A Web Service Selection Mechanism based on User's Preferences

Chin-Chih Chang and Wen-Wei Tsai
Department of Computer Science and Information Engineering
Chung Hua University
Hsinchu City, Taiwan

***Abstract*—With the fast development of Web services more and more Web services are available. The conventional issues of Information/knowledge discovery has evolved into service discovery. There is a need to discover, select and compose suitable services that meet user's requirements.**

The existing service discovery technologies are mainly keyword-based search and less emphasize on user's preferences. We believe user's preferences are the key factor for the decision on the service selection. In this paper we present a Web service selection mechanism based on user's preference. The mechanism enhances the existing Web service registry (UDDI) with the sorting, ranking, and feedback capability. In each step the user's preferences are taken into consideration.

A system is built based on the mechanism and the feasibility of the mechanism is validated by a travel plan example. The performance of the mechanism is also analyzed. It shows the system is feasible and effective.

Keywords-Web Service; UDDI; Service Discovery; Service Selection; Quality of Service

I. Introduction

Web Services are currently the de facto technology for service-oriented computing (SOC) which is widely adopted in academic and industrial software development. With the fast development of Web services more and more Web services are available. There is a trend that the data-oriented Web is migrating to the Service Web [1]. This migration has led to the need for Web services discovery and selection. To search and select a suitable Web service that can meet user's requirements from the numerous Web services has become an important research topic.

The existing service discovery technologies are mainly keyword-based search which cannot fit user's need well [2]. In a lot of business operations, users' preferences are often the decision factor. In order to offer Web services that satisfy users' preferences we present a user-centric mechanism for Web service selection. We use the existing Web service registry and integrate with the sorting, ranking, and feedback mechanisms which we proposed to construct a Web service system with emphasize on user's preferences. When a user wants to search and select a web service, she or he can access the system interface to accomplish the task. In the ranking mechanisms, we adopt two algorithms: the rule-based and weighting methods, to select a suitable Web Service and recommend to the user. Finally, we use the feedback mechanism to adjust the selection weighting for services. When a user is using or after using the system, the system will collect the rating values for the service from the user. These feedback values are used to adjust the weighting of Web Services dynamically to provide the user with the adaptable Web Services.

We use the tourist service as an example to validate the feasibility of our system. We will show how a user utilizes our system to plan a trip according to user's preferences. Finally, we will analyze the algorithms that we present, and compare their advantages and disadvantages. And we also analyze our system and compare with other research work.

II. Service Discovery and Service Selection

Web service discovery and selection is a complex process. There is still no clear distinction between discovery and selection. Service selection begins with discovery. In this paper we adopt the view that discovery is referred as the activities related to identifying the functional properties of user's requests and selection as nonfunctional properties [3]. Crasso et al. further enumerate four functional and four nonfunctional criteria [4]. The discovery process is mainly keyword-based search which returns a list of candidate services and the selection process is a refinement of the discovery process [5]. For the recent survey on service discovery and selection interested readers can refer to [4].

Functional criteria are mainly used to match against the specifications offered by the service provider. QoS (quality of service) is commonly used to indicate nonfunctional criteria [6]. In this paper, we focus on nonfunctional QoS because they are often the decision factor. With the development of Web services research, various QoS for Web services have been identified and can be classified into the following categories: runtime-related, transaction support related, configuration management and cost-related QoS, security-related QoS, and user-related QoS [7, 8]. From our point of view, user-related QoS is the most important factor in these nonfunctional characteristics because users are always the ones who make the final decision. Hence, we focus on Web service selection in terms of user's preferences.

A. Web service discovry

The goal of Web service discovery is to find appropriate Web services that match user's functional requirements. Web service discovery through UDDI (Universal Description, Discovery, and Integration) is the most basic discovery method. UDDI provides only a category-based browsing and keyword-based matching discovery service [8]. Discovery by UDDI is simply a process of matching the WSDL (Web Service Description Language) based on keyword and category. After matching keyword the possible services are replied but the most appropriate service is not retrieved now and then. Therefore, quite a number of approaches have been proposed to enhance UDDI. Four main categories of these approaches have been specified: information retrieval-based, QoS-aware, semantic-based, and highly scalable and available [4]. The discovery process usually finds a list of candidate services. To further identify the most suitable service for the requestor selection process is conducted.

