i = 26	(0,0,1,2)
i = 7	(0,0,1,2)	i = 17	(0,0,2,3)	i = 27	(0,0,1,2)
i = 8	(0,0,0,1)	i = 18	(0,0,1,2)	i = 28	(0,0,1,2)
i = 9	(0,0,1,2)	i = 19	(0,0,2,3)	i = 29	(0,0,1,2)
i = 10	(0,0,1,2)	i = 20	(0,0,2,3)	i = 30	(0,0,1,2)

By the use of Equation (1), $S_{ij} = S(\tilde{C}_i, \tilde{C}_j)$, the degree of similarity for each pair's opinions, $User_i$ and $User_j$, on the term *Cheap* can be calculated as follows:

$$S(\tilde{C}_1,\tilde{C}_2) = S(\tilde{C}_2,\tilde{C}_1) = \frac{21}{23}, \quad S(\tilde{C}_{27},\tilde{C}_{30}) = S(\tilde{C}_{30},\tilde{C}_{27}) = \frac{7}{13},$$
$$S(\tilde{C}_1,\tilde{C}_3) = S(\tilde{C}_3,\tilde{C}_1) = \frac{7}{8}, \quad S(\tilde{C}_{28},\tilde{C}_{30}) = S(\tilde{C}_{30},\tilde{C}_{28}) = \frac{53}{131},$$
$$S(\tilde{C}_1,\tilde{C}_4) = S(\tilde{C}_4,\tilde{C}_1) = \frac{3}{4}, \ldots, \; S(\tilde{C}_{29},\tilde{C}_{30}) = S(\tilde{C}_{30},\tilde{C}_{29}) = \frac{41}{91}$$

Once the similarities of opinions between all pairs are obtained, the AM (Agreement Matrix) for the term *Cheap* can be formed as follows. AM is a 30 × 30 matrix. For brevity, remaining elements of this matrix are omitted and the complete matrix is attached in the appendix.

$$AM = \begin{pmatrix} 1 & \frac{21}{23} & \frac{7}{8} & \frac{3}{4} & \square & \square & \square & \square & \square \\ \frac{21}{23} & 1 & \frac{23}{24} & \frac{23}{28} & \square & \square & \square & \square & \square \\ \frac{7}{8} & \frac{23}{24} & 1 & \frac{6}{7} & \square & \square & \square & \square & \square \\ \frac{3}{4} & \frac{23}{28} & \frac{6}{7} & 1 & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & 1 & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & 1 & \frac{13}{16} & \frac{13}{15} & 1 \\ \square & \square & \square & \square & \square & \frac{13}{16} & 1 & \frac{15}{16} & \frac{13}{16} \\ \square & \square & \square & \square & \square & \frac{13}{15} & \frac{15}{16} & 1 & \frac{13}{15} \\ \square & \square & \square & \square & \square & 1 & \frac{13}{16} & \frac{13}{15} & 1 \end{pmatrix}$$

of the primitive term *Cheap*

Once the AM for a term *Cheap* is available, Equation (3) is used to obtain the average agreement degree. For brevity, only four users are illustrated.

$$A(USER_1) = \frac{4046636820883}{5348279736800} = 0.7566$$
$$A(USER_2) = \frac{3530282949451}{4375865239200} = 0.8068$$
$$\cdots \qquad \cdots$$
$$A(USER_{29}) = \frac{912005413}{1091817520} = 0.8353$$
$$A(USER_{30}) = \frac{48244990777}{57076503120} = 0.8453$$

of the primitive term *Cheap*

Through Equation (4), each individual *RAD* can be calculated. Again, only four *RAD*s are demonstrated for brevity.

$$RAD_1 = 0.7566 \div (0.7566 + 0.8068 + 0.8353 + 0.8453) = 0.2332$$
$$RAD_2 = 0.8068 \div (0.7566 + 0.8068 + 0.8353 + 0.8453) = 0.2487$$
$$\cdots \qquad \cdots \qquad \cdots$$
$$RAD_{29} = 0.8353 \div (0.7566 + 0.8068 + 0.8353 + 0.8453) = 0.2575$$
$$RAD_{30} = 0.8453 \div (0.7566 + 0.8068 + 0.8353 + 0.8453) = 0.2606$$

of the primitive term *Cheap*

As mentioned previously, we treated each individual opinion (feedback) with equal importance so $\beta = 0$ and $CDC_i = RAD_i$ (see Equation (5)).

$$CDC_1 = RAD_1 = 0.2332$$
$$CDC_2 = RAD_2 = 0.2487$$
$$\vdots$$
$$CDC_{29} = RAD_{29} = 0.2575$$
$$CDC_{30} = RAD_{30} = 0.2606$$

of the primitive term *Cheap*

Through the use of Equation (6), the primitive term *Cheap* can be aggregated from 30 different consumers' opinions, $\tilde{C}_i(a_i, b_i, c_i, d_i)$ where $i = 1,2,3,\ldots,30$.

$$\begin{aligned}\tilde{C} = {} & 0.2332 \times \tilde{C}_1(0,0,450,600) + \\ & 0.2487 \times \tilde{C}_2(0,0,500,650) + \\ & \quad \ldots \\ & 0.2575 \times \tilde{C}_{29}(0,0,700,800) + \\ & 0.2606 \times \tilde{C}_{30}(0,0,600,700)\end{aligned}$$

$$\tilde{C} = (0, 0, 596.1289, 778.4472)$$

Initially, a subjective value, $\tilde{C}_{init} = (0,0,700,1000)$, was given to carry out reasoning. Before the moderation starts, consumers provide their feedbacks and opinions on the term *Cheap*, and then a number of steps for reaching consensus have been taken. Finally, a moderated consensus value for primitive term *Cheap* is derived, that is $\tilde{C} = (0,0,596.1289,778.4472)$, to replace an existing $\tilde{C}_{init}$. Following the same steps, the consensus value for the remaining primitive terms, *DepartureTime*, *TravelTime*, *ArrivalTime, and Satisfaction*, can be obtained as follows:

$\tilde{D} = (500.906,623.325,770.581,890.581)$,

$\tilde{T} = (0,0,1388.56,1580.58)$,

$\tilde{A} = (621.255,741.255,994.676,1064.68)$, and

$\tilde{S} = (0,0,0.95,1.95)$

The above replace existing $\tilde{D}_{init}$, $\tilde{T}_{init}$, $\tilde{A}_{init}$, and $\tilde{S}_{init}$ respectively. *Fuzzy Classifier* could use the less subjective values, $\tilde{C}, \tilde{D}, \tilde{T}, \tilde{A}$, and $\tilde{S}$, to attain better effectiveness in service discovery, since the new values represent consumers' have consensus on the definition of different terms.

The RMGDP Process:

The SAM method allows service providers and consumers to reach a consensus on the definitions of primitive terms and derive new values for these terms. However, even with the new values of $\tilde{C}$, $\tilde{D}$, $\tilde{T}$, $\tilde{A}$, and $\tilde{S}$, the difficulty of determining the value for the composite term *Satisfaction*(Q) or $\tilde{Q}$ still exists. This results in the adoption of the equal weighting assigned to $\tilde{C}_{init}$, $\tilde{D}_{init}$, $\tilde{T}_{init}$, $\tilde{A}_{init}$, and $\tilde{S}_{init}$ which are the contributing elements for the value of $\tilde{Q}$. Note that default weighting (equal weighing approach) may not be a realistic assignment. In order to model the composite term in a way that can be acceptable to service consumers and providers, it is essential to take their preferences into account. Thus, the service consumers have to express their preference on terms *Cheap*, *DepartureTime*, *TravelTime*, *ArrivalTime* and *Stops*, explicitly in the order accorded to their importance (this is so called preference ordering). Through the use of RMGDP, the group consensus on the importance of different criteria based on their subjective preferences can be reached. Finally, two indexes, GDD and GNDD, can be used to determine the weighting for each individual criterion. As a result, the composite term can be defined less subjectively.

Assume that there is a list of alternatives, $A = \{a_1, a_2, a_3, a_4, a_5\}$, where a_1 represents primitive term *Cheap*, a_2 is *DepartureTime*, a_3 is *TravelTime*, a_4 is *ArrivalTime*, and a_5 is *Stops*. Furthermore, each consumer *k*, denoted as $User_k\ (k = 1,2,3,...,30)$, sorts these alternatives descendingly according to his / her preference as shown in Table 5-14. For example, $User_1$ has a preference $A^1 = \{a_1, a_3, a_2, a_5, a_4\}$ which means $User_1$ prefers a_1 to a_4 and a_2 to a_5, and $User_1$ assigned 1st order to a_1, 2nd order to a_3, 3rd order to a_2, 4th order to a_5, and 5th order to a_4.

Table 5-14 Individual preferences on a list of alternatives, $A = \{a_1, a_2, a_3, a_4, a_5\}$

User #	Sorted list	User #	Sorted list	User #	Sorted list
k = 1	A{1,3,2,5,4}	k = 11	A{1,3,2,4,5}	k = 21	A{3,2,4,5,1}
k = 2	A{1,3,2,5,4}	k = 12	A{1,3,2,4,5}	k = 22	A{1,3,5,2,4}
k = 3	A{1,5,3,2,4}	k = 13	A{2,4,3,5,1}	k = 23	A{4,2,5,1,3}
k = 4	A{2,5,1,3,4}	k = 14	A{3,4,2,5,1}	k = 24	A{3,2,4,5,1}
k = 5	A{3,4,5,1,2}	k = 15	A{1,2,4,3,5}	k = 25	A{3,2,4,1,5}
k = 6	A{1,3,4,5,2}	k = 16	A{1,3,2,4,5}	k = 26	A{3,2,4,1,5}
k = 7	A{1,5,3,2,4}	k = 17	A{1,4,2,5,3}	k = 27	A{1,4,2,3,5}
k = 8	A{5,3,4,2,1}	k = 18	A{5,3,1,2,4}	k = 28	A{2,4,5,3,1}
k = 9	A{1,2,3,5,4}	k = 19	A{1,4,2,3,5}	k = 29	A{3,2,4,1,5}
k = 10	A{3,2,1,4,5}	k = 20	A{2,4,5,3,1}	k = 30	A{1,4,3,2,5}

For a specific $User_k$, data in Table 5-14 can be reformulated as $O^k = \{o_1^k, o_2^k, .., o_m^k\}$, where m is the number of alternative and o_m^k means the order assigned to alternative a_m. For instance, $A^3 = \{a_1, a_5, a_3, a_2, a_4\}$ can be reformulated as $O^3 = \{1,4,3,5,2\}$. All the individual preferences of alternatives are reformulated as shown in Table 5-15.

Table 5-15 $O^k = \{o_1^k, o_2^k, .., o_m^k\}$

$O^k = \{o_1^k, o_2^k, o_3^k, o_4^k, o_5^k\}$					
k = 1	{1,3,2,5,4}	k = 11	{1,3,2,4,5}	k = 21	{5,2,1,3,4}
k = 2	{1,3,2,5,4}	k = 12	{1,3,2,4,5}	k = 22	{1,4,2,3,5}
k = 3	{1,4,3,5,2}	k = 13	{5,1,3,2,4}	k = 23	{4,2,5,1,3}
k = 4	{3,1,4,5,2}	k = 14	{5,3,1,2,4}	k = 24	{5,2,1,3,4}
k = 5	{4,5,1,2,3}	k = 15	{1,2,4,3,5}	k = 25	{4,2,1,3,5}
k = 6	{1,5,2,3,4}	k = 16	{1,3,2,4,5}	k = 26	{4,2,1,3,5}
k = 7	{1,4,3,5,2}	k = 17	{1,3,5,2,4}	k = 27	{1,3,4,2,5}
k = 8	{5,4,2,3,1}	k = 18	{3,4,2,5,1}	k = 28	{5,1,4,2,3}
k = 9	{1,2,3,5,4}	k = 19	{1,3,4,2,5}	k = 29	{4,2,1,3,5}
k = 10	{3,2,1,4,5}	k = 20	{5,1,4,2,3}	k = 30	(1,4,3,2,5)

For any two ordering preference values, o_i^k, o_j^k, assessed by $User_k$, a preference relation, p_{ij}^k in Equation (7) (Section 3.3.1), shows that $User_k$ has a subjective ordering preference of the alternative a_i over alternative a_j. For each consumer, the preference ordering $O^k = \{o_1^k, o_2^k, .., o_m^k\}$ can be transformed again into preference relation (p_{ij}^k) as follows: (Note that $p_{ij}^3 \sim p_{ij}^{28}$ are omitted for brevity.)

$$p_{ij}^{1}\begin{bmatrix} 0.5 & 0.75 & 0.625 & 1 & 0.875 \\ 0.25 & 0.5 & 0.375 & 0.75 & 0.625 \\ 0.375 & 0.625 & 0.5 & 0.875 & 0.75 \\ 0 & 0.25 & 0.125 & 0.5 & 0.375 \\ 0.125 & 0.375 & 0.25 & 0.625 & 0.5 \end{bmatrix}, p_{ij}^{2}\begin{bmatrix} 0.5 & 0.75 & 0.625 & 1 & 0.875 \\ 0.25 & 0.5 & 0.375 & 0.75 & 0.625 \\ 0.375 & 0.625 & 0.5 & 0.875 & 0.75 \\ 0 & 0.25 & 0.125 & 0.5 & 0.375 \\ 0.125 & 0.375 & 0.25 & 0.625 & 0.5 \end{bmatrix}$$

$$,\ldots,$$

$$p_{ij}^{29}\begin{bmatrix} 0.5 & 0.25 & 0.125 & 0.375 & 0.625 \\ 0.75 & 0.5 & 0.375 & 0.625 & 0.875 \\ 0.875 & 0.625 & 0.5 & 0.75 & 1 \\ 0.625 & 0.375 & 0.25 & 0.5 & 0.75 \\ 0.375 & 0.125 & 0 & 0.25 & 0.5 \end{bmatrix}, p_{ij}^{30}\begin{bmatrix} 0.5 & 0.875 & 0.75 & 0.625 & 1 \\ 0.125 & 0.5 & 0.375 & 0.25 & 0.625 \\ 0.25 & 0.625 & 0.5 & 0.375 & 0.75 \\ 0.375 & 0.75 & 0.625 & 0.5 & 0.875 \\ 0 & 0.375 & 0.25 & 0.125 & 0.5 \end{bmatrix}$$

