FUNCTIONAL AND NON-FUNCTIONAL QoS AWARE WEB SERVICE DISCOVERY

B. Aruna Gandhi¹, S. Hemalatha¹, R. Sai Saranya¹, K. Prem Kumar² & P. Iyappan²

¹UG Student, Department of Computer Science Engineering, Sri Manakula Vinayagar Engineering College, Puducherry.
²Research Scholar, Department of Computer Science Engineering, Manonmaniam Sundaranar University, Thirunelveli, Tamil Nadu.
(E-mail: aruna3191@gmail.com, hema.siva21@gmail.com, saisaranyak@gmail.com)

Abstract: Web services are becoming widespread nowadays. Determining the desired Web services is becoming an emergent and challenging research problem in many of the enterprise business applications. In this paper, we propose a Web service discovering approach which compares functional similarities to users’ queries, and non-functional QoS characteristics of Web services to obtain normalized result based on Qos metrics. Services are searched based on keywords and user preferences. The similarity computation will be designed to improve the functional accuracy of searching result. Finally, the integration of both functional and non-functional evaluation component will lead to quality information of Web services and user will be provided with appropriate services.

Keywords: Web Services, Web Service Discovery, Qos, Non-functional Qos characteristics.

1. INTRODUCTION

UDDI have enabled service providers and requestors to publish and find Web services through Registries. Discovery of Web services is the fact that existing search exploit keyword-based search techniques which may not be suitable for choosing Web services. So, there is a need for a mechanism that can distinguish between Web services using well-defined criteria such as considering Quality of Service (QoS) attributes. Differentiating between Web services that share similar functionalities is significantly achieved by examining non-functional Web service attributes such as response time, throughput, availability, usability, performance, integrity. The Web Service publisher can have different approaches and levels of providing security depending on the subscribers’ request. Thus Web Services (WS) technology[1] is becoming increasingly popular because of its potential in many fields, such as loose coupling, heterogeneous services, interaction and dynamic sharing, and it allows the existence of interoperation between software that are provided by the different providers, running on the different operating systems and written by the different programming languages. Thus the QoS delivered to a customer is highly affected by various factors such as the performance of the Web Service itself, the hosting platform, and the underlying network.

Figure 1: Register-Find-Bind Model for Web Services Discovery

Figure 1 shows the model for register-find-bind services is adopted by WS currently. UDDI (Universal Description, Discovery and Integration) [2] is the services registry mechanism of the WS system, which provides only functional services; WSDL (Web Service Description Language) is an important component of WS technology, which describes the abstract-definition interfaces and ports of implementation for the distributed service operation in the Internet environments. The adoption rate has been very slow mostly because
of its discovery mechanism, which is based on the functional aspects of the desired Web Service, and lacks non-functional guarantee, i.e. QoS (Quality of Service) [3] guarantee. QoS determines availability and efficiency of services, which greatly affect popularity of certain services. QoS means a certain level (not necessarily the highest level, but the lowest) of service that can meet the needs of service consumers. There is an agreement between the consumer and the provider, which is used to describe what level should be reached which can be accepted by both sides.

2. OVERVIEW OF WEB SERVICE DISCOVERY

2.1. Discovery of Web Services
The Web service discovery for the service discovery request and the matchmaking process is summarized below.

1. The service request is preprocessed.
2. The action list, qualifier list, object list and noun list of the functional knowledge are searched to get the corresponding identifiers. Unavailability of any identifier results in discovery failure.
3. On locating required identifiers from the functional knowledge, the operation pattern for the discovery request is formed.
4. The abstract Web service information of all published Web services is searched by traversing Service Operation Time for the requested operation identifier and all the Web services having requested operation (operation identifier) are returned to the requester as the functionally suitable Web services.

As a consequence of this rapid growth and the abundance of service providers, quality of service is becoming a key aspect in Web Services competition. The consumer is faced with the inevitability of selecting the right service provider. In such scenarios, the quality of service becomes essential to differentiate services and providers. Quality of service forms the keystone for a broad range of Web Service activities such as admission control, server selection, scheduling, pricing, and specifying service level agreements.