B. Web service composition

One of powerful and highly expected capabilities is service composition. Especially for a complex goal if a single service cannot fulfill the request, the service composition is activated. Usually, the service composition is undertaken by decomposing the complex goal into a sequence of simple sub-goals. Each sub-goal can be fulfilled by a single service. The main issue of the service composition is how to decompose the task and then select suitable services to complete the task effectively and efficiently. For example, the task of booking a hotel in a foreign country can be decomposed into two tasks: exchange rate conversion and hotel booking as shown in Figure 1. Once two tasks are fulfilled, the complete task is completed.

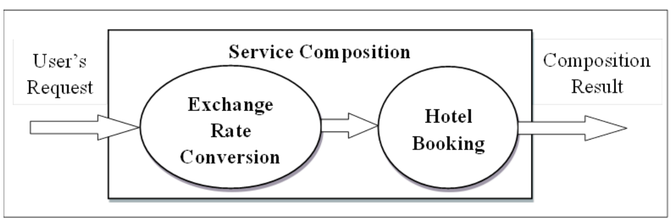

Figure 1. Example of service compostion

Basically, the service composition strategies are classified as static and dynamic composition based on the time of service composition [9]. Static composition takes place during design time while dynamic composition takes place at run time. A composition mechanism must satisfy four requirements: connectivity, nonfunctional QoS properties, correctness, and scalability [10]. The effective dynamic Web service composition is still a highly complex and challenging task [11]. Most current solutions are either too theoretic or only suitable to some specific situations. Much effort is continuously devoted to Web service composition. For a recent survey interested readers can refer to [12].

C. Web Service selection

Web service selection is a process to select the most suitable service from a list of candidate services after service discovery or composition mainly based on nonfunctional properties. Singh and Huhns pointed out three categories of Web service selection strategies: semantic service selection, social service selection, and economic service selection [13]. Semantic service selection finds a match based on the semantic description. Social service selection uses social rating such as reputation, recommendation, referrals, etc. to select a service. Economic service selection uses cost related information to choose a service. The main activities of service selection contain (1) matching nonfunctional service request, (2) evaluation of service offerings, and (3) result aggregation [8]. After discovery a list of candidate services is presented. First, services are matched against the nonfunctional specifications. Secondly, service offering is evaluated. Lastly, the result is aggregated.

Though quite amount of work has been dedicated to Web service selection, a solution that is more practical, less costly, and more universal is still now there. Web service selection is still a research topic for the time being. For a recent development interested reader can refer to [14].

Most selection practice is a composition of semantic, social, and economical strategies. From our point of views social service selection is the key factor. In this paper we will focus on user's preferences in social selection because the user is the final one who makes the decision.

III. System Architechture

This section introduces the system architecture and its modules. The system architecture as shown in Fig. 2 contains four modules: query interface, UDDI repository, discovery and selection module, provider interface, and as follows.