After transforming preference orderings into fuzzy preference relations, the collective preference relation p_{ij}^{c} can be calculated by using Equation (8) (Section 3.3.2). In this case, all consumers' opinions are treated on an equal basis so that the corresponding OWA operator with the weighting vector would be w = (1/30, 1/30, 1/30, 1/30 ...1/30, 1/30, 1/30) and the p_{ij}^{c} is as follows:

$$p_{ij}^{c}\begin{bmatrix} 0.5 & 0.5167 & 0.4833 & 0.575 & 0.6542 \\ 0.4833 & 0.5 & 0.4667 & 0.5583 & 0.6375 \\ 0.5167 & 0.5333 & 0.5 & 0.5917 & 0.6708 \\ 0.425 & 0.4417 & 0.4083 & 0.5 & 0.5792 \\ 0.3458 & 0.3625 & 0.3292 & 0.4208 & 0.5 \end{bmatrix}$$

Moreover, the Quantifier Guided Dominance Degree (QGDD) and Quantifier Guided Non-Dominance Degree (QGNDD) could be obtained using Equation (10) and Equation (11) (Section 3.3.3). The level of three alternatives importance is evidently identified through the application of QGDD and QGNDD. This result is shown in Table 5-16. It is interesting to note that both QGDD and QGNDD have drawn the same conclusion that is, $a_3(TravelTime)$ > $a_1(Cheap)$ > $a_2(DepartureTime)$ > $a_4(ArrivalTime)$ > $a_5(Stops)$ which is shown in Figure 5-4 and Figure 5-5.

Table 5-16　QGDD, QGNDD and the consensus weightings for alternatives

QGDD for alternatives	a_1 $(Cheap)$	a_2 $(DepartureTime)$	a_3 $(TravelTime)$	a_4 $(ArrivalTime)$	a_5 $(Stops)$
	0.5573	0.5365	0.5781	0.4635	0.3646
QGNDD for alternatives	a_1 $(Cheap)$	a_2 $(DepartureTime)$	a_3 $(TravelTime)$	a_4 $(ArrivalTime)$	a_5 $(Stops)$
	0.9917	0.975	1	0.8875	0.7292
consensus weightings for alternatives from QGDD	wa_1 $(Cheap)$	wa_2 $(DepartureTime)$	wa_3 $(TravelTime)$	wa_4 $(ArrivalTime)$	wa_5 $(Stops)$
	0.2229	0.2146	0.2312	0.1854	0.1459
consensus weightings for alternatives from QGNDD	wa_1 $(Cheap)$	wa_2 $(DepartureTime)$	wa_3 $(TravelTime)$	wa_4 $(ArrivalTime)$	wa_5 $(Stops)$
	0.2164	0.2127	0.2182	0.1936	0.1591

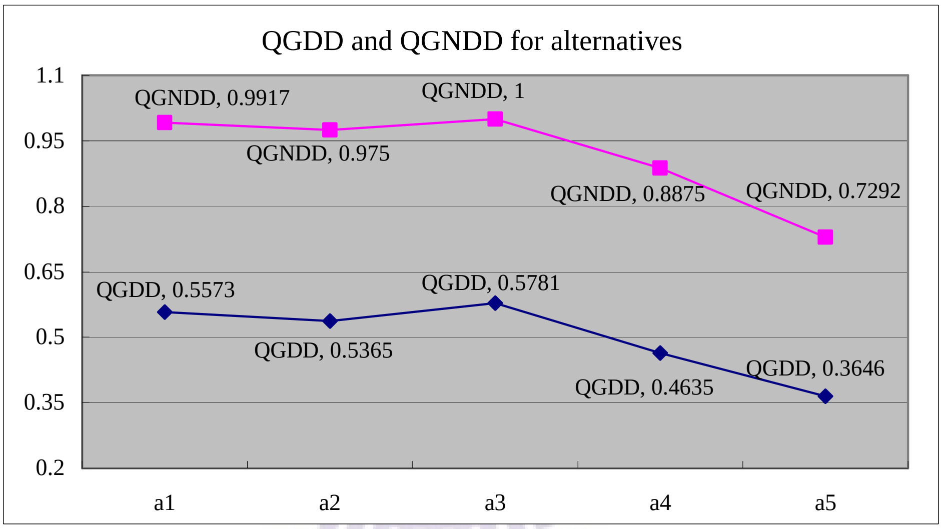

Figure 5-4　QGDD and QGNDD for alternatives

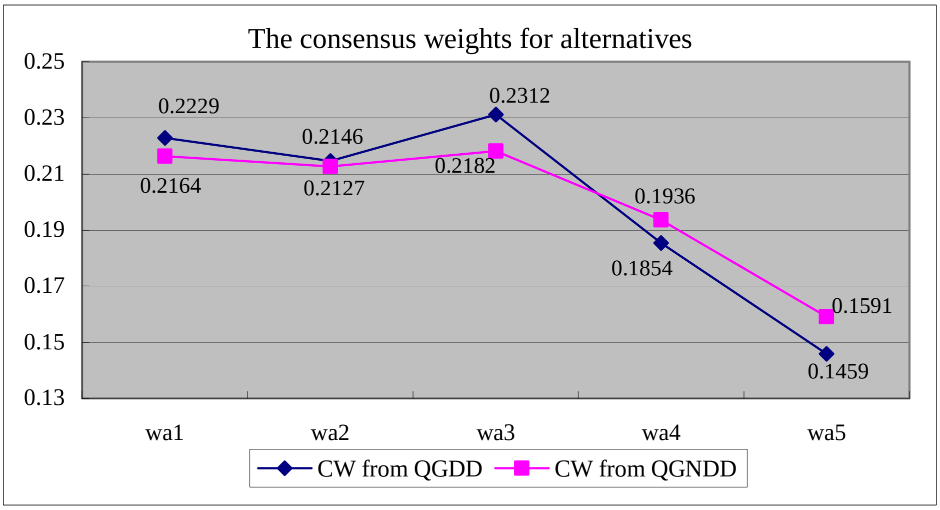

Figure 5-5　The consensus weightings (CW) for alternatives

In addition to identifying preference orderings, the value of QGDD and QGNDD can also be used to calculate the weightings for each alternative. The consensus weightings for alternatives that are derived from QGDD and QGNDD are given by $W_{QGDD} = (0.2229, 0.2146, 0.2312, 0.1854, 0.1459)$ and $W_{QGNDD} = (0.2164, 0.2127, 0.2182, 0.1936, 0.1591)$. That is, the consensus weightings for the primitive term *Cheap* is 0.2229 (derived from QGDD) and the composite term *Satisfaction*(Q) or $\tilde{Q}$ can be moderated as:

$$\tilde{Q} = 0.2229 \times \tilde{C} + 0.2146 \times \tilde{D} + 0.2312 \times \tilde{T} + 0.1854 \times \tilde{A} + 0.1459 \times \tilde{S}$$

5.2.2 Performance evaluation for Case II

This section describes the evaluation of the proposed approach. The evaluation is based on a case study that comprises thirty different service consumers and nine different airlines services from different service providers. The proposed approach Moderated Fuzzy Discovery Method (MFDM) is evaluated in comparison to Capability Discovery Method (CDM) and Fuzzy Discovery Method (FDM).

5.2.2.1 Capability Discovery Method (CDM)

The CDM method is a service discovery approach, which adopts the function or capability of the service as the main criterion for searching. In the first set of experiments, CDM was adopted without involving any fuzzy discovery and higher level abstraction mechanisms (quality rating). In this method, the capability matchmaker suggests all the nine Web services to the consumers, as they satisfy the requirements in terms of capability constraints. This method is therefore inappropriate as each consumer has to interrogate the data repositories of each service in order to discover the required service. Table 5-9 to Table 5-13 illustrate the fuzzy sets for the service consumers appeared in Case II. Table 5-17 shows the results related to the precision rate. The fuzzy set for Consumer 1, for example, is represented as follows:

$$\tilde{C}_1(a_1, b_1, c_1, d_1) = (0,0,450,600)$$
$$\tilde{D}_1(a_1, b_1, c_1, d_1) = (540,660,960,1080)$$
$$\tilde{T}_1(a_1, b_1, c_1, d_1) = (0,0,1350,1950)$$
$$\tilde{A}_1(a_1, b_1, c_1, d_1) = (540,660,960,1080)$$
$$\tilde{S}_1(a_1, b_1, c_1, d_1) = (0,0,1,2)$$

Table 5-17 CDM precision rates for Consumer 1, 2, 29, and 30

CDM Suggestions (No filtering)	C1	C2	…	C29	C30
AlitaliaAir	∨		…	∨	
BritishAir			…		
Cathay PacificAir	∨	∨	…	∨	∨
EvaAir			…		∨
KlmRoyal DutchAir	∨		…	∨	∨
KoreanAir			…		
MalaysianAir			…		
SingaporeAir	∨	∨	…	∨	∨
ThaiAir	∨	∨	…	∨	
Precision Rate for Specific Consumer	5 / 9 = 0.5556	3 / 9 = 0.3333	…	5 / 9 = 0.5556	4 / 9 = 0.4444

$\tilde{C}_1(a_1, b_1, c_1, d_1) = (0,0,450,600)$ means that Consumer 1 has a subjective opinion on the price for a flight (between 0 and 600 (GBP)). Consider the other primitive terms, $\tilde{D}_1, \tilde{T}_1, \tilde{A}_1, \tilde{S}_1$, only five services can meet Consumer 1's requirements. So the precision rate is 55.66% (5 / 9 = 0.5556). The same principle can be applied to other service consumers in order to evaluate the precision rates. Table 5-17 shows the derived precision rates 0.5556, 0.3333, 0.5556, and 0.4444 for the Consumer 1, 2, 29, and 30, respectively. Again, precision rate for the other consumers are omitted for brevity.

5.2.2.2 Fuzzy Discovery Method (FDM)

The second set of experiments is carried out to test FDM [17]. FDM was deployed after the fuzzy classification has been conducted on the underlying data about each service. In this experiment, *Fuzzy Classifier* adopts the initial composite inference rule, $\tilde{Q}_{init}$ = 1/5 × $\tilde{C}_{init}$ + 1/5 × $\tilde{D}_{init}$ + 1/5 × $\tilde{T}_{init}$ + 1/5 × $\tilde{A}_{init}$ + 1/5 × $\tilde{S}_{init}$, to calculate the value for composite term *Satisfaction* according to the actual content of a specific flight. Before FDM is applied for service discovery, each of the ten services will be rated by $\tilde{Q}_{init}$ and therefore each service gets a value representing its higher level informative declaration (quality rating or QoS) on the composite term *Satisfaction*. The classification results are shown in Table 5-18.

Table 5-18 Classification results for each service with *Satisfaction*, $\tilde{Q}_{init}$, with equal weightings

Service	QoS Value for *Satisfaction*	Service	QoS Value for *Satisfaction*
AlitaliaAir	0.6*	KoreanAir	0.5*
BritishAir	0.76*	MalaysianAir	0.86*
Cathay PacificAir	0.8*	SingaporeAir	0.82*
EvaAir	0.65*	ThaiAir	0.65*
KlmRoyal DutchAir	0.83*	* added when value >= θ	

Suppose that the threshold θ = 0.45 is adopted for all consumers. θ, the threshold, is used in the *Fuzzy Discovery* to filter out those services with less possibility to meet the requirement. In this experiment, *Fuzzy Discovery* recommends all nine Web services. From the information presented in Table 5-9 to Table 5-13, Consumer 2 has the following preferences for *Satisfaction*:

$$\tilde{C}_2(a_2, b_2, c_2, d_2) = (0,0,500,650)$$
$$\tilde{D}_2(a_2, b_2, c_2, d_2) = (420,540,900,1020)$$
$$\tilde{T}_2(a_2, b_2, c_2, d_2) = (0,0,1170,1470)$$
$$\tilde{A}_2(a_2, b_2, c_2, d_2) = (480,600,1080,1200)$$
$$\tilde{S}_2(a_2, b_2, c_2, d_2) = (0,0,0,2)$$

$\tilde{C}_2(a_2, b_2, c_2, d_2) = (0,0,500,650)$ indicates that Consumer 2 has the subjective opinion on *Cheap* which lies between 0 and 650 (GBP), *TravelTime* lies between 0 and 1470 minutes, and *Stops* is between 0 to 2 stop. Consider the all the primitive terms, $\tilde{C}_2, \tilde{D}_2, \tilde{T}_2, \tilde{A}_2, \tilde{S}_2$, only three airline services can satisfy the consumer's requirements, and therefore the precision rate for Consumer 2 is 33.33% (3 / 9 = 0.3333). The same principle is applicable to other consumers. Table 5-19 illustrates that Consumer 1, 2, 29, and 30 gain values of 55.56%, 33.33%, 55.56%, and 44.44% respectively for their precision rates. The precision rates for the other consumers are omitted for the sake of brevity.