2.2. Evolution of Web Service Discovery Techniques
In literature, different methods are proposed to capture syntactical, semantical and contextual similarity of discovery request and service advertisements for the discovery. We classify the Web service discovery methods into two broad categories as functional property based and non-functional property based approaches. In functional property based Web service discovery, the functional i.e. behavioral details of Web services are used for matchmaking of discovery request with the published i.e. advertised profiles of Web services. The non-functional criterions like QoS [4], trust [5], usability [6] and personalization [7] are also used for effective Web service discovery. We classify functional property based discovery mechanisms based on the nature of information retrieved or stored for the matchmaking. They are syntax based matching, behavior based matching and semantic matchmaking methods.

The syntax based Web service discovery mechanisms normally find the matching information for the request from the WSDL documents of advertised services. The matchmaking is commonly based on keyword/string, service category, interface and input/output of the published Web services. The behavior based matching mechanisms discover Web services based on the internal process structure of the service or service requester’s behavior. The semantic based matchmaking mechanisms find Web services based on the service semantics. The matchmaking is mainly performed based on the semantic descriptions like information theoretic information [8], functional semantics [9], Ontological concepts [10] context information [11] and goal (Input, Output, Pre-condition, Effect) [12] of advertised Web services.

3. QUALITY OF SERVICE
The term quality of service (QoS) originated in the areas of telecommunications, distributed multimedia, and networking. QoS refers to a
collection of qualities or characteristics of a service, such as availability, security, response time, throughput, latency, reliability, and accuracy.

Three main approaches are considered to provide QoS support in Web Services:

1. **UDDI extension approach**, which consists to extend the current UDDI data structure with QoS information of a Web Service. QoS parameters, such as response time and throughput, are associated with service descriptions.

2. **SOAP extension approach** which consists to extend the SOAP header to include QoS information and

3. **Deployment of an independent brokerage service** in which the QoS broker is a service, or component. These approaches may be combined together for better QoS support in Web Services. Efforts are as well carried out to add further structure, specifically QoS constraints, to Web Services descriptions through the use of ontologies.

### 3.1. QoS Criteria

QoS criteria for different domains may be different. In order to map services into the system, we only consider more generic criteria: response time (RT), Throughput (TP), Availability (AV), Accessibility (AC), Interoperability Analysis (IA), and Cost of Service (C). QoS parameters help determine which of the available Web services is the best and meets clients’ requirements.

1. **Response Time (RT)**: The time taken to send a request and receive a response (Unit: milliseconds).

2. **Throughput (TP)**: The maximum requests that are handled at a given unit in time (Unit: requests/min).

3. **Availability (AV)**: A ratio of the time period when a Web service is available (Unit: %/3-day period).

4. **Accessibility (AC)**: The probability a system is operating normally and can process requests without any delay. (Unit: %/3-day period).

5. **Interoperability Analysis (IA)**: A measure indicating whether a Web service is in compliance with a given set of standards. (Unit: % of errors and warnings reported).

6. **Cost of Service (C)**: The cost per Web service request or invocation (cents per service request).

### 3.2. Classification of QoS

A classification according to the characteristics of QoS is as follows [4, 13, and 14].

1. **Computational behavior**: These QoS includes Execution Attributes (Latency, Accuracy, Throughput, Reliability, Extendibility, Capacity, and Exception Handling), Privacy (such as Encryption, Authentication, and Authorization), Secrecy, Availability etc.

2. **Business behavior**: These QoS mainly refers to Execution Cost, Reliability of the provided service.

3. **Metadata restriction**: These QoS includes Constraints That has to be followed regarding UDDI /WSDL /SOAP parameters such as location, specific companies, schema etc.

As more and more Web Services appear on the web service consumers are presented with a group of service providers offering similar services. Different service providers may afford different QoS. This will require sophisticated patterns of negotiation. To address the problem above, this paper proposes a QoS-aware model for Web Services discovery based on the analyses of foregone models. In the new model, QoS is taken as constraints when searching for Web Services, which would give some confidence to the service consumers about the availability and efficiency of services.