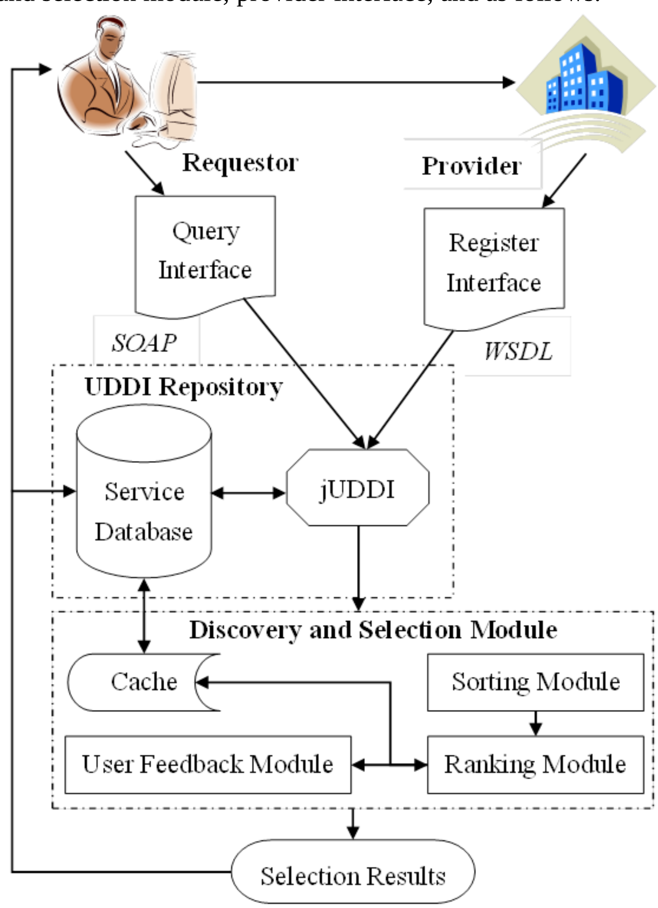

Figure 2. System architecture

- Query Interface: A Web-based interface for a user to query and invoke the service.
- UDDI repository: This module contains a UDDI registry and a database as the service repository. We implement the registry using jUDDI which is an open source Java UDDI implementation.
- Discovery and selection module: This module enhances UDDI with sorting, ranking, and feedback capability. Sorting filters out those services whose attributes are quite different from those of the request. Ranking ranks those services based on similarity to user's request. The feedback module collects user's rating for each service features, analyzes the ratings, and adjusts the weighting of services. The details of the mechanism of this module will be illustrated in the next section. This is the core contribution of this work.
- Provider Interface: A Web-based interface that facilitates service providers to register their services.

IV. Methodology of Service Selection

The core part of the proposed system is the discovery and selection module which is composed of three modules: sorting, ranking, and feedback. As shown in Figure 3 after the user's request from the query interface is collected, the user's selected features and preferences are analyzed. After the sorting and ranking process the results are produced. The user is asked if she or he is satisfied with the result. If the answer is yes, the service is invoked. Otherwise, the ranking process will be rerun based on either adjusting the priority of selection rules or the weighting of selected services. The process will repeat a number of times until the user is satisfied.

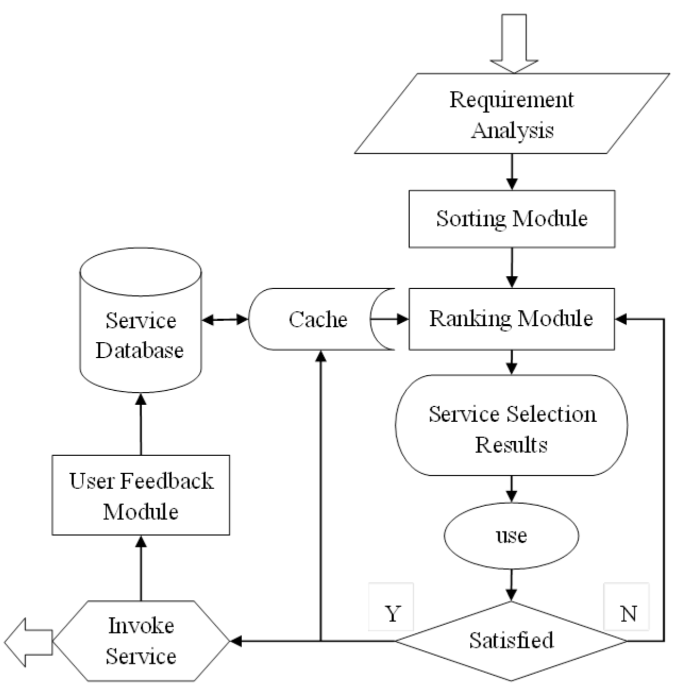

Figure 3. Flowchart of the service discovery and selection.

The major work of the sorting module is to sort all Web services in the service repository by user's preferences and select those more suitable services and filter out those suitable services. In this module user's request (U_{req}), all available services (WS_{all}), and service attributes (WS_{i_attr}) are the input.

When the data (UA_{pre}) of requirement analysis is received, user's preference weighting is acquired. The services will be sorted by the level of how the service meets the user's requirement from high to low. The lower ones are excluded from the next step of refinement, ranking. The algorithm is shown in Figure 4.

Sorting Algorithm

$U_{req} \leftarrow \{ur_1, ur_2, ur_3, \dots, ur_n\}$ // User's requests
$WS_{all} \leftarrow \{ws_1, ws_2, ws_3, \dots, ws_m\}$ // All services
$WS_{i\_attr} \leftarrow \{wsc_1, wsc_2, wsc_3, \dots, wsc_l\}$ // Service attribute
$i \leftarrow 0$
for each ur_i
    $UA_{pre}[] \leftarrow analyze(ur_i)$
end
$i \leftarrow 0, j \leftarrow 0$
//Send to Sorting Module
for each ur_i in UA_{pre}
    for each wsc_j in WS_{i_attr}
        if $ur_i = wsc_j$ then $WS_{all}[] \leftarrow sort(wsc_j)$
        end
    end
end
//Dismiss low level of WS_{sort}
$m \leftarrow b$
$WS_{sort} \leftarrow WS_{all}$
//Output of Sorting Module
Return WS_{sort}

Figure 4. Sorting Algorithm

After sorting the services are further ranked. We proposed a priority-based method which ranks sorted services based on the attribute priority. First a list of services with higher score on the attribute which the user designates as the first priority is selected. Then the similar process will go through the next attribute until the last attribute. In this way the sorted service in the previous step will be ranked. The ranking process is shown in Figure 5 and the algorithm is shown in Figure 6.

