Fall 9-30-2008

A QoS-Based Services Selected Method in Service-Oriented Architectures Using Ant Colony System - A Case Study of Airflights

Chi-Chun Lo

Yin-Jung Lu

Chi-Hua Chen

Ding-Yuan Cheng

A QoS-Based Services Selected Method in Service-Oriented Architectures Using Ant Colony System - A Case Study of Airflights

Chi-Chun Lo, National Chiao Tung University, Taiwan, cclo@faculty.nctu.edu.tw
Yin-Jung Lu, National Chiao Tung University, Taiwan, yinglung0407.iim96g@nctu.edu.tw
Chi-Hua Chen, National Chiao Tung University, Taiwan, chihua0826.iim96g@nctu.edu.tw
Ding-Yuan Cheng, National Chiao Tung University, Taiwan, kewas@iim.nctu.edu.tw

Abstract

Semantic web is becoming more and more popular these days, and it's an opportune moment to look at the field's current state and future opportunities. However, most researchers focus on only one single service recommend from semantic web inference. In some situations, the Multi-Services which are combined many complex services from various service providers are better than single service. The designed Multi-Services Semantic Search System (MS^4), which provides the cooperation web-based platform for all related mobile users and service providers, could strengthen the ability of Multi-Services suggestion. In this research, MS^4 chooses the adaptable airflight as a case study. MS^4 is a five-components system composed of the Mobile Users (MUs), UDDI Registries (UDDIRs), Service Providers (SPs), Semantic Web Services Server (SWSS), and Database Server (DS). Using SOA, OWL-S to build semantic web environment to inference user's requirements and search various web services which are published in UDDI through the communication networks include internet and 3G/GPRS/GSM mobile networks. In this airline case, we propose the Adaptive Airflights Inference Module (AAIM) combined QoS-Based Services Selected Method (QBSSM) using Ant Colony System (ACS) to reference the adaptable airflights to MUs.

Keywords: Web Services, Service-Oriented Architecture, QoS-Based Services Selected Method, Fuzzy Theory, Ontology.

1. Introduction

Semantic web is becoming more and more popular these days, and it's an opportune moment to look at the field's current state and future opportunities. A semantic web can be though of as a web that is highly intelligent and sophisticated and one needs little or no human intervention to carry out tasks such as adaptable air fight collocation, scheduling appointments, coordinating activities, and searching for complex documents as well as integrating various databases and information systems.

Recently there have been many developments on recommendation model for the semantic web. For inference user's requirement to recommend, the semantic web possibly combines Service-Oriented Architecture (SOA, includes UDDI (Universal Description, Discovery and Integration), SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language)) with RDF (Resource Description Framework), DAML (DARPA Agent Markup Language), DAML-S, DAML+OIL, OWL (Web Ontology Language), OWL-S, or etc. [2, 3, 8]. However, most researchers focus on only one single service recommend from semantic web inference. The single service is mean only one service which maybe a complex-service but provided by the same one. In this airline case, the single service is like making one airfight ticket reservation to Airline Company. In some situations, the Multi-Services which are combined many complex services from various service providers are better than single service. For example, to make a set of airflights tickets (such as Multi-Services, which possibly need some turning points) reservation is cheaper than make only one airflights ticket reservation.

The need for Multi-Services recommend in semantic web is driven by three demands.
(i) To inference user's requirements by semantic engine.
(ii) To search, compare, reorganize, and integrate relevant web services to be Multi-Services.
(iii) To reduce query processes and time.

This paper provides an overview of the Multi-Services recommend in semantic web, combines the technical application of the SOA, OWL-S, semantic web on information system, the system gives strong auxiliary utility to support users while they have some complex problem. The designed Multi-Services Semantic Search System (MS^4) is a five-components system composed of the Mobile Users (MUs), UDDI Registries (UDDIRs), Service Providers (SPs), Semantic Web Services Server (SWSS), and Database Server (DS). Using SOA, OWL-S to build semantic web environment to inference user's requirements and search various web services which are published in UDDI through the communication networks include internet and 3G/GPRS/GSM mobile networks. In this airline case, we propose the Adaptive Airflights Inference Module (AAIM) combined QoS-Based Services Selected Method (QBSSM) using Ant Colony System (ACS) to reference the adaptable airflights to MUs.