Table 5-19 FDM precision rates for Consumer 1, 2, 29, and 30 with θ = 0.45

θ = 0.45 FDM Suggestions	C1	C2	…	C29	C30
AlitaliaAir	V		…	V	
BritishAir			…		
Cathay PacificAir	V	V	…	V	V
EvaAir			…		V
KlmRoyal DutchAir	V		…	V	V
KoreanAir			…		
MalaysianAir			…		
SingaporeAir	V	V	…	V	V
ThaiAir	V	V	…	V	
Precision Rate for Specific Consumer	5 / 9 = 0.5556	3 / 9 = 0.3333	…	5 / 9 = 0.5556	4 / 9 = 0.4444

5.2.2.3 Moderated Fuzzy Discovery Method (MFDM)

The third set of experiments is conducted to test MFDM. These experiments first employ SAM to gain the consensus from the distinct opinions on the specific primitive terms ($\tilde{C}$, $\tilde{D}$, $\tilde{T}$, $\tilde{A}$, and $\tilde{S}$) and then employ RMGDP to obtain the group preferences on the different selection criteria which are the ingredients of the composite term (Section 5.2.1). MFDM was deployed after the fuzzy classification has been conducted on the underlying data about each service. In this experiment, *Fuzzy Classifier* first replaces the initial primitive terms and it adopts the moderated primitive terms with equal weightings to calculate the value for composite term *Satisfaction* according to the actual content of a specific flight. According to Section 5.2.1, this less subjective inference rule is as follow:

$\tilde{Q}$ = 1/5 × $\tilde{C}$ + 1/5 × $\tilde{D}$ + 1/5 × $\tilde{T}$ + 1/5 × $\tilde{A}$ + 1/5 × $\tilde{S}$, where

$\tilde{C} = (0,0,596.1289,778.4472)$,

$\tilde{D} = (500.906,623.325,770.581,890.581)$,

$\tilde{T} = (0,0,1388.56,1580.58)$,

$\tilde{A} = (621.255,741.255,994.676,1064.68)$, and

$\tilde{S} = (0,0,0.95,1.95)$.

Before MFDM is applied for service discovery, each of the nine services will be rated by the moderated $\tilde{Q}$ and therefore each service gets a value representing its higher level informative declaration (quality rating or QoS) related to the composite term *Satisfaction*. The classification results are illustrated in Table 5-20.

Table 5-20 Classification results for each service with moderated *Satisfaction*, $\tilde{Q}$, with equal weightings

Service	QoS Value for *Satisfaction*	Service	QoS Value for *Satisfaction*
AlitaliaAir	0.45*	KoreanAir	0.36
BritishAir	0.48*	MalaysianAir	0.59*
Cathay PacificAir	0.71*	SingaporeAir	0.57*
EvaAir	0.58*	ThaiAir	0.52*
KlmRoyal DutchAir	0.25	* added when value >= θ	

Suppose that the threshold θ = 0.45 is adopted for all consumers. θ, the threshold, is used in the *Fuzzy Discovery* to filter out those services with less likelihood of meeting the requirement. In this experiment, *Fuzzy Discovery* recommends only seven services which are satisfactory, that is, AlitaliaAir, BritishAir, CathayPacificAir, EvaAir, MalaysianAir, SingaporeAir, and ThaiAir. According to the information presented in Table 5-9 to Table 5-13, Consumer 29 has the following preferences for *Satisfaction*:

$$\tilde{C}_{29}(a_{29}, b_{29}, c_{29}, d_{29}) = (0,0,700,800)$$
$$\tilde{D}_{29}(a_{29}, b_{29}, c_{29}, d_{29}) = (540,660,720,840)$$
$$\tilde{T}_{29}(a_{29}, b_{29}, c_{29}, d_{29}) = (0,0,1470,1710)$$
$$\tilde{A}_{29}(a_{29}, b_{29}, c_{29}, d_{29}) = (420,549,600,720)$$
$$\tilde{S}_{29}(a_{29}, b_{29}, c_{29}, d_{29}) = (0,0,1,2)$$

The above reveals that Consumer 29 has a subjective opinion on *Cheap* which sits between 0 and 800 (GBP), *TravelTime* which lies between 0 and 1710 minutes, and *Stops* rests between 0 to 2 stops. Consider the all the primitive terms, $\tilde{C}_{29}, \tilde{D}_{29}, \tilde{T}_{29}, \tilde{A}_{29}, \tilde{S}_{29}$, only four airline services can satisfy the consumer's requirements. However, the precision rate for Consumer 29 has increased to 57.14% (4 / 7 = 0.5714), due to the contribution of the moderation. Table 5-21 shows service Consumers 1, 2, 29 and 30 obtain their precision rates 57.14%, 42.86%, 57.14% and 42.86% respectively by employing the MFDM.

Table 5-21　MFDM precision rates for Consumer 1, 2, 29, and 30 with θ = 0.45 (with equal weightings)

θ = 0.45 MFDM (equal weightings) Suggestions	C1	C2	…	C29	C30
AlitaliaAir	V		…	V	
BritishAir			…		
Cathay PacificAir	V	V	…	V	V
EvaAir			…		V
MalaysianAir			…		
SingaporeAir	V	V	…	V	V
ThaiAir	V	V	…	V	
Precision Rate for Specific Consumer	4 / 7 = 0.5714	3 / 7 = 0.4286	…	4 / 7 = 0.5714	3 / 7 = 0.4286

After the completion of the SAM process, the RMGDP process is applied to acquire the consensus weightings for the predefined five criteria. According to Section 5.2.1, the consensus weightings derived from QGDD is $W_{QGDD} = (0.2229, 0.2146, 0.2312, 0.1854, 0.1459)$. Therefore, the composite term *Satisfaction* with consensus weightings is as follow:

$$\tilde{Q} = 0.2229 \times \tilde{C} + 0.2146 \times \tilde{D} + 0.2312 \times \tilde{T} + 0.1854 \times \tilde{A} + 0.1459 \times \tilde{S}$$

This new moderated $\tilde{Q}$ will be employed for fuzzy classification in order to obtain new classification results on *Satisfaction*.　Table 5-22 illustrates the classification results.

Table 5-22　Classification results for each service with moderated *Satisfaction*, $\tilde{Q}$, with consensus weightings

Service	QoS Value for *Satisfaction*	Service	QoS Value for *Satisfaction*
AlitaliaAir	0.48*	KoreanAir	0.33
BritishAir	0.45*	MalaysianAir	0.57*
Cathay PacificAir	0.72*	SingaporeAir	0.55*
EvaAir	0.58*	ThaiAir	0.5*
KlmRoyal DutchAir	0.27	* added when value >= θ	

If θ = 0.45 is also adopted for all consumers. *Fuzzy Discovery* will recommends only seven satisfactorily services, that is, AlitaliaAir, BritishAir, CathayPacificAir, EvaAir, MalaysianAir, SingaporeAir, and ThaiAir. According to the information presented in Table 5-9 to Table 5-13, Consumer 30 has the following preferences for *Satisfaction*:

$$\tilde{C}_{30}(a_{30}, b_{30}, c_{30}, d_{30}) = (0,0,600,700)$$
$$\tilde{D}_{30}(a_{30}, b_{30}, c_{30}, d_{30}) = (900,1020,1140,1260)$$
$$\tilde{T}_{30}(a_{30}, b_{30}, c_{30}, d_{30}) = (0,0,1410,1650)$$
$$\tilde{A}_{30}(a_{30}, b_{30}, c_{30}, d_{30}) = (900,1020,1140,1260)$$
$$\tilde{S}_{30}(a_{30}, b_{30}, c_{30}, d_{30}) = (0,0,1,2)$$

It shows that Consumer 30 has a subjective opinion on *Cheap* which sits between 0 and 700 (GBP), *TravelTime* which lies between 0 and 1650 minutes, and *Stops* rests between 0 to 2 stops. Considering the all the primitive terms, only three airline services can satisfy the consumer’s requirements and the precision rate for Consumer 30 is 42.86% (3 / 7 = 0.4286). Table 5-23 shows service Consumers 1, 2, 29 and 30 obtain their precision rates 57.14%, 42.86%, 57.14% and 42.86% respectively by employing MFDM.

Table 5-23 MFDM precision rates for Consumer 1, 2, 29, and 30 with θ = 0.45 (with consensus weightings)

θ = 0.45 MFDM (consensus weightings) Suggestions	C1	C2	…	C29	C30
AlitaliaAir	V		…	V	
BritishAir			…		
Cathay PacificAir	V	V	…	V	V
EvaAir			…		V
MalaysianAir			…		
SingaporeAir	V	V	…	V	V
ThaiAir	V	V	…	V	
Precision Rate for Specific Consumer	4 / 7 = 0.5714	3 / 7 = 0.4286	…	4 / 7 = 0.5714	3 / 7 = 0.4286

5.2.2.4 Summary of Case II

Table 5-24 shows an integrated view of Table 5-17, Table 5-19, Table 5-21 and Table 5-23. It shows the average precision rate for Capability Discovery Method, Fuzzy Discovery Method, Moderated Fuzzy Discovery Method (with equal weightings) and Moderated Fuzzy Discovery Method (with consensus weightings).

Table 5-24 Average precision rate for CDM, FDM and MFDM with θ = 0.45

Precision Rate for Specific Consumer	# of Suggestions	C1	C2	…	C29	C30	Average Precision Rate
CDM	9	0.5556	0.3333	…	0.5556	0.4444	0.3926
FDM	9	0.5556	0.3333	…	0.5556	0.4444	0.3926
MFDM (equal weightings)	7	0.5714	0.4286	…	0.5714	0.4286	0.4476
MFDM (consensus weightings)	7	0.5714	0.4286	…	0.5714	0.4286	0.4476

From Table 5-24, it is observed that the proposed MFDM has outperformed CDM and FDM. With the derived consensus weightings, it also produces better average precision rates (i.e., 5.5%) than CDM and FDM and the number of suggested services has been reduced by two. Note that the average precision rate for FDM is identical to the rate for CDM. This is because that both of FDM and CDM have the same number of recommended services when θ = 0.45. In addition, MFDM with equal weightings and MFDM with consensus weightings have the same performance when θ = 0.45.

For any value chosen for the threshold (θ) lying between 0.45 and 0.5, MFDM also produces better average precision rates (5.5% ~ 12.41%) than CDM and FDM as shown in Table 5-25. In this case, MFDM with consensus weightings suggests only six services and it produces even better precision rate than the MFDM with the equal weightings (6.91%).

Table 5-25 Average precision rate for CDM, FDM and MFDM when 0.45< θ < 0.5

Precision Rate for Specific Consumer	# of Suggestions	C1	C2	…	C29	C30	Average Precision Rate
CDM	9	0.5556	0.3333	…	0.5556	0.4444	0.3926
FDM	9	0.5556	0.3333	…	0.5556	0.4444	0.3926
MFDM (equal weightings)	7	0.5714	0.4286	…	0.5714	0.4286	0.4476
MFDM (consensus weightings)	6	0.6667	0.5	…	0.6667	0.5	0.5167

The average precision rates with the other thresholds, θ = 0.5 or 0.55, are shown in Table 5-26 and Table 5-27 which also demonstrate that MFDM is able to produce better results than CDM and FDM and the number of suggested services has been greatly reduced by four and five.

Table 5-26 Average precision rate for CDM, FDM and MFDM with θ = 0.5

Precision Rate for Specific Consumer	# of Suggestions	C1	C2	…	C29	C30	Average Precision Rate
CDM	9	0.5556	0.3333	…	0.5556	0.4444	0.3926
FDM	9	0.5556	0.3333	…	0.5556	0.4444	0.3926
MFDM (equal weightings)	5	0.6	0.6	…	0.6	0.8	0.5533
MFDM (consensus weightings)	5	0.6	0.6	…	0.6	0.8	0.5533

Table 5-27 Average precision rate for CDM, FDM and MFDM with θ = 0.55

Precision Rate for Specific Consumer	# of Suggestions	C1	C2	…	C29	C30	Average Precision Rate
CDM	9	0.5556	0.3333	…	0.5556	0.4444	0.3926
FDM	8	0.625	0.375	…	0.625	0.5	0.4417
MFDM (equal weightings)	4	0.5	0.5	…	0.5	0.75	0.475
MFDM (consensus weightings)	4	0.5	0.5	…	0.5	0.75	0.475

In conclusion, MFDM has produced a higher average precision rate than CDM by 5.5%~16% and FDM by 3~16% with various thresholds. In Case II, moreover, MFDM with consensus weightings can perform a better average precision rate than MFDM with equal weightings (at least the same rate). The reason why the average precision rate of MFDM is not increased dramatically is that the initial values for *Satisfaction* were set by an experienced person and these values are close to the consensus values. Even though it does increase the average precision rate by 3%~16%, it is observed that the number of recommended services is significantly reduced (by 22%~55%). In other words, with the provision of MFDM, service consumers do not have to obey the advertisements preset by the service providers and they are able to use more objective values to eliminate unnecessary consideration of details and increase the precision rate of locating the required services at the same time.

Instead of using a threshold, the discovery mechanism could only suggests those services which best match the consumers' requirements. In other words, the discovery mechanism only highlights those services with the most significant *Satisfaction* values (best choice). Under this scenario, FDM recommends airline MalaysianAir (*Satisfaction* value 0.86), MFDM (with equal weightings) recommended airline CathayPacificAir (*Satisfaction* value 0.71) and MFDM (with consensus weightings) suggested airline CathayPacificAir (*Satisfaction* value 0.72). Table 5-28 shows that the average precision rates have dramatically increased to 90%. This also resulted from the effectiveness of moderation process.