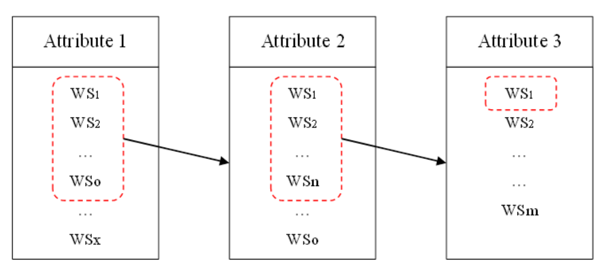

Figure 5. Proity-based ranking process

The last step is the feedback module. After the service is used, an interface is provided to the user. The user can specify the satisfaction rating for each service attribute. These rating is collected by the feedback module and used for adjusting the service selection weighting. In this way the selection weighting can be dynamically adjusted and will produce the results that fit the user's requirements better. The algorithm is shown is Figure 7.

Priority-based Ranking Algorithm

$WS_{sort} = \{ws_1, ws_2, ws_3, \ldots, ws_m\}$ // Sorting results
$U_{req} = \{ur_1, ur_2, ur_3, \ldots, ur_n\}$ // User's requests
$WS_{i\_attr} = \{wsc_1, wsc_2, wsc_3, \ldots, wsc_l\}$ // Service attributes
$i \leftarrow 0$
for each ur_i
    $UB_{pre}[] \leftarrow analyze(ur_i)$
end
//*Send to Ranking Module*
$i \leftarrow 0, j \leftarrow 0, k \leftarrow 0$
for each ur_i in UB_{pre}
    for each ws_j in WS_{sort}
        for each wsc_k in WS_{i_attr}
            if $wsc_k = ur_i$ then $WS_{rank}[] \leftarrow ws_j$
        end
    end
end
//Select high level WS in WS_{rank}
$WS_{rank} \leftarrow WS_{sort}$
//Output of Ranking Model
Return WS_{rank}

Figure 6. Proity-based ranking algorithm

Feedback Algorithm

$WS_{i\_attr} = \{wsc_1, wsc_2, wsc_3, \ldots, wsc_l\}$ // Selected attribute
//User fills the feedback interface system to get U_{ws}
Get $U_{ws} = \{rws_1, rws_2, rws_3, \ldots, rws_n\}$ // User's feedback
for each wsc_i in WS_{i_attr}
    // Recalculating service attribute
    average $\leftarrow$ compute(U_{ws}, WS_{i_attr})
    $WS_{attr} \leftarrow$ average // Updating the attribute in database
end
Cache $\leftarrow ws_i$ // Storing in Cache

Figure 7. Feedback algorithmExperiments and results

V. Experiment and Results

This section describes the system development, service provision, and experimental results.

A. System and service development and nterfaces

This system is built on an Intel Core Quad Q8300 2.5GHz CPU, 4GB DDR RAM with Microsoft Windows Server 2008. JSP is selected as the programming language execute on Glassfish v3 Web server.

To verify the system we develop a travel plan example which includes three categories of services: accommodation, meal, and activity. We register 53 accommodation, 40 meal, and 57 activity services as shown in Table 1, 2, and 3 respectively. There are total 150 services.

Table 1. Categories of accomodation service

Categories	Room capacity	Location	Price	Service Features (Attributes)
Luxury Hotel Motel Log cabin Resort Bed & Breakfast Camping	1 2 4	Kenting Hualian/ Taitung Green Island Surrounding Islands	Based on the listed price	Service Cleanliness Room quality Fire safety Convenience Popularity Total satisfaction Combined average

Table 2. Categories of meal service

Categories	Location	Price	Service Features (Attributes)
Barbecue Quick meal Hot pot Exotic meal Highly exquisite Local specialty Dish set	Breakfast Lunch Dinner	Based on listed price	Service Cleanliness Atmosphere Meal quality Popularity Total satisfaction Combined average

Table 3. Activity classification

Categories	Service Features
Leisure	Popularity
Sightseeing	Total satisfaction
Nautical activity	Combined average
Excitement	
Backpacking	

The system main functions include activity planning, travel information enquiry, and customer feedback. A user can either queries a single function or a set of functions which can fulfill the user's request. When the system is asked to organize a trip, the activities, accommodation, and meals will be queried in the same time. The results from each query join together to form a travel plan for the user as shown in Figure 8.