The remainder of the thesis is built as follows. In Section 2 we provide background knowledge through the description of related technologies, such as the concept of Semantic Web (SW), fuzzy logic, and Ant Colony System (ACS), and the discussion about the current methods for Multi-Services searching. The complete framework we proposed is explained in Section 3. In Section 4 will illustrate how we implement the architecture for Airline service as an example in our proposed framework. Finally conclusion and the future work are given in Section 5.

2. Research Background and Theory Discussion

Design Multi-Services Semantic Search System (MS^4) is to provide (i) user's requirement inference, (ii) Multi-Services decision support, (iii) searching and inference performance. Necessary research background and relevant technology include as follows: (1) Semantic Web (SW), (2) fuzzy theory, and (3) Ant Colony System (ACS).

2.1 Semantic Web (SW)

To solve the problem of lacking effective service query mechanism in existing web services, a Semantic Web (SW) based technology based web services query mechanism was proposed by Tim Berner-Lee whose proposed vision [10] is shown as Fig. 1. In this paper, we focus on (1) semantic inference and (2) system performance described as follow.

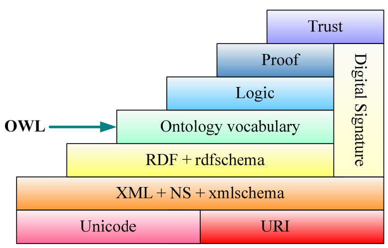

Figure 1. Semantic Web Stack [10]

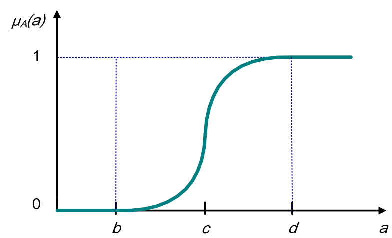

Figure 2. Trapezoidal Membership Function

2.1.1 Semantic Inference

For inference user's requirement semantically, Ora Lassila and James Hendler [8]proposed a architecture of SW applications based on RDF, with patterns in which one component uses another as a data source (via SPARQL) and acts as a data source to yet another component. However, RDF and RDF schema provide properties and syntax not completely to build ontology architecture. In this paper, we use the OWL-S which is an OWL-based Web service ontology that supplies web service providers with a core set of markup language constructs for describing the properties and capabilities of Web services in unambiguous, computer-interpretable form.

2.1.2 System Performance

For efficient selection of QoS-aware web service, in reference [5], we can know the inquiry API of JUDDI has better performance than JWSDP (Java Web Services Development Pack). And there were some approaches proposed by reference [7, 13], which used "cache" mechanism for reducing process and queries while service broker inferences QoS-aware web services. Therefore, we choice JUDDI to build MS^4 with "cache" mechanism to provide SW services.

2.2 Fuzzy Theory

The Theory of fuzzy set is proposed on the basis of the classical set theory. A fuzzy set is a set with the boundary between 0 and 1. Unlike classical set theory, the value of fuzzy set isn't just 0 or 1. It is a smooth boundary for the fuzzy set theory. We review S-membership function [1](shown as Fig. 2) for representing the proposed algorithm in Section 3.4.2.

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

2.3 Ant Colony System (ACS)

The complex social behaviors of ants have been much studied by science, and computer scientists are now finding that these behavior patterns can provide models for solving difficult combinatorial optimization problems. Ant Colony System (ACS) consists of a set of cooperating agents called ants that cooperate to find a good solution for optimization problem on graphs similar to the Travel Salesman Problem (TSP, such as this airflights case). Each single ant reflects a very trivial behavior: it simply goes from a node to another across an arc, but when all ants cooperate, like actual ants do in a real colony, the whole system reveals an intelligent behavior, as mush as it is able to find a good solutions for the TSP [12].
To compare the ACS with Dijkstra algorithm and Artificial Neural Network (ANN) is shown as table 1. The Dijkstra algorithm is a divergent function and can't input various arguments to inference. ANN [11]is a strong convergence function and can support inference by many arguments, but it is inadaptable in dynamic environment such as this airflights case. In this case, the weather sometime becomes bad accounted for cancelling the airflights.