### 4. PROPOSED SYSTEM ARCHITECTURE

Now we describe the system architecture of our QoS aware Web service search engine. As shown in Figure 2, after accepting a user’s query specification, our search engine should be able to provide a practical Web service recommendation list.
The search engine consists of three components: nonfunctional evaluation, functional evaluation, and QoS-aware Web service ranking. There are two phases in the nonfunctional evaluation component. In phase 1, the search engine obtains QoS criteria values of all the available Web services. In phase 2, the search engine computes the QoS utilities of different Web services according to the constraints and preferences specified in the QoS part of the user’s query. The functional evaluation component contains two phases. In phase 1, the search engine carries out a preprocessing work on the WSDL files associated to the Web services. This work aims at removing noise and improving accuracy of functional evaluation. In phase 2, the search engine evaluates the Web service candidates’ functional features. These features are described by similarities between the functionality specified in the query and the functionality of operations provided by those Web services. Finally, the search engine combines both functional and non-functional features of Web services in the QoS-aware Web service ranking component. A practical and reasonable Web service recommendation list is then provided as a result to the user’s search query.

5. QoS-AWARE WEB SERVICE SEARCHING

5.1. Qos Model
In our QoS model we describe the quantitative nonfunctional properties of Web services as quality criteria. These criteria include generic criteria and business specific criteria. Generic criteria are applicable to all Web services like response time, throughput, availability and price, while business criteria are specified to certain kinds of Web services. By assuming m criteria are employed for representing a Web service quality, we can describe the service quality using a QoS vector \( (q_1, q_2, \ldots, q_i, \ldots, q_m) \), where \( q_j \) represents the \( j \)-th criterion value of Web service \( i \). Some QoS criteria values of Web services can be obtained from the service providers directly. However, other QoS attributes’ values like response time, availability and reliability need to be generated from all the users’ invocation records due to the differences between network environments. In this paper, we use the approach proposed in [16] to collect QoS performance on real-world Web services. By putting all the Web services QoS vectors together, we can obtain the following matrix \( Q \). Each row in \( Q \) represents a Web service, while each column represents a QoS criterion value.

\[
Q = \begin{pmatrix}
q_{1,1} & q_{1,2} & \ldots & q_{1,t} \\
q_{2,1} & q_{2,2} & \ldots & q_{2,t} \\
\vdots & \vdots & \ddots & \vdots \\
q_{s,1} & q_{s,2} & \ldots & q_{s,t}
\end{pmatrix}
\]

5.2. Utility Function
A utility function is used to evaluate the multidimensional quality of a Web service. The utility function maps a QoS vector into a real value for evaluating the Web service candidates. To represent user priorities and preferences, two steps are involved into the utility computation. First, the QoS criteria values are normalized to enable a uniform measurement of the multidimensional quality of service independent of their units and ranges. Second, the weighted evaluation on criteria needs to be carried out for representing user’s constraints, preference and special requirements.

5.3. Normalization
In this step each criterion value is transformed to a real value between 0 and 1 by comparing it with the maximum and minimum values of that
particular criterion among all available Web service candidates. The maximum value $Q_{\text{max}}(k)$ and minimum value $Q_{\text{min}}(k)$ of $k$th criterion are computed as follows:

$$Q_{\text{max}}(k) = \max_{j \in [1, n]} q_{j, k}$$

$$Q_{\text{min}}(k) = \min_{j \in [1, n]} q_{j, k}$$

The normalized value of $q_{i,j}$ can be represented by $q'_{i,j}$ as follows:

$$q'_{i,j} = \frac{q_{i,j} - Q_{\text{min}}(k)}{Q_{\text{max}}(k) - Q_{\text{min}}(k)}$$

Thus, the QoS matrix $Q$ is transformed into a normalized Matrix $Q'$ as follows:

$$Q' = \begin{bmatrix}
q'_{1,1} & q'_{1,2} & \cdots & q'_{1,t} \\
q'_{2,1} & q'_{2,2} & \cdots & q'_{2,t} \\
\vdots & \vdots & \ddots & \vdots \\
q'_{s,1} & q'_{s,2} & \cdots & q'_{s,t}
\end{bmatrix}$$

Using this normalized matrix $Q'$ we provide the user desired services with some of its QoS parameters computation results explicitly through which the user will realize the availability and efficiency of the corresponding web services.