Table 5-28 Average precision rate for CDM, FDM and MFDM with best choice

Precision Rate for Specific Consumer	# of Suggestions	C1	C2	…	C29	C30	Average Precision Rate
CDM	9	0.5556	0.3333	…	0.5556	0.4444	0.3926
FDM	1	0	0	…	0	0	0
MFDM (equal weightings)	1	1	1	…	1	1	0.9
MFDM (consensus weightings)	1	1	1	…	1	1	0.9

5.3 Case III - Flight booking case study with numerous criteria

In Case II, RMGDP sub-process can be used to obtain the consensus weightings on different selection criteria which are the ingredients of a composite term. It is assumed that consumers' preferences between various criteria are collected by a popular method – "Preference Ordering / PO" which requires users to provide their preference over different criteria in a precise sequence (complete order). Sometimes, however, users are not confident with their preferences and it is difficult to have consumers provide the complete order for alternatives to which they are indifferent or they find indistinguishable. For example, some users cannot distinguish the relative importance of *Cheap* and *Comfortable*.

In addition, to have too many insignificant criteria is harmful to the system performance. It is good to find out what the most important (top-N) alternatives are when numerous criteria exist. System complexity can be reduced by limiting the number of alternatives and the performance can then be increased. Once the top-N alternatives are produced, the RMGDP process, described in Section 3.3, can be adopted to resolve the quantified consensus weightings for these important alternatives.

In Case III, Pseudo-Order Preference Model (POPM), described in Section 3.4, is introduced to help consumers to collect preferences on indifferent or indistinguishable alternatives and it also helps to reduce the system complexity by selecting only the top-N alternatives for moderation.

5.3.1 Scenario and the moderation process for Case III

In Case III, the Pseudo-Order Preference Model (POPM) is introduced to collect consumers' preferences in the situation which does not require the users to express their preference for alternatives in complete order. The scenario of Case III is also based on searching an appropriate flight booking service. Four consumers and ten airline services are included in Case III. Consumers have their different subjective opinions on the ingredients of the composite term *Satisfaction*. In Case III, six primitive terms comprise the composite term, *Satisfaction*, to represent the overall quality of a flight ticket. These primitive terms, or alternatives, are *Cheap*(a_1), *MultimediaEquipment*(a_2), *Food*(a_3), *Airtime*(a_4), *Seatsize*(a_5), and *FlightServiceOfCrew*(a_6). It is assumed that consumers in Case III cannot easily distinguish the relative importance of some of the alternatives and only the top-3 alternatives are required, hence gaining better performance by reducing the system complexity.

First, POPM can be adopted to prioritize the order of various alternatives by identifying the relatively most important criteria (top-3) accepted by the four consumers in order to filter out the remaining three less significant criteria. For each consumer *k*, his / her preference over these six alternatives is not collected (transformed) from a complete order of the sorted alternatives. Instead, it is gathered pair by pair by using preference relations (p_{ij}^k) [66],[68],[75], as follows:

$$p_{ij}^1 = \begin{bmatrix} 0.5 & 0.6 & 0.8 & 0.3 & 0.4 & 0.7 \\ 0.4 & 0.5 & 0.7 & 0.2 & 0.3 & 0.6 \\ 0.2 & 0.3 & 0.5 & 0 & 0.1 & 0.4 \\ 0.7 & 0.8 & 1 & 0.5 & 0.4 & 0.9 \\ 0.6 & 0.7 & 0.9 & 0.6 & 0.5 & 0.8 \\ 0.3 & 0.4 & 0.6 & 0.1 & 0.2 & 0.5 \end{bmatrix}, p_{ij}^2 = \begin{bmatrix} 0.5 & 0.8 & 0.9 & 0.6 & 0.7 & 1 \\ 0.2 & 0.5 & 0.6 & 0.3 & 0.4 & 0.7 \\ 0.1 & 0.4 & 0.5 & 0.2 & 0.3 & 0.6 \\ 0.4 & 0.7 & 0.8 & 0.5 & 0.6 & 0.9 \\ 0.3 & 0.6 & 0.7 & 0.4 & 0.5 & 0.8 \\ 0 & 0.3 & 0.4 & 0.1 & 0.2 & 0.5 \end{bmatrix},$$

$$p_{ij}^3 = \begin{bmatrix} 0.5 & 0.9 & 0.8 & 0.6 & 0.7 & 1 \\ 0.1 & 0.5 & 0.4 & 0.2 & 0.3 & 0.6 \\ 0.2 & 0.6 & 0.5 & 0.3 & 0.4 & 0.7 \\ 0.4 & 0.8 & 0.7 & 0.5 & 0.4 & 0.9 \\ 0.3 & 0.7 & 0.6 & 0.6 & 0.5 & 0.8 \\ 0 & 0.4 & 0.3 & 0.1 & 0.2 & 0.5 \end{bmatrix}, p_{ij}^4 = \begin{bmatrix} 0.5 & 0.8 & 0.9 & 0.4 & 0.6 & 0.7 \\ 0.2 & 0.5 & 0.6 & 0.1 & 0.3 & 0.4 \\ 0.1 & 0.4 & 0.5 & 0 & 0.2 & 0.3 \\ 0.6 & 0.9 & 1 & 0.5 & 0.7 & 0.8 \\ 0.4 & 0.7 & 0.8 & 0.3 & 0.5 & 0.6 \\ 0.3 & 0.6 & 0.7 & 0.2 & 0.4 & 0.5 \end{bmatrix}.$$

In fuzzy preference relations, the importance of alternatives is collected pair by pair. For example, $p_{16}^{2} = p_{16}^{3} = 1$ means that both $User_2$ and $User_3$ completely prefer a_1 to a_6. $p_{51}^{1} = 0.6$ and $p_{15}^{1} = 0.4$ indicate that $User_1$ slightly prefer a_5 to a_1.

Each consumer possesses a fuzzy preference relation and all fuzzy preference relations can be aggregated to calculate the collective preference relation (p_{ij}^{c}) by Equation (8) or (17) (Section 3.3.2, Section 3.4). In this experiment, the linguistic quantifier *'most'* with pair [0.3, 0.8] is applied to conclude that "most of the consumers agree to six alternatives / criteria for the flight booking service discovery". The corresponding OWA operator with the weighting vector for *'most'* will be $w = (0.00, 0.40, 0.50, 0.10)$ and the p_{ij}^{c} is as follows:

$$p_{ij}^{c} = \begin{bmatrix} 0.5 & 0.78 & 0.84 & 0.47 & 0.62 & 0.82 \\ 0.19 & 0.5 & 0.58 & 0.19 & 0.3 & 0.58 \\ 0.14 & 0.39 & 0.5 & 0.08 & 0.23 & 0.47 \\ 0.48 & 0.79 & 0.87 & 0.5 & 0.48 & 0.89 \\ 0.34 & 0.69 & 0.73 & 0.47 & 0.5 & 0.78 \\ 0.12 & 0.39 & 0.47 & 0.1 & 0.2 & 0.5 \end{bmatrix}$$

By applying various thresholds (q=0.1~0.9) and Equations (18~19), the distinct preference order for six alternatives can be derived, as shown in Table 5-29. Finally, the grouped preference order for the six primitive terms is determined when a common indifference threshold is applied [72]. To identify a distinct top-3 it can be seen from the following table that q = 0.3, 0.4 or 0.5 will provide such a division. In these cases {*Cheap*(a_1), *Airtime*(a_4), *Seatsize*(a_5)} > {*MultimediaEquipment*(a_2), *Food*(a_3), *FlightServiceOfCrew*(a_6)}.

Table 5-29 Derived distinct preference orders for six alternatives with various thresholds

Indifference Threshold	Derived Preference Order
q=0.1	$\{a_1, a_4\} > \{a_5\} > \{a_2\} > \{a_3, a_6\}$
q=0.2	$\{a_1, a_4\} > \{a_5\} > \{a_2, a_3, a_6\}$
q=0.3	$\{a_1, a_4, a_5\} > \{a_2, a_3, a_6\}$
q=0.4	$\{a_1, a_4, a_5\} > \{a_2, a_3, a_6\}$
q=0.5	$\{a_1, a_4, a_5\} > \{a_2, a_3, a_6\}$
q=0.6	$\{a_1, a_4, a_5, a_2\} > \{a_3, a_6\}$
q=0.7	$\{a_1, a_4, a_5, a_2\} > \{a_3, a_6\}$
q=0.8	$\{a_1, a_4, a_5, a_2, a_3, a_6\}$
q=0.9	$\{a_1, a_4, a_5, a_2, a_3, a_6\}$

According to Table 5-29, the top-3 alternatives (a_1, a_4, a_5) are selected to be the most important primitive terms which are used in *Fuzzy Moderator* for calculating the consensus weightings. These three primitive terms will be denoted as *Cheap*($\tilde{C}$), *Airtime*($\tilde{T}$), and *Seatsize*($\tilde{S}$). The four consumers should bring themselves to an agreement over the definition of these three primitive terms ($\tilde{C}, \tilde{T}, \tilde{S}$). For instance, *Cheap* was initially defined as $\tilde{C}_{init} = (0,0,14500,16500)$. However, the four consumers have different views on these definitions which are formulated as the following fuzzy sets:

$$\tilde{C}_1(a_1, b_1, c_1, d_1) = (0,0,13500,16500),$$
$$\tilde{C}_2(a_2, b_2, c_2, d_2) = (0,0,14500,14500),$$
$$\tilde{C}_3(a_3, b_3, c_3, d_3) = (0,0,14000,15500),$$
$$\tilde{C}_4(a_4, b_4, c_4, d_4) = (0,0,11000,13000)$$
$$\tilde{S}_1(a_1, b_1, c_1, d_1) = (1.5,2,2.5,2.5),$$
$$\tilde{S}_2(a_2, b_2, c_2, d_2) = (1,1,2,2),$$
$$\tilde{S}_3(a_3, b_3, c_3, d_3) = (0.8,1,2,3),$$
$$\tilde{S}_4(a_4, b_4, c_4, d_4) = (0.6,1.2,1.8,2.5),$$
$$\tilde{T}_1(a_1, b_1, c_1, d_1) = (0,0,2.5,2.5),$$
$$\tilde{T}_2(a_2, b_2, c_2, d_2) = (0,0,2.5,3.5),$$
$$\tilde{T}_3(a_3, b_3, c_3, d_3) = (0,0,1.8,2.8),$$
$$\tilde{T}_4(a_4, b_4, c_4, d_4) = (0,0,1.5,3)$$

In the following, Similarity Aggregation Method (SAM) is applied to calculate the consensus value for *Cheap*($\tilde{C}$), *Airtime*($\tilde{T}$), and *Seatsize*($\tilde{S}$). After the application of SAM, the initial subjective value, $\tilde{C}_{init} = (0,0,14500,16500)$, which was given for the *Fuzzy Classifier* to carry out reasoning has been modified to the new derived consensus value: $\tilde{C} = (0, 0, 13314.333, 14925.007)$. The same principle is applicable to the other two primitive fuzzy terms *Airtime*($\tilde{T}$) and *Seatsize*($\tilde{S}$), and therefore their consensus values are $\tilde{T} = (0,0,2.0666,2.9379)$ and $\tilde{S} = (0.8911,1.2008,2.0105,2.5181)$ respectively. For more detailed procedure, please refer to Section 5.1.1.

When the preference order and consensus value of the three primitive terms have been resolved, it is able to adopt the RMGDP process (Section 3.3) to carry out transformation, aggregation, and exploitation processes in order to reach a consensus on the weightings of criteria which comprise the composite terms *Satisfaction*.

Assume that each consumer provides his / her preferences on a list of alternatives, $A = \{a_1, a_5, a_4\}$ where a_1 is *Cheap*, a_4 is *Airtime*, and a_5 is *Seatsize*, using a preference ordering $O^k = \{o_1^k, o_2^k, .., o_m^k\}$ (*m* is the number of alternatives). For instance, each consumer *k*, denoted as $User_k\,(k = 1,2,3,4)$, provides his / her preferences on alternatives by the following preference ordering $O^1 = \{3,12\}$, $O^2 = \{1,3,2\}$, $O^3 = \{1,2,3\}$ and $O^4 = \{2,3,1\}$. For any two ordering preference values, o_i^k, o_j^k, assessed by $User_k$, a difference-scale transformation function, p_{ij}^k in Equation (7), shows that $User_k$ has a subjective ordering preference of the alternative a_i over alternative a_j. For each consumer, the preference ordering $O^k = \{o_1^k, o_2^k, .., o_m^k\}$ can be transformed into fuzzy preference relation (p_{ij}^{k}) as follows:

$$p_{ij}^{1}=\begin{bmatrix}0.50 & 0.75 & 0.25\\ 0.25 & 0.50 & 0.00\\ 0.75 & 1.00 & 0.50\end{bmatrix},\quad p_{ij}^{2}=\begin{bmatrix}0.50 & 1.00 & 0.75\\ 0.00 & 0.50 & 0.25\\ 0.25 & 0.75 & 0.50\end{bmatrix},$$

$$p_{ij}^{3}=\begin{bmatrix}0.50 & 0.75 & 1.00\\ 0.25 & 0.50 & 0.75\\ 0.00 & 0.25 & 0.50\end{bmatrix},\quad p_{ij}^{4}=\begin{bmatrix}0.50 & 0 & 0.25\\ 1.00 & 0.50 & 0.75\\ 0.75 & 025 & 0.5\end{bmatrix}.$$

After transforming preference orderings into fuzzy preference relations, the collective preference relation (p_{ij}^{c}) can be calculated by Equation (8). In this case, consumers' opinions are treated on an equal basis so that the corresponding OWA operator with the weighting vector will be w = (1/4, 1/4, 1/4, 1/4) and p_{ij}^{c} is as follows:

$$p_{ij}^{c}=\begin{bmatrix}0.500 & 0.625 & 0.563\\ 0.375 & 0.500 & 0.437\\ 0.437 & 0.563 & 0.500\end{bmatrix}$$