Table. 1. The compare of various algorithms

	ACS	Dijkstra	ANN
Various Arguments	O	X	O
Dynamic Environment	O	O	X
Strong Convergence	O	X	O

3. Systematic Structure

The Multi-Services Semantic Search System (MS^4) is a five-tier system, shown as Fig. 3, Mobile Users (MUs) can utilize various terminal devices that include PC, notebook, Tablet PC, Personal Digital Assistant (PDA), and mobile phone to access Semantic Web Services Server (SWSS) through various web browsers. The UDDI Registries (UDDIRs) such as JUDDI offer UDDI standard APIs which are Inquiry API and Publication API for Service Providers (SPs) and SWSS (as Service Requests). SPs are many kinds of various businesses which provide some services to publish to UDDIRs. There are Intelligent Agents (IAs), Model Base System (MBS) in the SWSS. There is the collecting of user's requirement information, geographical information, and multimedia's file in the database server. Relevant system functions design as follows.

3.1 Mobile Users (MUs)

Mobile users (MUs) provide the functions that include Adaptable Multi-Services (AMS), Customized Service, Heterogeneous Networks.

3.1.1 Adaptable Multi-Services (AMS)

MUs use mobile device to request their requirements to MS^4, in order to carry on the inference of Multi-Services by SWSS using MBS and Semantic Inference Module (SIM). In this airline case, Offering relevant adaptable air flights to users who can choose difference routes in accordance with their situation (such as cost-oriented or time-oriented), and MS^4 makes reservation automatically according user request. Model of MBS depend on Semantic Inference Module (SIM), Adaptive Airflights Inference Module (AAIM), and other extedsion module to estimate. The SIM inference user's requirement and calculate QoS cost of web services (such as price-cost or time-cost) by Fuzzy logic. The AAIM offers QoS-Based Services Selected Method (QBSSM) using Ant Colony System (ACS) to provides adaptable air flights (such as multi-services) on different user's situation by QoS cost of web services (as cost of every section).

3.1.3 Heterogeneous Networks (HN)

Users use in the different network protocol , so the designing of system lets the terminal device or mobile equipment can be integrated services such environments as GSM, GPRS, IEEE802.11x wireless network, etc..

3.2 UDDI Registries (UDDIRs)

The aim of semantic web is to locate services automatically based on the functionalities Web services provide. UDDI is helpful to discovery Web services with semantic web. Therefore, we use the JUDDI to build UDDI environment which provides Business Entities, Service Entities, Binding Templates, and tModels to represent the detail of business and its services. Services in JUDDI can be searched by name, by location, by business, by bindings or by tModels. However, JUDDI doesn't support any inference based on the taxonomies referred to by the tModels. Integration of semantic web and JUDDI will solve this problem.