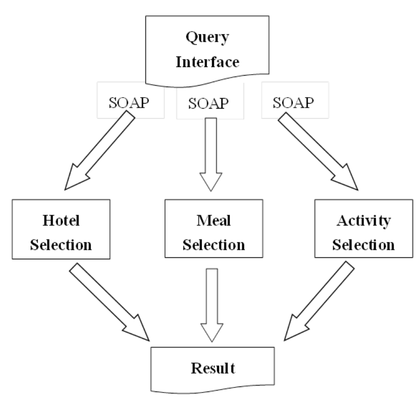

Figure 8. User query process

Figure 9 shows the Web service selection interface. A user can specify their preferences through this interface (Web page). These data are sent to the system for analysis and produce the result. Figure 10 shows the results after going through our proposed mechansim.

Figure 9. Web service selection interface

Choose Trip Composition Service

ID:0	trip_Paintball	trip_Swimming	trip_Kart racing	trip_Jet ski
	Satisfaction : 4.8 Service Popularity : 77.0	Satisfaction : 1.9 Service Popularity : 61.0	Satisfaction : 4.0 Service Popularity : 53.0	Satisfaction : 3.7 Service Popularity : 44.0
ID:1	trip_Swimming	trip_Kart racing	trip_Jet ski	trip_Bungy jumping
	Satisfaction : 1.9 Service Popularity : 61.0	Satisfaction : 4.0 Service Popularity : 53.0	Satisfaction : 3.7 Service Popularity : 44.0	Satisfaction : 2.8 Service Popularity : 32.0
ID:2	trip_Kart racing	trip_Jet ski	trip_Bungy jumping	trip_Sport utility vehicle
	Satisfaction : 4.0 Service Popularity : 53.0	Satisfaction : 3.7 Service Popularity : 44.0	Satisfaction : 2.8 Service Popularity : 32.0	Satisfaction : 1.6 Service Popularity : 28.0

Figure 10. Query results

B. Experiment analysis and discussion

In this experiment we use MAP (mean average precision) as a measure to evaluate the effectiveness of the system. The basic idea is that a service with a higher rank will have a higher MAP value. The equation is shown as follows:

$\text{MAP} = \frac{\sum_{i=1}^{N}\frac{i}{\text{Rank i}}}{N}$ where N is the total number of selected services and Rank_i is the rank of the ith service.

For example, if a user requests three services and the system finds three services located in the rank first, second, and fifth. The MAP value is 0.866 where $(\frac{1}{1}+\frac{2}{2}+\frac{3}{5})/3 = 0.866$.

We used 150 Web services as shown in Table 1, 2, and 3 to verify our system. We invited 30 students as participants. To minimize the subjective problem, we exclude the 5 values that are too low or too high. We let the participants try the selection with our proposed mechanism and then the original UDDI mechanism which is without sorting or ranking. In such a sequence the effect of the user's familiarity with the system is minimized. The result is shown as in Figure 11 and Figure 12. It shows our proposed mechanism it better than the original mechanism.

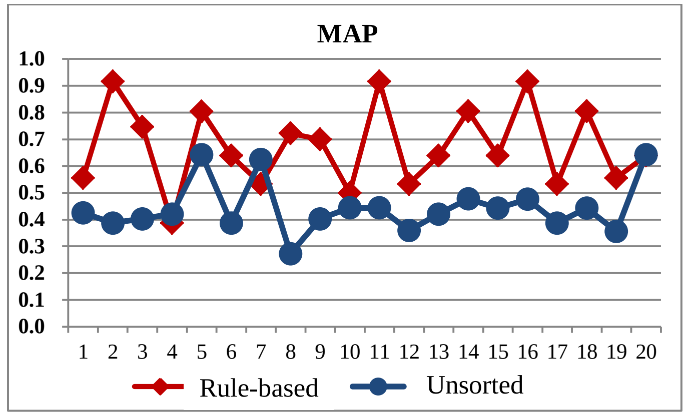

Figure 11. MAP comparison between selection with rule-based ranking and without sorting