Moreover, Quantifier Guided Non-Dominance Degree (QGNDD) and Quantifier Guided Dominance Degree (QGDD) could be obtained by using Equation (10, 11), and the consensus of four consumers is reached as shown in Table 5-30. In the exploitation process, the derived values of QGDD and QGNDD are used to determine the complete order and weightings for each alternative. The complete order of primitive terms is the same as the preference order of the service criteria. The consensus weightings, as shown in Table 5-30, for alternatives derived from QGDD and QGNDD are formulated as $W_{QGDD}=(0.3959, 0.3333, 0.2708)$ and $W_{QGNDD}=(0.3636, 0.3409, 0.2955)$, that is, the consensus weighting for *Cheap* is 0.3959 when consensus weighting from QGDD is adopted. Thus, the initial composite term, *Satisfaction*, with new consensus weightings can be moderated as:

$$\tilde{Q} = 0.3959 \times \tilde{C} + 0.3333 \times \tilde{T} + 0.2708 \times \tilde{S}$$

Table 5-30 QGDD, QGNDD and the consensus weightings for alternatives

QGDD for Alternatives	a_1 (*Cheap*)	a_4 (*Airtime*)	a_5 (*Seatsize*)	Consensus Weightings for Alternatives from QGDD	wa_1 (*Cheap*)	wa_4 (*Airtime*)	wa_5 (*Seatsize*)
	0.5938	0.5000	0.4063		0.3959	0.3333	0.2708
QGNDD for Alternatives	a_1 (*Cheap*)	a_4 (*Airtime*)	a_5 (*Seatsize*)	Consensus Weightings for Alternatives from QGNDD	wa_1 (*Cheap*)	wa_4 (*Airtime*)	wa_5 (*Seatsize*)
	1.000	0.9375	0.8125		0.3636	0.3409	0.2955

5.3.2 Performance evaluation for Case III

To examine the performance when numerous criteria are involved in service discovery, a case study with four different service consumers and ten airline service was adopted to evaluate three different methods namely Capability Discovery Method (CDM), Fuzzy Discovery Method (FDM), and Moderated Fuzzy Discovery Method (MFDM). Three different sets of experiments were carried out in order to gain their average precision rates and to examine their average performance.

In the first set of experiments, Capability Discovery Method is used without involving any pre-classification mechanism. The CDM suggests all the ten Web services to the consumers, since all of them satisfy the requirements in terms of capability constraints. So, the Web service consumers have to interrogate the data repositories to discover the required service. In one instance, Consumer 1's fuzzy set for *Cheap* is ($\tilde{C}_1(a_1, b_1, c_1, d_1)$ =(0, 0, 13500, 16500)). It means that Consumer 1 has a subjective opinion on the price which is between 0 and 16500. $\tilde{S}_1(a_1, b_1, c_1, d_1)$ = (1.5, 2, 2.5, 2.5) and $\tilde{T}_1(a_1, b_1, c_1, d_1)$ = (0, 0, 2.5, 2.5) are the Consumer 1's subjective opinion on *Seatsize* and *Airtime*. There are only five airline Web services which can meet Consumer 1's requirement. So the precision rate is 50% (5 / 10 = 0.5). In the same round, the service Consumers 2, 3 and 4 obtain different precisions 0.3, 0.7, and 0.1 respectively.

FDM with the same case study was used to carry out the next set of experiments. FDM was deployed after *Fuzzy Classifier* has conducted fuzzy classification on the data of each service. In one of experiments, the FDM only recommends six possible satisfactory services. Consumer 2 has a subjective opinion for *Satisfaction* in which $\tilde{C}_2(a_2, b_2, c_2, d_2)$ = (0, 0, 14500, 14500), $\tilde{S}_2(a_2, b_2, c_2, d_2)$ = (1, 1, 2, 2) and $\tilde{T}_2(a_2, b_2, c_2, d_2)$ = (0, 0, 2.5, 3.5). Its subjective *Cheap* price sits between 0 and 14500 dollars, *Seatsize* lies between 1 and 2 units and *Airtime* sits between 0 to 3.5 hours. Therefore, only two airline Web services can satisfy its requirement. For Consumer 2, the precision rate is 33.3% (2 / 6 = 0.333). In addition, Consumers 1, 3, and 4 gained 83.3%, 83.3% and 16.7% for the precision rates respectively.

For the last set of experiments, SAM, POPM and RMGDP are adopted altogether for MFDM to aggregate the group consensus on the composite term *Satisfaction* in order to produce less subjective inference rules. With the new derived inference rules, *Fuzzy Classifier* was able to gain new fuzzy values of *Satisfaction* and consequently MFDM only suggested three possible satisfactory services. For instance, Consumer 3 has inference rules for *Satisfaction* ($\tilde{C}_3(a_3, b_3, c_3, d_3)$ = (0, 0, 14000, 15500), $\tilde{S}_3(a_3, b_3, c_3, d_3)$ = (0.8, 1, 2, 3), $\tilde{T}_3(a_3, b_3, c_3, d_3)$ = (0, 0, 1.8, 2.8)) such that its subjective *Cheap* price sits between 0 and 15500 dollars, *Seatsize* lies between 0.8 and 3 units, and *Airtime* sits between 0 to 2.8 hours. So, only three airline services can satisfy Consumer 3's subjective opinion. However, the precision rate has been increased to 100% (3 / 3 = 1), due to the contribution of MFDM. Consumers 1, 2, and 4 have their precision rates 100%, 66.7%, and 33.3% respectively.

After all experiments have been carried out and the result of each experiment was recorded, each consumer's satisfaction rates with the recommended services were classified and averaged according to three different methods for the investigation of their precision rates. Figure 5-6 shows that the MFDM with 75% precision rate has the best performance. FDM has produced correct recommendations in just over half of the cases. CDM has only a 40% precision rate. It can be concluded that the proposed MFDM has outperformed FDM (by 20.8%), and FDM has produced better precision rate than CDM (by 14.2%), and MFDM has in this cases performed nearly twice as well (by 35%) as the CDM in terms of precision rate.

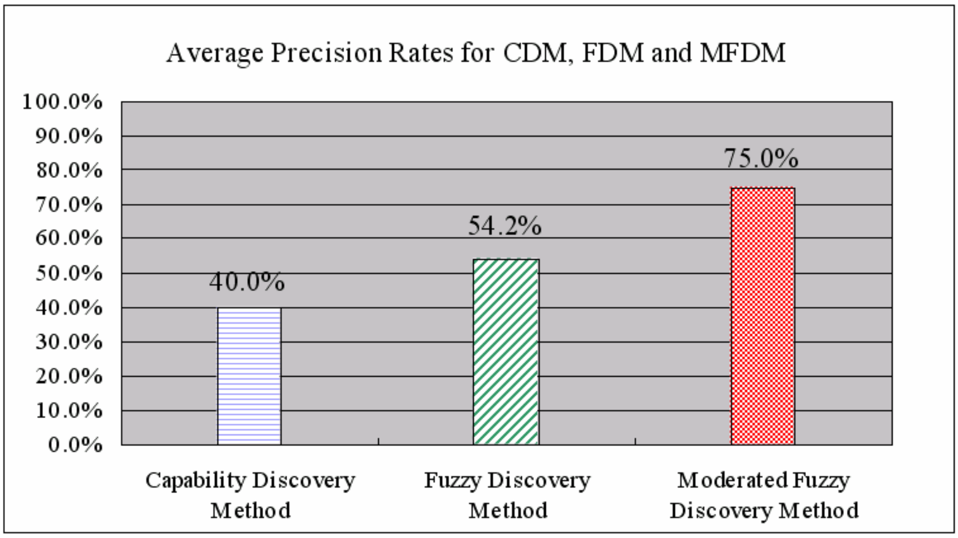

Figure 5-6 Average Precision Rates for CDM, FDM and MFDM

CHAPTER 6 CONCLUSION AND FUTURE WORKS

Service discovery is a critical process for the automation of Web service composition and utilization. The existing functional service discovery methods focus on search using service capabilities, interface signatures, or functionalities, but these methods have not paid sufficient attention to the use of underlying data and information on services as a search criterion. Since the issues associated with service discovery that involves the Web service having data repositories are not well addressed by the existing methods, these methods are inadequate to identify appropriate services among the services which have similar functionalities. It requires service consumers to include additional non-functional aspects (i.e. content of service or reputation) to evaluate the services.

Although there are a number of service discovery mechanisms based on the use of non-functional criteria to select appropriate services from a set of overlapping services which provide similar or identical functions, the aspects used for discrimination, such as such as *fees*, *security*, *privacy*, *time*, *availability* and *latency*, are technical viewpoints. These can be extended by considering the underlying data on services as a selection criterion. Moreover, current Web service discovery mechanisms are based on the search of service advertisements which are often provided by service providers. However, most of existing non-functional discovery methods do not address the issues associated with the impact of the diverse preferences and subjective expectations of service consumers and providers which are generally used in searching for or in advertising services.

In order to resolve the above issues, this research introduced a consensus-based service discovery approach which incorporates: the Semantic Web; fuzzy logic, and, group consensus methods, to improve the precision rate of identifying appropriate services. The main contribution of this work is that it presents a moderated service discovery mechanism which allows services to be discovered not only by the general business advertisements but also can be discovered according to a higher level abstraction (QoS) of its content using several different perspectives (different QoS terms).

Another contribution of this research is that the proposed method, Moderated Fuzzy Discovery Method (MFDM), provides a method to calculate the value of any QoS parameters from a group of users. Since service providers and consumers may have different perspectives on the selection criteria, the proposed MFDM provide the ability to mitigate this divergence by allowing providers and consumers to reach the consensus on the discovery criteria and the weightings of these criteria. MFDM is not proposed to replace most of existing discovery methods. It is complementary to them as it provides a way to calculate the value of predefined QoS parameters. It can be applied in any specific domain where service discovery is made based on independent feedback representing the quality rating of the underlying content (QoS or reputation), and where gaps exist between the expectations and preferences of service providers and consumers. It is not suitable to have providers setting their own QoS or reputation values subjectively. The proposed approaches are helpful to form the values objectively based on a consensus, and it can be iteratively applied for reaching a consensus to mitigate these gaps.

The advantage of the proposed method, MFDM, is its learning mechanism as the MFDM can be triggered iteratively and it allows the service consumers and providers to have arbitrary opinions initially and the system will assist them in moderating their expectations and reaching a group consensus during the process. The more feedback that is gathered from the users, the less subjective the consensus is, since the subjectivity of users' opinions is decreasing and the objectivity of the group's consensus is increasing.

Three cases, each with a number of experiments, have been carried out to demonstrate the effectiveness of the proposed method. The first set of experiments introduced a non-functional discovery criterion, *Cheap*, which is a higher level abstraction from the existing Web services attributes in the trial environment. The QoS term, *Cheap*, represents the quality rating of each service in terms of the cost. The proposed method helps consumers and providers to reach a consensus on the selection criterion *Cheap* and therefore service consumers can, not only discover the services with correct capabilities, but can also locate their desired cheap services by ranking the services according to the QoS term, *Cheap*. The second set of experiments introduced a composite QoS term, *Satisfaction*, which comprised five primitive QoS terms (*Cheap*, *DepartureTime*, *ArrivalTime*, *TravelTime* and *Stops*). In this set of experiments, the proposed method enables consumers and providers to reach a consensus on the each of the primitive terms and helps to calculate the degree of *Satisfaction* by reasoning with the weightings of different primitive terms. Thus, consumers in the second set of experiments can discover the services with higher satisfaction ratings. The third set of experiments demonstrated how the Pseudo-Order Preference Model (POPM) is applied to help consumers to collect preferences on indifferent or indistinguishable criteria and how to reduce the system complexity by selecting only the top-N QoS criteria before the consensus reaching process. After three sets of experiments, the overall result has shown that the proposed method could facilitate consumers and providers to reach a consensus on the discovery criteria, and therefore the proposed method conducts a better average precision rate than other service discovery methods (by 3%~35%) and can greatly reduce the number recommended services (by 22%~55% reductions) which is very useful in helping consumers evaluate the services.

The proposed approach can produce the values for QoS terms and it is applicable to a number of QoS models, such as [90],[91],[92]. Currently, it is assumed that there is only simple dependency among the selection criteria, which in some cases may not be realistic. The design of a layered QoS model based on an application domain can be considered as future work. Such a QoS model should have the ability to keep track of the relationships between different layered QoS terms and have the ability to deal with partial criteria. Furthermore, the SAM and RMGDP processes can be extend to improve the scalability of the proposed approach because these processes are based on matrix calculations which can be extend by divide-and-conquer ways to attain a better performance when there are a large number of participants involved in the moderation process. It is assumed that users will change their subjective opinions and preferences in line with the group consensus. This may not be the case when users have very strong opinions or when they used to change their mind quickly, and therefore a sophisticated negotiation (or personalization) system is required to be in place in order to relax this assumption in the future.

ACKNOWLEDGEMENTS

The author would like to acknowledge Dr. Nick Godwin for his review and valuable comments on this dissertation. The author also thanks the other anonymous reviewers for their useful comments on earlier draft papers related to this research and thanks the National Science Council, Taiwan owing to the financial support for partial of this research. Project number NSC 095–2917–I–009–001.