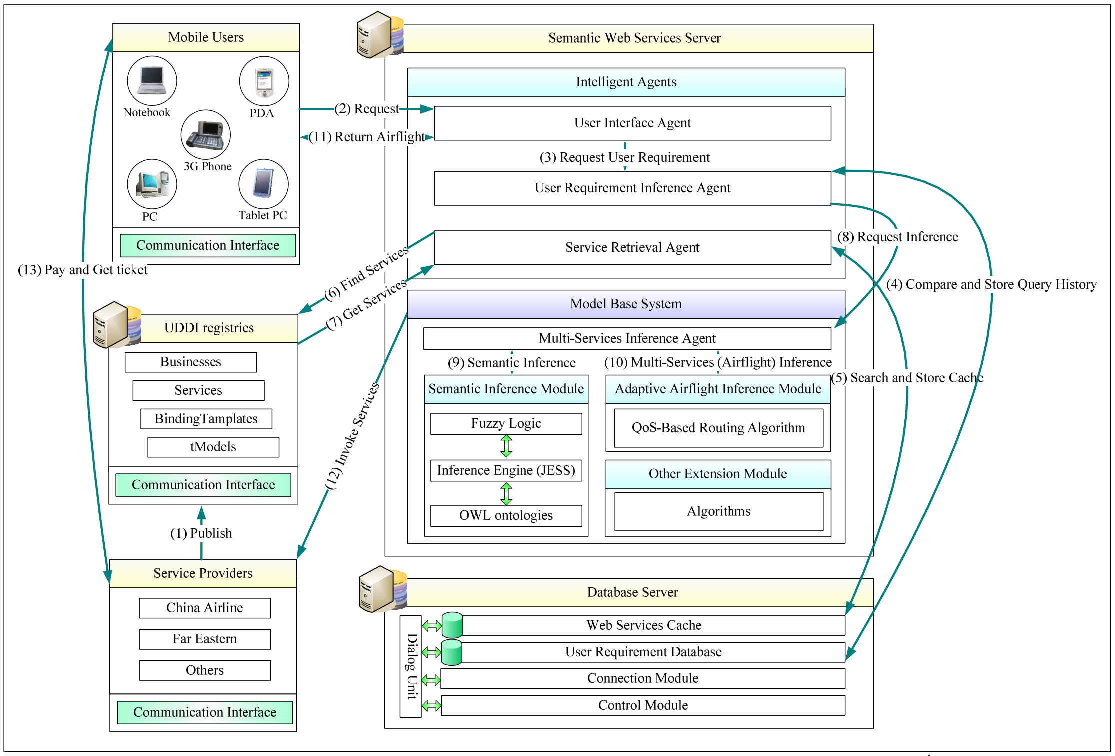

Figure 3. The architecture of Multi-Services Semantic Search System (MS[4])

In order to inference user's requirement semantically, Anton Naumenko ,Sergiy Nikitin, Vagan Terziyan and Jari Veijalainen propose the mapping method of transforming subject-predicate-object triples of OWL-S into tModels, which is shown in Fig. 4. After mapping, Service Retrieval Agent (SRA) can retrieve the detail and relationship of those services in JUDDI by UDDI4J APIs for the semantic inference easier.

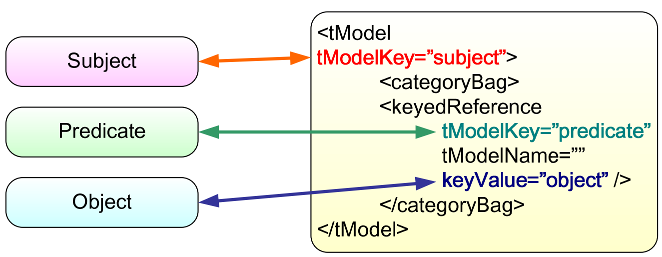

Figure 4. Mapping from triples to tModels

3.3 Service Providers (SPs)

Service Providers (SPs) build SOAP environment such as AXIS2 to provider some services for user invocation. After building services, SPs can publish the information of business, services, and binding templates to UDDIRs. For security, we can modify the AXIS2 API (such as upload.jsp) to build the hash code of service by MD5 algorithm. In this airline case, SPs which can publish the air flight service to UDDIRs through heterogeneous networks are China Airline, Far Eastern Air, and etc..

3.4 Semantic Web Services Server (SWSS)

The Semantic Web Services Server (SWSS) offers the relevant services of multi-services semantic search, those services compose of the Intelligent Agents (IAs) and Model Base System (MBS).

3.4.1 Intelligent Agents (IAs)

The IAs proceed such function as collection of the materials, searching, classifying, dealing with work, etc., the work can let users get the most multi-services automatically. The intelligent agent system includes User Interface

Agent (UIA), User Requirement Agent (URA), and SRA.

(1) User Interface Agent (UIA)
To know that user's equipment type, when the users login in and give them the proper webpage.

(2) User Requirement Inference Agent (URIA)
To collect the user's requirement, such as query, operation, search history, and canned query, the information will be transmitted to the Multi-Services Inference Agent (MSIA) the DS in order to let the inference engine to analyze and recommend in advance.