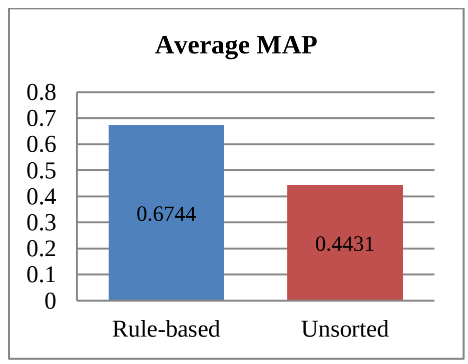

Figure 12. Average MAP comparison between selection with rule-based ranking and without sorting

VI. Conclusion and Future Work

In this paper we present a method for Web service selection that focuses users' preferences. The main objective is to devise a mechanism that takes user's preferences into consideration in every step of the service selection process. In addition, the users' feedback is collected to adjust the selection weighting of the services. To verify our mechanism a system is built and a travel plan example is analyzed. The result shows our mechanism is functional and feasible.

Currently, we are increasing the number of services and enhancing mechanism with fuzzy logics or neural networks. We are also conducting a comparison with other similar system. The other issue is to address the preferences for each user rather than the average of all users' preferences.

Acknowledgment

This work is supported by the National Science Council of Taiwan under the grant number NSC 99-2218-E-216-001.

References

[1] C. Petrie and C. Bussler, “Service Agents and Virtual Enterprises: A Survey”, IEEE Internet Computing, vol. 7, no. 4, pp. 68-78, 2003.

[2] Qi Yu and A. Bouguettaya, “Guest Editorial: Special Section on Query Models and Efficient Selection of Web Services”, IEEE Transactions on Services Computing, vol. 3, no. 1, pp. 161-162, 2010

[3] A. Carenini, D. Cerizza, M. Comerio, Emanuele D. Valle, F. De Paoli, A. Maurino, M. Palmonari, M. Sassi, and A. Turati, “Semantic Web Service Discovery and Selection: a Test Bed Scenario”, Proceedings of the 6th International Workshop on Evaluation of Ontology-based Tools and the Semantic Web Service Challenge (EON-SWSC-2008), Tenerife, Spain, June 1-2, 2008.

[4] Marco Crasso, Alejandro Zunino, Marcelo Campo: A Survey of Approaches to Web Service Discovery in Service-Oriented Architectures. J. Database Manag. Vol. 22 no. 1, pp. 102-132, 2011

[5] Wolf-Tilo Balke and Matt Wagner, “Towards Personalized Selection of Web Services,” WWW (Alternate Paper Tracks) 2003, Budapest, Hungary, 20-24 May 2003.

[6] Hei-Chia Wang, C-S Lee, and Tsung-Hsien Ho, “Combining Subjective and Objective QoS Factors for Personalized Web Service Selection,” Expert Systems with Applications, vol. 32, no. 2, pp. 571-584, 2007

[7] S.P. Ran, “A Model for Web Services Discovery with QoS. ACM SIGecom Exchanges”, vol. 4 no. 1, pp. 1–10, 2003.

[8] H.Q. Yu and S. Reiff-Marganiec, "Non-functional Property based service selection: A survey and classification of approaches", Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop co-located with The 6th IEEE European Conference on Web Services , November 12, 2008, Ireland, Dublin.

[9] W. Schreiner and S. Dustdar, “A Survey on Web Services Composition,” International Journal of Web and Grid Services, vol. 1, no. 1, 2005, pp. 1–30.

[10] N. Milanovic and M. Malek, "Current solutions for Web service composition," Internet Computing, IEEE , vol.8, no.6, pp. 51- 59, Nov.-Dec. 2004.

[11] J. Rao and X. Su. "A Survey of Automated Web Service Composition Methods". In Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition, SWSWPC 2004, San Diego, California, USA, July 6th, 2004.

[12] D. A. D’Mello, V.S. Ananthanarayana, and S. Salian, “A Review of Dynamic Web Service Composition Techniques,” Communications in Computer and Information Science, Vol. 133, Part 2, pp. 85-97, 2011.

[13] M. P. Singh and M. N. Huhns, “Service-Oriented Computing: Semantics, Processes, Agents,” Wiley, 2005.

[14] Qi Yu and Athman Bouguettaya, “Foundations for Efficient Web Service Selection,” Springer, 2010