Reference

[1] T. Berners-Lee, "Services and Semantics: Web Architecture," W3C, April 2001, Available: http://www.w3.org/2001/04/30-tbl

[2] C. Bussler, D. Fensel and A. Maedche, A conceptual architecture for Semantic Web enabled Web Services. SIGMOD Rec. Vol. 31, Issue 4, pp. 24–29

[3] S. Tsur, S. Abiteboul, R. Agrawal, U. Dayal, J. Klein, and G. Weikum, "Are Web services the next revolution in e-commerce?," *Proceedings of VLDB conference*, Rome, September 2001, pp. 614–617

[4] G. Chafle, S. Chandra, V. Mann, and M. G. Nanda, "Decentralized Orchestration of Composite Web Services," pp. 134–143

[5] Edited by L. Clement, A. Hately, C. von Riegen, and T. Rogers, " UDDI Version 3.0.2," OASIS, 19th October 2004, Available: http://uddi.org/pubs/uddi-v3.0.2-20041019.htm

[6] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web Services Description Language (WSDL) 1.1," W3C, 15th Mar., 2001, Available: http://www.w3.org/TR/2001/NOTE-wsdl-20010315

[7] Q. Liang, L. N. Chakarapani, S. Y. W. Su, R. N. Chikkamagalur, and H. Lam, "A Semi-automatic Approach to Composite Web Service Discovery, Description and Invocation," *IJWSR*, Vol. 1, No. 4, 2004, pp. 64–89

[8] Q. A. Liang, J.-Y. Chung and S. Miller, "Towards Semantic Service Request of Web Service Composition," *Proceedings of the 2005 IEEE International Conference on e-Business Engineering (ICEBE 2005)*, IEEE CS, Beijing, China, October 2005

[9] B. Medjahed, A. Bouguettaya, and A. K. Elmagarmid, "Composing Web services on the Semantic Web," *The International Journal on Very Large Data Bases (The VLDB Journal)*, Springer-Verlag, Vol. 12, Issue 4, November 2003, pp. 333–351

[10] M. Gudgin, M. Hadley, N. Mendelsohn, J.-J. Moreau, and H. F. Nielsen , "SOAP Version 1.2 Part 1: Messaging Framework," W3C, 24th June, 2003, Available: http://www.w3.org/TR/2003/REC-soap12-part1-20030624/

[11] M. C. Jaeger, G. R.-Goldmann, C. Liebetruth, G. Muhl, and K. Geihs, "Ranked Matching for Service Descriptions using OWL-S," Proceedings of Kommunikation in Verteilten Systemen (KiVS 2005), TU Kaiserslautern, February 2005, pp. 91–102

[12] M. Paolucci, T. Kawamura, T. Payne, and K. Sycara, "Semantic matching of web services capabilities," *Proceedings of the International Semantic Web Conference (ISWC'02)*, Springer Verlag, Sardegna, Italy, June 2002, pp. 333–347

[13] T. Berners-Lee, J. Hendler and O. Lassila, "The Semantic Web," Scientific American, May 2001, Available: http://www.scientificamerican.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21&catID=2

[14] Edited by M. K. Smith, C. Welty and D. L. McGuinness, "OWL Web Ontology Language Guide," W3C, February 2004, Available: http://www.w3.org/TR/2004/REC-owl-guide-20040210/

[15] D. Martin, M. Burstein, J. Hobbs, O. Lassila, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, E. Sirin, N. Srinivasan, and K. Sycara, "OWL-S: Semantic Markup for Web services", W3C, November 2004, Available: http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/

[16] K. Sycara, and S. Widoff "LARKS: Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace", *Journal of Autonomous Agents and Multi-Agent Systems*, Kluwer Academic Publishers, Vol. 5, May 2002, pp. 173–203

[17] K.-M. Chao, M. Younas, C.-C. Lo and T.-H. Tan, "Fuzzy Matchmaking for Web services," *Proceedings of the 19th IEEE International Conference on Advanced Information Networking and Applications (AINA2005)*, IEEE CS, Vol. 2, Taipei, Taiwan, March 2005, pp. 721–726

[18] Y. Wang, E. Stroulia "Flexible Interface Matching for Web-Service Discovery" *Proceedings of Fourth International Conference on Web Information System Engineering*, IEEE CS, December, 2003

[19] L. Zeng, B. Benatallah, A.H.H Ngu, M. Dumas, J. Kalagnanam, and H Chang, "Qos-Aware Middleware for Web Service Composition," *IEEE Transactions on Software Engineering*, IEEE CS, Vol. 30, No. 5, 2004, pp. 311–327

[20] D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris, and D. Orchard "Web Services Architecture: 1.4 What is a Web service?," W3C, February, 2004, Available: http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#whatis

[21] S. Graham, S. Simeonov, T. Boubez, G. Daniels, D. Davis, Y. Nakamura, and R. Neyama, *Building Web Services with Java: Making sense of XML, SOAP, WSDL and UDDI.* Indianapolis, Ind. SAMS, 2002

[22] Tao-Hsin Tan, *Ontology-based Fuzzy Matchmaking for Web Services*, National Chiao Tung University, M.B.A thesis, June 2005

[23] "MCI UDDI Registry", 2002, Available: http://uddi.mci.com/about_uddi.jsp

[24] F. Sommers, “Publish and find UDDI TModels with JAXR and WSDL,” JavaWorld, December 2002, pp. 2, Available: http://www.javaworld.com/javaworld/jw-12-2002/jw-1213-webservices.html

[25] W3Schools, “WSDL Tutorial – WSDL Documents,” W3Schools, pp. 1–4, Available: http://www.w3schools.com/wsdl/wsdl_documents.asp

[26] W3Schools, “SOAP Tutorial –Introduction to SOAP,” W3Schools, pp. 1–7, Available: http://www.w3schools.com/soap/soap_intro.asp

[27] M. Bunge, “Ontology I: The Furniture of the World,” *Treatise on Basic Philosophy*, Vol. 3, Boston, 1977

[28] B. Chandrasekaran, J.R. Josephson and V.R. Benjamins, “What are ontologies, and why do we need them?,” *IEEE Intelligent systems*, IEEE CS, Vol. 14, Issue 1, Jan./Feb. 1999, pp.20–26

[29] I. Herman, “Short introduction to the Semantic Web - Semantic Web Stack,” W3C, February 2006, pp. 34, Available: http://www.w3.org/Consortium/Offices/Presentations/SemanticWeb/Overview.html

[30] P. F. Patel-Schneider and I. Horrocks, “OWL Web Ontology Language Semantics and Abstract Syntax Section 2. Abstract Syntax,” W3C, February 2004, Available: http://www.w3.org/TR/2004/REC-owl-semantics-20040210/syntax.html

[31] D. L. McGuinness and F. van Harmelen, “OWL Web Ontology Language – Overview,” W3C, February 2004, Available: http://www.w3.org/TR/2004/REC-owl-features-20040210/

[32] J. J. Carroll and J. De Roo, “OWL Web Ontology Language – Test Cases,” W3C, February 2004, Available: http://www.w3.org/TR/owl-test/

[33] J. Kopena, W. C. Regli, “DAMLJessKB: A Tool for Reasoning with the Semantic Web,” *IEEE Intelligent systems*, IEEE CS, Vol. 18, No. 3, May/June 2003, pp. 74–77

[34] J. Kopena, “OWLJessKB: A Semantic Web Reasoning Tool,” owljesskb-20050121.tar.gz, January 2005, Available: http://edge.cs.drexel.edu/assemblies/software/owljesskb/

[35] M. Paolucci, T. Kawamura, T. R. Payne, and K. Sycara, “Importing the Semantic Web in UDDI,” *Lecture Notes in Computer Science (LNCS)*, Springer-Verlag, Vol. 2512, 2002, pp. 225–236

[36] IBM, BEA, and Microsoft, “Business Process Execution Language for Web Services (BPEL4WS),” Technology Report, November 2005, Available: http://xml.coverpages.org/bpel4ws.html

[37] IBM, et al., “Business Process Execution Language for Web Services. Version 1.1,” May 2003, Available: http://xml.coverpages.org/BPELv11-May052003Final.pdf

[38] R. Khalaf, N. Mukhi and S. Weerawarana, “Service-Oriented Composition in BPEL4WS,” *Proceedings of the Twelfth International World Wide Web Conference*, Budapest, Hungery, May 2003

[39] IBM, "Web Services Flow Language (WSFL)," Technology Reports, April 2002, Available: http://xml.coverpages.org/wsfl.html

[40] A. Arkin, S. Askary, S. Fordin and W. Jekeli, et al. "Web Services Choreography Interface (WSCI)," 2003, Available: http://xml.coverpages.org/wsci.html

[41] D. W. Cheung. E. Lo, C. Y. Ng, and T. Lee, "Web Services Oriented Data Processing and Integration," *Proceedings of the Twelfth International World Wide Web Conference*, Budapest, Hungery, May 2003

[42] D. Florescu, A. Grünhagen and D. Kossmann, "XL: An XML Programming Language for Web Service Specification and Composition," *Proceedings of the Eleventh International World Wide Web Conference (WWW02)*, ACM Press, Hawaii, USA , May 2002

[43] F. Casati, S. Ilnicki, L. Jin, V. Krishnamoorthy, and and M.-C. Shan, "Adaptive and Dynamic Service Composition in eFlow," *Proceedings of Conference on Advanced Information Systems Engineering (CAiSE)*, Springer-Verlag, Stockholm, Sweden, June 2000, pp. 13–31

[44] M. G. Nanda, S. Chandra, and V. Sarkar, "Decentralizing Execution of Composite Web Services," *Proceedings of OOPSLA'04*, Vancouver, Canada, ACM Press, 24th–28th October, 2004

[45] IBM alphaworks, "Business Process Execution Language for Web Services JavaTM Run Time (BPWS4J)," IBM, Available: http://www.alphaworks.ibm.com/tech/bpws4j

[46] M. Paolucci, "Toward Semantic Web E-Commence - OWL-S Tools and Applications," The OWL-S Coalition, Available: http://www.daml.org/services/swsl/straw-proposals /OWL-S-Straw-Tools.ppt

[47] H. Wu, H. Jin, Y.-F. Li, and H. Chen, "An Approach for Service Discovery Based on Semantic Peer-to-Peer," *Lecture Notes in Computer Science (LNCS)*, Springer-Verlag, Vol. 3818, 2005, pp. 259–260

[48] M. C. Jaeger and S. Tang, "Ranked Matching for Service Descriptions using DAML-S," *Proceedings of Conference on Advanced Information Systems Engineering (CAiSE'04)*, Springer-Verlag, Vol. 3, Riga, Latvia, June 2004, pp. 217–228

[49] K. Sycara, M. Paolucci, J. Soudry, and N. Srinivasan, "Dynamic Discovery and Coordination of Agent-Based Semantic Web Services," *IEEE Internet Computing*, IEEE CS, Vol. 8, No. 3, May/June 2004, pp. 66–73

[50] M. Paolucci, J. Soudry, N. Srinivasan, and K. Sycara, "A Broker for OWL-S Web services," *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI04)*, AAAI Press, March 2004

[51] R. Akkiraju, R. Goodwin, P. Doshi, and S. Roeder, "A Method for Semantically Enhancing the Service Discovery Capabilities of UDDI," *Proceedings of IJCAI Workshop on Information Integration on the Web*, Acapulco, Mexico, August 2003, pp. 87–92

[52] A. Maedchea, S. Staabb, “Services on the Move — Towards P2P-Enabled Semantic Web Services,” *Proceedings of the 10th International Conference on Information Technology and Travel & Tourism*, Springer-Verlag, Helsinki, Finland, January 2003

[53] D. Fensel, S. Staab, R. Studer, and F. van Harmelen, “Peer-2-Peer Enabled Semantic Web for Modern Knowledge Management,” *Towards the Semantic Web – Ontology-based Knowledge Management*, Wiley, London, UK, 2002

[54] S. Bansal and J. M. Vidal, “Matchmaking of Web Services Based on the DAMLS Service Model,” *Proceedings of International Conference on Autonomous Agents and MultiAgent Systems (AAMAS’03)*, ACM Press, Melbourne, Australia, July 2003, pp. 926–927

[55] I. Elgedawy, Z. Tari and M. Winikoff , “Exact Functional Context Matching for Web Services,” *Proceeding of the 2nd International Conference on Service Oriented Computing (ICSOC’04)*, ACM Press, New York, USA, November 2004, pp. 143–152

[56] R. Fagin, R. Kumar, D. Sivakumar, “Efficient Similarity Search and Classification via Rank Aggregation,” *Data Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, ACM Press, 2003, pp. 301–312

[57] U. Nambiar and S. Kambhampati, “Answering Imprecise Database Queries: A Novel Approach,” *Proceedings of the 5th ACM International Workshop on Web Information and Data Management*, ACM Press, 2003, pp. 126–33

[58] M. Lin, J. Xie and H. Guo, “Solving QoS-Driven Web Service Dynamic Composition as Fuzzy Constraint Satisfaction,” *Proceedings of 2005 IEEE International Conf. on e-Technology, e-Commerce and e-Service (EEE-05)*, IEEE CS, Hong Kong, 2005, pp. 9–14

[59] S Ran, “A Model for Web Services Discovery with QoS,” *ACM SIGecom Exchange*, ACM Press, Vol. 4, Issue 1, 2003, pp. 1–10

[60] W.-T. Balke and M. Wagner, “Through Different Eyes – Assessing Multiple Conceptual Views for Querying Web Services,” *Proceedings of the Thirteenth International World Wide Web Conference (WWW2004)*, ACM Press, New York, USA, May 2004, pp. 196–205

[61] H.-M. Hsu and C.-T. Chen, ”Aggregation of Fuzzy Opinions Under Group Decision Making,” *Fuzzy Sets and Systems*, Elsevier Science, Vol. 79, 1996, pp. 279–285