(3) Service Retrieval Agent (SRA)
In traditional, the semantic web combined UDDI takes a long time to do the hierarchical queries such as find_business(), find_service(), find_binding(), and find_tModel(). Therefore, we design the SRA to separate service information of huge quantity in UDDI to the Web Services Cache (WSC) in Database Server (DS), in order to save the time for accessing various UDDIRs by complex queries while MBS analyze the user's requirements. SRA which is allowed an accelerated lookup process for finding the best match for users and their requirements is powerful to reduce the UDDI query processes to provide a brilliant performance in the multi-services inference.

3.4.2 Model Base System (MBS)

The Model Base System (MBS) includes intelligent deduction engine that uses Data Mining technology to produce the inference. First, the multi-services are established automatically by the system, and the Multi-Services Inference Agent (MSIA) will recommend information to mobile user for relevant services. The MBS provides MSIA, Semantic Inference Module (SIM), Adaptive Airflights Inference Module (AAIM), and other extension modules. The SIM is combined fuzzy logic, inference engine, and OWL-S ontologies to inference QoS of web services value by user's requirements. And the AAIM uses the QoS-Based Services Selected Method (QBSSM) to inference the adaptive airflights which are combined various services.

(1) Multi-Services Inference Agent (MSIA)
Multi-Services Inference Agent (MSIA) receives the user's requirement from URIA and builds the communication with Semantic Inference Module (SIM), Adaptive Airflights Inference Module (AAIM), and other extension modules for inference Multi-Services. First, MSIA forwards the requirement information to SIM, and SIM will calculate the QoS costs of each relevant web services. Second, MSIA sends those QoS costs and the information of web services to request AAIM or other extension modules. Final, when those modules return the results , MSIA will recommend Multi-Services and invoke these services after user's submission.

(2) Semantic Inference Module (SIM)
The Semantic Inference Module (SIM) exploits fuzzy logic and ontology to explain and to represent the data of Web services. There are four steps in SIM:
a. First, Fuzzy classifier asks for the Web Service Caches (WSC) that were inserted and updated properties of web services in UDDIRs by SRA.
b. Fuzzy Classifier uses OWL-S to explain the each Web service in WSC.
c. In inference engine such as OWLJessKB, it defines fuzzy terms and related membership functions that are based on the data schema of web services.
d. Based on the OWL-S interpretation, fuzzy classifier asks inference engine to calculate the fuzzy value (as QoS cost) of web services for different fuzzy terms.

In this case, we use ontology to explain the meaning of slot name in the databases. Based on airline ticket data, we define three fuzzy terms (such as "cheap", "medium", "expensive") related with ticket fare to classify Web services. Generally, people evaluate ticket fare with three different levels, the cheap fare, or the medium fare, or the expensive fare. Therefore, we use the three degrees for ticket fare as fuzzy terms [4]. The membership functions related with three fuzzy terms mentioned previously are shown in Fig. 5.

We make use of protégé_3_3_beta as the tool for editing ontology. The meaning of each slot name resided in airline database is defined by OWL-S. After building OWL-S document for interpreting the database of airline service, we use the following query in OWLJessKB to tell if the input slot name is the subclass of one class in OWL-S document. For example, based on the OWL-S definition, the input slot name "value" is the subclass of "price", and price is related with three fuzzy terms, "cheap", "medium" and "expensive". Therefore we can obtain the three fuzzy values (as QoS cost) for "cheap", "medium", and "expensive" as fuzzy terms to describe Web service. Fig. 6 shows the query with OWL-S. And then QoS cost can be modified to combine different fuzzy terms by other fuzzy operations such as addition, subtraction, multiplication, division, log, and etc.

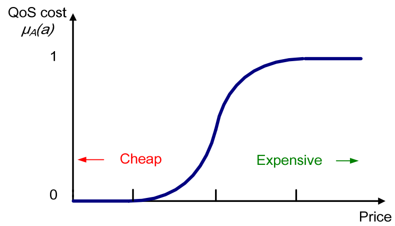

Figure 5. Fuzzy Terms for Price

```
(deftemplate triple (slot predicate) (slot subject) (slot object))

(defquery query-for-superclasses
        (declare (variables ?x))
        (triple
                (predicate "http://www.w3.org/2000/01/rdf-schema#subClassOf")
                (subject ?x)
                (object ?y))
)
```

Figure 6. OWL-S Query in OWLJessKB

(3) Adaptive Airflights Inference Module (AAIM)

Adaptive Airflights Inference Module (AAIM) provides QoS-Based Services Selected Method (QBSSM) algorithm to inference adaptable airflights (such as Multi-Services) which are like to solve the Travel Salesman Problem (TSP). In this airline case, there are many airline companies to provide various airflights (such as web services in UDDI) in each airport (such as routing node in TSP), we uses QoS-Based Services Selected Method (QBSSM) of AAIM to inference by QoS costs (such as each cost of arc in TSP) calculated in below and return the results to MSIA.