6. COLLABORATIVE FILTERING MECHANISM

We present a collaborative filtering approach for predicting QoS values of Web services and making Web service recommendation by taking advantages of past usage experiences of service users. Then, based on the collected QoS data, a collaborative filtering approach is designed to predict Web service QoS values. This filtering can also predict active user’s interest for a target item based on his interest. The majority of existing item-based collaborative filtering approaches emphasizes the personalized factor of recommendation separately, but ignores the user’s general opinions about items. The active user will more likely prefer similar items to those which he has preferred previously. Each user rates items that they have experienced, in order to establish a profile of interests. The collaborative filtering system then matches together that user with people of similar interests.

For a given active user, rating prediction for the target unvisited item is generally a weighted sum of the observed ratings for his visited items similar to the target, in which the importance of each neighbor item is weighted by its similarity respectively. The active users who contribute more Web service QoS data will obtain more accurate QoS value predictions. Predicting the change of QoS accurately in filtering can effectively reduce the web services. Filtering decisions are based on human and not machine analysis of content.

Collaborative filtering provides three additional advantages:

1. Supports for filtering items whose content is not easily analyzed by automated processes
2. Ability to filter items based on quality and taste.
3. Ability to provide serendipitous recommendations.

The user’s rating calculation starts by gathering the initial rating from the users. If the ratings exist for a particular item, the user’s average rating will be calculated by dividing the sum of all user ratings by the number of users who have rated that particular item. The user’s average rating is then stored in the rating database and will be used for rating recommendation and for the calculation of user’s prediction rating. The user’s rating prediction will be calculated when the user ratings do not
exist for a particular item. To calculate the prediction rating, the recommendation engine will retrieve both the items’ similarity and its corresponding user average rating, and divide the product between them with the sum of the items’ similarity. The prediction ratings are then stored in the rating database. Once the document has received ratings from users, the prediction rating will be overwritten and the user’s average rating will be recommended. The final stage in the recommendation process is to recommend the user rating for the recommend top (items with the highest similarity) similar items. For this purpose, the recommendation engine will query the item’s similarity from the item-similarity database and based on the item’s similarities (that exceed a threshold value), the top documents will be retrieved from the database. Concurrently, the user rating for the viewing item is retrieved from the rating database to be recommended to the user.

7. RELATED WORK
The problems pertaining to Web service discovery have long been taking attention of both academia and industry the majority of the approaches relies on keyword search which is based on only functional requirements that shows services in a simple list format. The location of Web services is difficult as it does not provide an efficient interface for querying services. The name of a Web service, a Web service provider must be known to get further details. The keyword search takes only names into account and ignores service descriptions. And moreover discovery is a timeconsuming effort. Web service resources provide functionality to determine whether a service is active or not. Currently, most of the work is in the description of Web services, the syntax of their flows, and how they could be executed. In the future, it is necessary to view Web services in the context of specifying, validating, and automatically synthesizing complex, reactive processes.

8. CONCLUSION AND FUTURE WORK
A Web service functional and non functional characteristics based on QoS parameters has been presented in this paper for the purpose of finding the best available Web service during Web service discovery process based on a set of given client QoS preferences. The use of non-functional properties for Web services significantly improves the probability of having relevant output results. The proposed solution has shown usefulness and effectiveness of incorporating both functional and non functional QoS parameters as part of the search criteria and in distinguishing Web services from one another during the discovery process. The ability to discriminate on selecting appropriate Web services relied on the client’s ability to identify appropriate QoS parameters. The proposed solution provides an effective Web service relevancy function that is used for finding most relevant Web services. For future work, we plan to extend QoS parameters to include information such as reputation, penalty rates, reliability, and fault rates.

References