[62] H.-S. Lee, “Optimal Consensus of Fuzzy Opinions Under Group Decision Making Environment,” *Fuzzy Sets and Systems*, Elsevier Science, Vol. 132, 2002, pp. 303–315

[63] A. Bardossy, L. Duckstein and I. Bogardi, ”Combination of fuzzy numbers representing expert opinions,” *Fuzzy Sets and Systems*, Elsevier Science, Vol. 57, pp 173–181, 1993

[64] R. Zwick, E. Carlstein, D.V. Budescu, ”Measures of similarity among fuzzy concepts: A comparative analysis,” *International Journal of Approximate Reasoning*, pp 221–242, 1987

[65] E. Herrera-Viedma, F. Herrera and F. Chiclana, "A Consensus Model for Multiperson Decision Making With Different Preference Structures," *IEEE Transactions on Systems, Man and Cybernetics*, IEEE, Vol. 32, 2002, pp. 394–402

[66] F. Chiclana, F. Herrera and E. Herrera-Viedma, "Integrating Multiplicative Preference Relations in A Multipurpose Decision-making Model Based on Fuzzy Preference Relations," *Fuzzy Sets and Systems*, Elsevier Science, Vol. 122, 2001, pp. 277–291

[67] F. Chiclana, F. Herrera, and E. Herrera-Viedma, "A Classification Method of Alternatives for Multiple Preference Ordering Criteria Based on Fuzzy Majority," *Journal. Fuzzy Math.*, Vol. 34, 1996, pp. 224–229

[68] F. Chiclana, F. Herrera and E. Herrera-Viedma, "Integrating Three Representation Models in Fuzzy Multipurpose Decision Making Based on Fuzzy Preference Relations," *Fuzzy Sets and Systems*, Elsevier Science, Vol. 97, 1998, pp. 33–48

[69] F. Herrera, et al., "A Rational Consensus Model in Group Decision Making Using Linguistic Assessments," *Fuzzy set and Systems*, Elsevier Science, Vol. 88, 1997, pp. 31–49

[70] R.R. Yager, "On Ordered Weighted Averaging Aggregation Operators in Multi Criteria Decision Making," *IEEE Transaction on Systems, Man. and Cybernetics*, IEEE, Vol. 18, 1988, pp. 183–190

[71] B. Roy and P. H. Vincke, "Relational System of Preference with One or More Pseudo-Criteria: Some New Concepts and Results," *Management Science*, Vol. 30, No. 11, 1984, pp. 1323–1335

[72] J. Wang and Y.I. Lin, "A Fuzzy Multicriteria Group Decision Making Approach to Select Configuration Items for Software Development," *Fuzzy Sets and Systems*, Elsevier Science, Vol. 134, 2003, pp. 343–363

[73] H.-J. Zimmermann, *Fuzzy Set Theory and Its Applications*, Kluwer Academic Publishers, 1991

[74] J. Yen and R. Langari, *Fuzzy Logic: Intelligence, Control, and Information*, Prentice Hall, 1999

[75] S.A. Orlovski, "Decision-Making with A Fuzzy Preference Relation," *Fuzzy Sets and Systems,* Elsevier Science, Vol. 1, 1978, pp. 155–167

[76] J. Wang, "A Fuzzy Outranking Approach for Design Evaluation in Conceptual Design," *International Journal of Production Research*, Taylor & Francis, Vol. 35, Issue 4, 1997, pp. 995–1010

[77] M. Horridge, H. Knublauch, A. Rector, R. Stevens, and C. Wroe, "A Practical Guide to Building OWL Ontologies: Using the Protégé-OWL Plugin and CO-ODE Tools Edition 1.0," The University of Manchester, August 2004

[78] E. Friedman-Hill, "Jess, The Rule Engine for the Java Platform," March 2006, Available: http://herzberg.ca.sandia.gov/jess/

[79] B. N. Grosof, M. D. Grandhe and T. W. Finin, “SweetJess: Inferencing in Situated Courteous RuleML via Translation to and from Jess Rules,” May 2003

[80] Apache, “Welcome to jUDDI: an open source Java implementation of the Universal Description, Discovery, and Integration (UDDI) specification for Web Services,” Apache, June 2005, Available: http://ws.apache.org/juddi/

[81] R. Maucherat et al., “Apache Tomcat 5.5,” The Apache Software Foundation, Available: http://tomcat.apache.org/

[82] MySQL AB, “MySQL 5.0,” Available: http://dev.mysql.com/downloads/mysql/5.0.html

[83] Sun Microsystems, “Java 2 Platform Standard Edition 5.0 (J2SE 1.5),” Sun Microsystems, Available: http://java.sun.com/j2se/1.5.0/download.jsp

[84] Wolfram Research Inc., “Mathematica – The Way the World Calculates,” Wolfram Research Inc., 2006, Available: http://www.wolfram.com/products/mathematica/index.en.html

[85] The MathWorks, Inc., “MATLAB & Simulink Student Version Activation,” The MathWorks Inc., 2006, Available: http://www.mathworks.com/academia/student_version/activation.html

[86] Raw data source for Case I, ezfly Inc., June 2006, Available: http://www.ezfly.com/

[87] Raw data source for Case II, A2bTravel.com, August 2006, Available: http://www.a2bcheapflights.otc-uk.com/

[88] Raw data source for Case II, Cheapflights Ltd., August 2006, Available: http://www.cheapflights.co.uk/

[89] Raw data source for Case II, Airline Network, August 2006, http://www.airline-network.co.uk/index.asp

[90] Shuping Ran, “A Model for Web Services Discovery With QoS,” *ACM SIGecom Exchanges*, ACM, Vol. 4, Issue 1, 2003, pp. 1–10

[91] Sven Mathijssen, “A Fair Model for Quality of Web Services,” *3rd Twente Student Conference on IT*, University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science, 20 June, 2005

[92] Yannis Makripoulias, Christos Makris, Yiannis Panagis, Evangelos Sakkopoulos, Poulia Adamopoulou, Maria Pontikaki, and Athanasios Tsakalidis, “Towards Ubiquitous Computing with Quality of Web Service Support,” *UPGRADE, the European Journal for the Informatics Professional*, Novática, Vol. VI, No. 5, October 2005, pp. 29–34