QBSSM consists of a set cooperating agents called ants that cooperate to find a good solution for optimization problems on graphs similar to the TSP. The QBSSM floods searching ants to all its adjoining and QoS satisfied nodes. Every adjoining node selects out the best QoS cost one due to an early-setted judgmental function from all the arrive ants, then copies and floods it again. The ants go on flooding until it gets to the destination node at last [6, 9]. The QBSSM algorithm is described as follow.

a. Creating ants: Each ant includes ant_id, source_node, destination_node, search_routing, routing_fittness, QoS_cost, and success. The ant_id is identification number of each ant. The search_routing which is the routing from source_node to destination_node inference by routing_fittness. The routing_fittness is calculated by degree of QoS cost which includes price-cost, time-cost, and etc through fuzzy inference. Final, the success will be set when ant gets success routing.

b. Source node behaviors: This node creates many ants in all their individual adjoining and QoS satisfied nodes. And then it waits and stores for the ants which get success routing coming back. We will select the best one of these routings to suggest adaptable airflights.

c. Medial nodes behaviors: If the ant is new, it will come back to source node when the medial node is destination node. And medial node adds node information into the ant and floods it to its adjoining and QoS-satifsfied nodes. If the ant is old, it will be dead when its QoS cost is lower than other ants' which are stored in medial node before. Else its QoS cost is better, the medial node will store it and update node information.

(4) Other Extension Modules

Other extension modules which can be called by MSIA are designed some algorithms into to them such as QBSSM for inference. The different algorithms individually receive the QoS costs of web services from SIM to inference adaptable Multi-Services automatically for different user's situations.

3.5 Database Server (DS)

The database server includes Web Service Caches (WSC), User Requirement Database (URD), connection module, and control module. The server also offers the integrated web services properties and user requirements to store, and it is a powerful application tool to provide information to SWSS for multi-services inference.

4. Case Study and Evaluation

The airline ticket reservation as one kind of Web services is provided by various airline companies on the internet. MS^4 helps travelers find adaptable airline tickets for their traveling plans. Generally, users want to find airline ticket reservation services (such as a single service) through UDDI or the current matchmaking for web services. In MS^4, the system will recommend adaptable airflights (such as Multi-Services) to MUs. The proposed method which is shown as Fig. 3 is applied to solve this problem according to the following procedures:

Step 1: Many airline companies will provide their airline services on themselves SOAP site (such as AXIS2) and publish the information of those services which include company name, flight number, department time, department city, arrive time, arrive city, price, and etc. to UDDIRs based on JUDDI.

Step 2: When MUs inquire the SRA for Multi-Services as the adaptable airflights through UIA, they send their requirements as a part of the request.

Step 3~4: The UIA will send the MUs' requirement to URIA. URIA supported the processes include lexical

analysis, stemming algorithms, indexing, and searching will check and store user's information in URD for inference user's requirements.
Step 5~7: The SRA holds up-to-date information on offers currently available for a group of services which have been requested recently. To keep offer lists up-to-date, the SRA inquires the one or more UDDIRs periodically regularly in order to check, find, and get for new offers.
Step 8: The URIA will search and get relative services from SRA and send them to MSIA. When MSIA receives the user's requirement inference result from URIA, it will control and coordinate various modules in MBS.
Step 9: The SIM will inference user's affinity information by S-membership function combined fuzzy logic according to user's requirement from URIA. If user wants to have a cheaper travel from Taipei to Beijing at 2007/10/30 08:00, the SIM will test the available services and increase the weight of price to calculate QoS cost to be used by AAIM. The QoS cost of each services is shown as table 3, table 4, and table 5.

Table 3. The QoS cost of airflights from Taipei to Beijing
(Exchange Rate: 1 TWD = 0.2372 HKD = 0.2298 MCY at 2007/10/21)

Company	Flight NO.	Departure time	Departure City	Arrive time	Arrive City	Price (TWD)	QoS cost
EVA Airway	BR851	10/30 08:15	Taipei	10/30 15:20	Beijing	20522	0.2551
Cathay Pacific	CX407	10/30 08:30	Taipei	10/30 15:20	Beijing	24150	0.3832
China Airline	CI641	10/30 08:50	Taipei	10/30 15:20	Beijing	26621	0.4889
Cathay Pacific	CX467	10/30 08:55	Taipei	10/30 15:20	Beijing	24150	0.3832

Table 4. The QoS cost of airflights from Taipei to HongKong
(Exchange Rate: 1 TWD = 0.2372 HKD = 0.2298 MCY at 2007/10/21)

Company	Flight NO.	Departure time	Departure City	Arrive time	Arrive City	Price (TWD)	QoS cost
China Airline	CI 603	10/30 08:15	Taipei	10/30 10:00	HongKong	3820	0.0010
EVA Airway	BR851	10/30 08:20	Taipei	10/30 10:00	HongKong	8700	0.0081
Cathay Pacific	CX407	10/30 08:30	Taipei	10/30 10:15	HongKong	4200	0.0015
China Airline	CI 641	10/30 08:50	Taipei	10/30 10:35	HongKong	3820	0.0013
Cathay Pacific	CX471	10/30 08:55	Taipei	10/30 10:45	HongKong	4200	0.0018
EVA Airway	BR865	10/30 09:15	Taipei	10/30 10:55	HongKong	8700	0.0119
Cathay Pacific	CX403	10/30 09:25	Taipei	10/30 11:20	HongKong	4200	0.0022
China Airline	CI 605	10/30 10:00	Taipei	10/30 11:45	HongKong	3820	0.0020

Table 5. The QoS cost of airflights from HongKong to Beijing
(Exchange Rate: 1 TWD = 0.2372 HKD = 0.2298 MCY at 2007/10/21)

Company	Flight NO.	Departure time	Departure City	Arrive time	Arrive City	Price (TWD)	QoS cost
Air China	CA108	10/30 10:40	HongKong	10/30 13:55	Beijing	8704	0.0241
Cathay Pacific	CX6108	10/30 10:40	HongKong	10/30 13:55	Beijing	17119	0.1308
Cathay Pacific	CX6880	10/30 12:00	HongKong	10/30 15:20	Beijing	17119	0.1621
Air China	CA102	10/30 12:45	HongKong	10/30 16:00	Beijing	8704	0.0326

Step 10: To search the adaptable airflights (Multi-Services), the AAIM will find the adaptable routing from department city to arrive city by QBSSM according to nodes (such as each airports) and path (such as QoS cost). QBSSM creates many flooding ants from various nodes to search aim node by QoS cost shown as Fig. 7. We can find the adaptable airflight and save eight thousand dollars for MUs by Multi-Services which is better than single service. To compare those results of single service and Multi-Services is shown as Fig. 8 and Fig. 9.
Step 11~12: MS^4 returns the result and recommends the adaptable airflights to MUs. If MUs agree this suggestion, MS^4 will make those airflights tickets reservation automatically.
Step 13: MUs will pay the money to those airline which are booked airflight tickets and get those tickets and bills.