$$
\mathrm{AM} = \begin{pmatrix}
1 & \frac{21}{23} & \frac{7}{8} & \frac{3}{4} & \frac{21}{32} & \frac{6}{7} & \frac{7}{8} & \frac{21}{34} & \frac{21}{25} & \frac{21}{26} & \frac{6}{7} & \frac{21}{22} & \frac{3}{4} & \frac{21}{31} & \frac{17}{21} & \frac{7}{8} & \frac{21}{26} & \frac{21}{32} & \frac{7}{10} & \frac{21}{34} & \frac{21}{38} & \frac{7}{8} & \frac{21}{32} & \frac{7}{12} & \frac{3}{4} & \frac{21}{32} & \frac{21}{26} & \frac{21}{32} & \frac{7}{10} & \frac{21}{26} \\
\frac{21}{23} & 1 & \frac{23}{24} & \frac{23}{28} & \frac{23}{32} & \frac{18}{23} & \frac{23}{24} & \frac{23}{34} & \frac{23}{25} & \frac{23}{26} & \frac{18}{23} & \frac{22}{23} & \frac{23}{28} & \frac{23}{31} & \frac{17}{23} & \frac{23}{24} & \frac{23}{26} & \frac{23}{32} & \frac{23}{30} & \frac{23}{34} & \frac{23}{38} & \frac{23}{24} & \frac{23}{32} & \frac{23}{36} & \frac{23}{28} & \frac{23}{32} & \frac{23}{26} & \frac{23}{32} & \frac{23}{30} & \frac{23}{26} \\
\frac{7}{8} & \frac{23}{24} & 1 & \frac{6}{7} & \frac{3}{4} & \frac{3}{4} & 1 & \frac{12}{17} & \frac{24}{25} & \frac{12}{13} & \frac{3}{4} & \frac{11}{12} & \frac{6}{7} & \frac{24}{31} & \frac{17}{24} & 1 & \frac{12}{13} & \frac{3}{4} & \frac{4}{5} & \frac{12}{17} & \frac{12}{19} & 1 & \frac{3}{4} & \frac{2}{3} & \frac{6}{7} & \frac{3}{4} & \frac{12}{13} & \frac{3}{4} & \frac{4}{5} & \frac{12}{13} \\
\frac{3}{4} & \frac{23}{28} & \frac{6}{7} & 1 & \frac{7}{8} & \frac{9}{14} & \frac{6}{7} & \frac{14}{17} & \frac{25}{28} & \frac{13}{14} & \frac{9}{14} & \frac{11}{14} & 1 & \frac{28}{31} & \frac{17}{28} & \frac{6}{7} & \frac{13}{14} & \frac{7}{8} & \frac{14}{15} & \frac{14}{17} & \frac{14}{19} & \frac{6}{7} & \frac{7}{8} & \frac{7}{9} & 1 & \frac{7}{8} & \frac{13}{14} & \frac{7}{8} & \frac{14}{15} & \frac{13}{14} \\
\frac{21}{32} & \frac{23}{32} & \frac{3}{4} & \frac{7}{8} & 1 & \frac{9}{16} & \frac{3}{4} & \frac{16}{17} & \frac{25}{32} & \frac{13}{16} & \frac{9}{16} & \frac{11}{16} & \frac{7}{8} & \frac{31}{32} & \frac{17}{32} & \frac{3}{4} & \frac{13}{16} & 1 & \frac{15}{16} & \frac{16}{17} & \frac{16}{19} & \frac{3}{4} & 1 & \frac{8}{9} & \frac{7}{8} & 1 & \frac{13}{16} & \frac{63}{65} & \frac{15}{16} & \frac{13}{16} \\
\frac{6}{7} & \frac{18}{23} & \frac{3}{4} & \frac{9}{14} & \frac{9}{16} & 1 & \frac{3}{4} & \frac{9}{17} & \frac{18}{25} & \frac{9}{13} & 1 & \frac{9}{11} & \frac{9}{14} & \frac{18}{31} & \frac{17}{18} & \frac{3}{4} & \frac{9}{13} & \frac{9}{16} & \frac{3}{5} & \frac{9}{17} & \frac{9}{19} & \frac{3}{4} & \frac{9}{16} & \frac{1}{2} & \frac{9}{14} & \frac{9}{16} & \frac{9}{13} & \frac{9}{16} & \frac{3}{5} & \frac{9}{13} \\
\frac{7}{8} & \frac{23}{24} & 1 & \frac{6}{7} & \frac{3}{4} & \frac{3}{4} & 1 & \frac{12}{17} & \frac{24}{25} & \frac{12}{13} & \frac{3}{4} & \frac{11}{12} & \frac{6}{7} & \frac{24}{31} & \frac{17}{24} & 1 & \frac{12}{13} & \frac{3}{4} & \frac{4}{5} & \frac{12}{17} & \frac{12}{19} & 1 & \frac{3}{4} & \frac{2}{3} & \frac{6}{7} & \frac{3}{4} & \frac{12}{13} & \frac{3}{4} & \frac{4}{5} & \frac{12}{13} \\
\frac{21}{34} & \frac{23}{34} & \frac{12}{17} & \frac{14}{17} & \frac{16}{17} & \frac{9}{17} & \frac{12}{17} & 1 & \frac{25}{34} & \frac{13}{17} & \frac{9}{17} & \frac{11}{17} & \frac{14}{17} & \frac{31}{34} & \frac{1}{2} & \frac{12}{17} & \frac{13}{17} & \frac{16}{17} & \frac{15}{17} & \frac{33}{35} & \frac{17}{19} & \frac{12}{17} & \frac{16}{17} & \frac{17}{18} & \frac{14}{17} & \frac{16}{17} & \frac{13}{17} & \frac{16}{17} & \frac{15}{17} & \frac{13}{17} \\
\frac{21}{25} & \frac{23}{25} & \frac{24}{25} & \frac{25}{28} & \frac{25}{32} & \frac{18}{25} & \frac{24}{25} & \frac{25}{34} & 1 & \frac{74}{79} & \frac{18}{25} & \frac{22}{25} & \frac{25}{28} & \frac{25}{31} & \frac{17}{25} & \frac{24}{25} & \frac{25}{26} & \frac{25}{32} & \frac{5}{6} & \frac{25}{34} & \frac{25}{38} & \frac{24}{25} & \frac{25}{32} & \frac{25}{36} & \frac{25}{28} & \frac{25}{32} & \frac{25}{26} & \frac{25}{32} & \frac{5}{6} & \frac{25}{26} \\
\frac{21}{26} & \frac{23}{26} & \frac{12}{13} & \frac{13}{14} & \frac{13}{16} & \frac{9}{13} & \frac{12}{13} & \frac{13}{17} & \frac{74}{79} & 1 & \frac{9}{13} & \frac{11}{13} & \frac{13}{14} & \frac{26}{31} & \frac{17}{26} & \frac{12}{13} & \frac{25}{27} & \frac{13}{16} & \frac{13}{15} & \frac{13}{17} & \frac{13}{19} & \frac{12}{13} & \frac{13}{16} & \frac{13}{18} & \frac{13}{14} & \frac{13}{16} & \frac{25}{27} & \frac{13}{16} & \frac{13}{15} & \frac{25}{27} \\
\frac{6}{7} & \frac{18}{23} & \frac{3}{4} & \frac{9}{14} & \frac{9}{16} & 1 & \frac{3}{4} & \frac{9}{17} & \frac{18}{25} & \frac{9}{13} & 1 & \frac{9}{11} & \frac{9}{14} & \frac{18}{31} & \frac{17}{18} & \frac{3}{4} & \frac{9}{13} & \frac{9}{16} & \frac{3}{5} & \frac{9}{17} & \frac{9}{19} & \frac{3}{4} & \frac{9}{16} & \frac{1}{2} & \frac{9}{14} & \frac{9}{16} & \frac{9}{13} & \frac{9}{16} & \frac{3}{5} & \frac{9}{13} \\
\frac{21}{22} & \frac{22}{23} & \frac{11}{12} & \frac{11}{14} & \frac{11}{16} & \frac{9}{11} & \frac{11}{12} & \frac{11}{17} & \frac{22}{25} & \frac{11}{13} & \frac{9}{11} & 1 & \frac{11}{14} & \frac{22}{31} & \frac{17}{22} & \frac{11}{12} & \frac{11}{13} & \frac{11}{16} & \frac{11}{15} & \frac{11}{17} & \frac{11}{19} & \frac{11}{12} & \frac{11}{16} & \frac{11}{18} & \frac{11}{14} & \frac{11}{16} & \frac{11}{13} & \frac{11}{16} & \frac{11}{15} & \frac{11}{13} \\
\frac{3}{4} & \frac{23}{28} & \frac{6}{7} & 1 & \frac{7}{8} & \frac{9}{14} & \frac{6}{7} & \frac{14}{17} & \frac{25}{28} & \frac{13}{14} & \frac{9}{14} & \frac{11}{14} & 1 & \frac{28}{31} & \frac{17}{28} & \frac{6}{7} & \frac{13}{14} & \frac{7}{8} & \frac{14}{15} & \frac{14}{17} & \frac{14}{19} & \frac{6}{7} & \frac{7}{8} & \frac{7}{9} & 1 & \frac{7}{8} & \frac{13}{14} & \frac{7}{8} & \frac{14}{15} & \frac{13}{14} \\
\frac{21}{31} & \frac{23}{31} & \frac{24}{31} & \frac{28}{31} & \frac{31}{32} & \frac{18}{31} & \frac{24}{31} & \frac{31}{34} & \frac{25}{31} & \frac{26}{31} & \frac{18}{31} & \frac{22}{31} & \frac{28}{31} & 1 & \frac{17}{31} & \frac{24}{31} & \frac{26}{31} & \frac{31}{32} & \frac{30}{31} & \frac{31}{34} & \frac{31}{38} & \frac{24}{31} & \frac{31}{32} & \frac{31}{36} & \frac{28}{31} & \frac{31}{32} & \frac{26}{31} & \frac{92}{97} & \frac{89}{94} & \frac{26}{31} \\
\frac{17}{21} & \frac{17}{23} & \frac{17}{24} & \frac{17}{28} & \frac{17}{32} & \frac{17}{18} & \frac{17}{24} & \frac{1}{2} & \frac{17}{25} & \frac{17}{26} & \frac{17}{18} & \frac{17}{22} & \frac{17}{28} & \frac{17}{31} & 1 & \frac{17}{24} & \frac{17}{26} & \frac{17}{32} & \frac{17}{30} & \frac{1}{2} & \frac{17}{38} & \frac{17}{24} & \frac{17}{32} & \frac{17}{36} & \frac{17}{28} & \frac{17}{32} & \frac{17}{26} & \frac{17}{32} & \frac{17}{30} & \frac{17}{26} \\
\frac{7}{8} & \frac{23}{24} & 1 & \frac{6}{7} & \frac{3}{4} & \frac{3}{4} & 1 & \frac{12}{17} & \frac{24}{25} & \frac{12}{13} & \frac{3}{4} & \frac{11}{12} & \frac{6}{7} & \frac{24}{31} & \frac{17}{24} & 1 & \frac{12}{13} & \frac{3}{4} & \frac{4}{5} & \frac{12}{17} & \frac{12}{19} & 1 & \frac{3}{4} & \frac{2}{3} & \frac{6}{7} & \frac{3}{4} & \frac{12}{13} & \frac{3}{4} & \frac{4}{5} & \frac{12}{13} \\
\frac{21}{26} & \frac{23}{26} & \frac{12}{13} & \frac{13}{14} & \frac{13}{16} & \frac{9}{13} & \frac{12}{13} & \frac{13}{17} & \frac{25}{26} & \frac{25}{27} & \frac{9}{13} & \frac{11}{13} & \frac{13}{14} & \frac{26}{31} & \frac{17}{26} & \frac{12}{13} & 1 & \frac{13}{16} & \frac{13}{15} & \frac{13}{17} & \frac{13}{19} & \frac{12}{13} & \frac{13}{16} & \frac{13}{18} & \frac{13}{14} & \frac{13}{16} & 1 & \frac{13}{16} & \frac{13}{15} & 1 \\
\frac{21}{32} & \frac{23}{32} & \frac{3}{4} & \frac{7}{8} & 1 & \frac{9}{16} & \frac{3}{4} & \frac{16}{17} & \frac{25}{32} & \frac{13}{16} & \frac{9}{16} & \frac{11}{16} & \frac{7}{8} & \frac{31}{32} & \frac{17}{32} & \frac{3}{4} & \frac{13}{16} & 1 & \frac{15}{16} & \frac{16}{17} & \frac{16}{19} & \frac{3}{4} & 1 & \frac{8}{9} & \frac{7}{8} & 1 & \frac{13}{16} & \frac{63}{65} & \frac{15}{16} & \frac{13}{16} \\
\frac{7}{10} & \frac{23}{30} & \frac{4}{5} & \frac{14}{15} & \frac{15}{16} & \frac{3}{5} & \frac{4}{5} & \frac{15}{17} & \frac{5}{6} & \frac{13}{15} & \frac{3}{5} & \frac{11}{15} & \frac{14}{15} & \frac{30}{31} & \frac{17}{30} & \frac{4}{5} & \frac{13}{15} & \frac{15}{16} & 1 & \frac{15}{17} & \frac{15}{19} & \frac{4}{5} & \frac{15}{16} & \frac{5}{6} & \frac{14}{15} & \frac{15}{16} & \frac{13}{15} & \frac{119}{129} & \frac{29}{31} & \frac{13}{15} \\
\frac{21}{34} & \frac{23}{34} & \frac{12}{17} & \frac{14}{17} & \frac{16}{17} & \frac{9}{17} & \frac{12}{17} & \frac{33}{35} & \frac{25}{34} & \frac{13}{17} & \frac{9}{17} & \frac{11}{17} & \frac{14}{17} & \frac{31}{34} & \frac{1}{2} & \frac{12}{17} & \frac{13}{17} & \frac{16}{17} & \frac{15}{17} & 1 & \frac{17}{19} & \frac{12}{17} & \frac{16}{17} & \frac{17}{18} & \frac{14}{17} & \frac{16}{17} & \frac{13}{17} & \frac{127}{137} & \frac{15}{17} & \frac{13}{17} \\
\frac{21}{38} & \frac{23}{38} & \frac{12}{19} & \frac{14}{19} & \frac{16}{19} & \frac{9}{19} & \frac{12}{19} & \frac{17}{19} & \frac{25}{38} & \frac{13}{19} & \frac{9}{19} & \frac{11}{19} & \frac{14}{19} & \frac{31}{38} & \frac{17}{38} & \frac{12}{19} & \frac{13}{19} & \frac{16}{19} & \frac{15}{19} & \frac{17}{19} & 1 & \frac{12}{19} & \frac{16}{19} & \frac{18}{19} & \frac{14}{19} & \frac{16}{19} & \frac{13}{19} & \frac{16}{19} & \frac{15}{19} & \frac{13}{19} \\
\frac{7}{8} & \frac{23}{24} & 1 & \frac{6}{7} & \frac{3}{4} & \frac{3}{4} & 1 & \frac{12}{17} & \frac{24}{25} & \frac{12}{13} & \frac{3}{4} & \frac{11}{12} & \frac{6}{7} & \frac{24}{31} & \frac{17}{24} & 1 & \frac{12}{13} & \frac{3}{4} & \frac{4}{5} & \frac{12}{17} & \frac{12}{19} & 1 & \frac{3}{4} & \frac{2}{3} & \frac{6}{7} & \frac{3}{4} & \frac{12}{13} & \frac{3}{4} & \frac{4}{5} & \frac{12}{13} \\
\frac{21}{32} & \frac{23}{32} & \frac{3}{4} & \frac{7}{8} & 1 & \frac{9}{16} & \frac{3}{4} & \frac{16}{17} & \frac{25}{32} & \frac{13}{16} & \frac{9}{16} & \frac{11}{16} & \frac{7}{8} & \frac{31}{32} & \frac{17}{32} & \frac{3}{4} & \frac{13}{16} & 1 & \frac{15}{16} & \frac{16}{17} & \frac{16}{19} & \frac{3}{4} & 1 & \frac{8}{9} & \frac{7}{8} & 1 & \frac{13}{16} & \frac{63}{65} & \frac{15}{16} & \frac{13}{16} \\
\frac{7}{12} & \frac{23}{36} & \frac{2}{3} & \frac{7}{9} & \frac{8}{9} & \frac{1}{2} & \frac{2}{3} & \frac{17}{18} & \frac{25}{36} & \frac{13}{18} & \frac{1}{2} & \frac{11}{18} & \frac{7}{9} & \frac{31}{36} & \frac{17}{36} & \frac{2}{3} & \frac{13}{18} & \frac{8}{9} & \frac{5}{6} & \frac{17}{18} & \frac{18}{19} & \frac{2}{3} & \frac{8}{9} & 1 & \frac{7}{9} & \frac{8}{9} & \frac{13}{18} & \frac{8}{9} & \frac{5}{6} & \frac{13}{18} \\
\frac{3}{4} & \frac{23}{28} & \frac{6}{7} & 1 & \frac{7}{8} & \frac{9}{14} & \frac{6}{7} & \frac{14}{17} & \frac{25}{28} & \frac{13}{14} & \frac{9}{14} & \frac{11}{14} & 1 & \frac{28}{31} & \frac{17}{28} & \frac{6}{7} & \frac{13}{14} & \frac{7}{8} & \frac{14}{15} & \frac{14}{17} & \frac{14}{19} & \frac{6}{7} & \frac{7}{8} & \frac{7}{9} & 1 & \frac{7}{8} & \frac{13}{14} & \frac{7}{8} & \frac{14}{15} & \frac{13}{14} \\
\frac{21}{32} & \frac{23}{32} & \frac{3}{4} & \frac{7}{8} & 1 & \frac{9}{16} & \frac{3}{4} & \frac{16}{17} & \frac{25}{32} & \frac{13}{16} & \frac{9}{16} & \frac{11}{16} & \frac{7}{8} & \frac{31}{32} & \frac{17}{32} & \frac{3}{4} & \frac{13}{16} & 1 & \frac{15}{16} & \frac{16}{17} & \frac{16}{19} & \frac{3}{4} & 1 & \frac{8}{9} & \frac{7}{8} & 1 & \frac{13}{16} & \frac{63}{65} & \frac{15}{16} & \frac{13}{16} \\
\frac{21}{26} & \frac{23}{26} & \frac{12}{13} & \frac{13}{14} & \frac{13}{16} & \frac{9}{13} & \frac{12}{13} & \frac{13}{17} & \frac{25}{26} & \frac{25}{27} & \frac{9}{13} & \frac{11}{13} & \frac{13}{14} & \frac{26}{31} & \frac{17}{26} & \frac{12}{13} & 1 & \frac{13}{16} & \frac{13}{15} & \frac{13}{17} & \frac{13}{19} & \frac{12}{13} & \frac{13}{16} & \frac{13}{18} & \frac{13}{14} & \frac{13}{16} & 1 & \frac{13}{16} & \frac{13}{15} & 1 \\
\frac{21}{32} & \frac{23}{32} & \frac{3}{4} & \frac{7}{8} & \frac{63}{65} & \frac{9}{16} & \frac{3}{4} & \frac{16}{17} & \frac{25}{32} & \frac{13}{16} & \frac{9}{16} & \frac{11}{16} & \frac{7}{8} & \frac{92}{97} & \frac{17}{32} & \frac{3}{4} & \frac{13}{16} & \frac{63}{65} & \frac{119}{129} & \frac{127}{137} & \frac{16}{19} & \frac{3}{4} & \frac{63}{65} & \frac{8}{9} & \frac{7}{8} & \frac{63}{65} & \frac{13}{16} & 1 & \frac{15}{16} & \frac{13}{16} \\
\frac{7}{10} & \frac{23}{30} & \frac{4}{5} & \frac{14}{15} & \frac{15}{16} & \frac{3}{5} & \frac{4}{5} & \frac{15}{17} & \frac{5}{6} & \frac{13}{15} & \frac{3}{5} & \frac{11}{15} & \frac{14}{15} & \frac{89}{94} & \frac{17}{30} & \frac{4}{5} & \frac{13}{15} & \frac{15}{16} & \frac{29}{31} & \frac{15}{17} & \frac{15}{19} & \frac{4}{5} & \frac{15}{16} & \frac{5}{6} & \frac{14}{15} & \frac{15}{16} & \frac{13}{15} & \frac{15}{16} & 1 & \frac{13}{15} \\
\frac{21}{26} & \frac{23}{26} & \frac{12}{13} & \frac{13}{14} & \frac{13}{16} & \frac{9}{13} & \frac{12}{13} & \frac{13}{17} & \frac{25}{26} & \frac{25}{27} & \frac{9}{13} & \frac{11}{13} & \frac{13}{14} & \frac{26}{31} & \frac{17}{26} & \frac{12}{13} & 1 & \frac{13}{16} & \frac{13}{15} & \frac{13}{17} & \frac{13}{19} & \frac{12}{13} & \frac{13}{16} & \frac{13}{18} & \frac{13}{14} & \frac{13}{16} & 1 & \frac{13}{16} & \frac{13}{15} & 1
\end{pmatrix}
$$