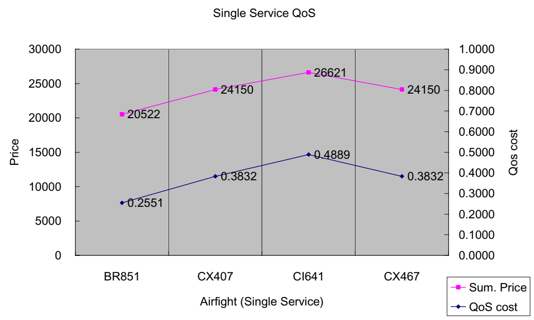

Figure 8. The adaptable airflights inference by QBSSM

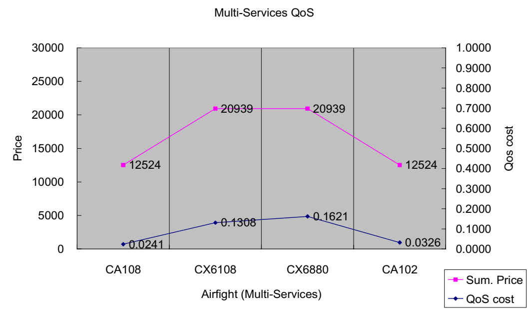

Figure 9. The adaptable airflights inference by QBSSM

5. Conclusion

In this research, we proposed a Multi-Services Semantic Search System (MS^4), which provides user's requirements inference and relative services search by semantic inference engine and find the best services composition to recommend Multi-Services. In this airflight case, we discover the result Multi-Services is better than traditional single service when MUs consider price-oriented services first. MUs can save more than eight thousand TWD from Taipei to Beijing. MUs also can conveniently obtain customized Multi-Services and decision to get and use those services according to their requirement in advance by MS^4.

Acknowledgments

The research is supported by the National Science Council of Taiwan under the grant No. NSC 96-2416-H-009-008-MY3.

References

[1] A. Kaufmann, and M. M. Gupta, "Introduction to Fuzzy Arithmetic. Theory and Application", pp. 2–35 and 69–72, New York, 1991.

[2] Chen Zhou, Liang-Tien Chia, Bu-Sung Lee, "DAML-QoS Ontology for Web Services", Proceedings of the IEEE International Conference on Web Services (ICWS'04), pp.472-479, Washington, DC, USA, 2004.

[3] Doris Janssen, Alexandre Lins, Thomas Schlegel, Michael Kühner, Gerhard Wanner, "A Framework for Semantic Web Service Retrieval", The 3rd Nordic Conference on Web Service – Proceedings (NCWS 2004), pp. 1-14, 2004.

[4] Kuo-Ming Chao, Muhammad Younas, Chi-Chun Lo, Tao-Hsin Tan, "Fuzzy Matchmaking forWeb services", The 19th IEEE International Conference on Advanced Information Networking and Applications (AINA 2005), pp. 28-30, Taipei, Taiwan, 2005.

[5] M. Brian Blake, Amy Lynn Sliva, Michael Zur Muehlen, Jeffery V. Nickerson, "Binding Now or Binding Later: The Performance of UDDI Registries", The 40th Annual Hawii International Conference on System Sciences (HICSS'07), pp. 171-177, Hawii, USA, 2007.

[6] Mikkel Bundgaard, Troels C. Damgaard, Federico Decara og Jacob W. Winther, "Ant Routing System", Spring 2002 IT University of Copenhagen–Internet technology, 2002.

[7] M. Tian, A. Gramm, H. Ritter, Jeffery J. Schiller, "Efficient Selection and Monitoring of QoS-aware Web Services with the WS-QOS Framework", The IEEE/WIC/ACM International Conference on Web Intelligence (WI'04), pp. 152-158, Beijing, China, 2004.

[8] Ora Lassila, James Hendler, "Embracing Web 3.0", IEEE Internet Computing, Vol. 11, No. 3, pp. 90-93, 2007.

[9] Sui-Xian Wei, Sheng-Hua Huang, "An ant colony system algorithm rooted in Dijkstra stategy", Computer Applications, Vol. 25, No. 12, pp. 2908-2910, 2005.

[10] Tim Berners-Lee, "Information Management: A Proposal", CERN, March 1989.

[11] Trenton Haines, Juraj V. Medanic, "A Neural Network Shortest Path Algorithm", IEEE International Symposium on Intelligent Control, pp.382-387, Columbus, Chio, USA, 1994.

[12] Ugo Chirico, "A Java Framework for Ant Colony Systems", Technical report, Siemens Informatica S.p.A, 2004.

[13] Xianjun Geng, Ram D. Gopal, R. Ramesh, Andrew B. Whinston, "Scaling Web Services with Capacity Provision Networks", Computer, pp. 64-72, Vol. 32, No. 11, 2003.