Efficient Discovery and Ranking of Web Services Using Non-Functional QoS Requirements for Smart Grid Applications

M.Suchithra\textsuperscript{a*}, M.Ramakrishnan\textsuperscript{b}

\textsuperscript{a}Research Scholar, Department of Computer Science & Engineering, Sathyabama University, Chennai, Tamil Nadu, India
\textsuperscript{b}Professor & Head, Department of Computer Science, Madurai Kamaraj University, Madurai, Tamil Nadu, India

Abstract

Web service discovery is the important process and traditional methods use keyword-based search to locate the appropriate web service. This method may not yield good results. Non-functional QoS parameters of a web service play a vital role in ranking a web service. Moreover, standards such as UDDI, WSDL and SOAP have potential of providing QoS based web service discovery. In this paper, a new method is proposed to rank the web services. The proposed method computes relevancy function value of the web services and normalization is performed. This normalized result is used to rerank the web service. The selection of appropriate web service provide user to access smart grid over internet. The experimental results show that the proposed method is effective in selecting appropriate web services than traditional methods.

Keywords: Web Service Discovery, Relevancy Function, UDDI, Ranking of Web Service

1. Introduction

UDDI has provided a platform for both requesters and publishers to find and publish web services of their interest. Publication of web service and searching are done through UDDI Business Registries (UBRs). UBR does not provide enough technology / support to search a relevant web service due to variety of reasons. One important among several reasons for the above problem is that existing APIs only exploit key-word based searching techniques[1]. This is not well suited for all the applications as many web services share common functionalities. By using appropriate web services, user can access smart homer (e.g. utility provider) to remotely access different elements of a smart home.

\* Corresponding author. Tel.: 919841721825
E-mail address: suchij24@gmail.com
User can remotely read the readings from a smart meter, control appliances, control temperature via thermostat. Hence web services are vital to remotely interact with smart home elements to manage energy consumption, in a smart grid environment. Moreover, APIs in UDDI allows publishing and discovery of data for a specific web service but does not provide an opportunity for a quality-based retrieval. And other difficulty in web service discovery process is that software vendors like IBM WebSphere, Microsoft, etc promoting their products and deploy their own internal UBRs for internet and intranet. When these UBRs are also integrated with web, a big complexity of finding appropriate web service arises. This leads to find a new automated process for web discovery process.

Since the information provided by the web services are technical, there is a great requirement for a technique that can categorize web services based on well-defined criteria like QoS attributes. By examining the non-functional attributes of a web service, it is highly possible to differentiate between different web service with same functionality[2]. It would be feasible if we modify the existing web service discovery technique so that it can include QoS parameters during discovery process.

Several researchers have proposed methods for including QoS in web service discovery process. Jyothishman Pathak et al [3] presented a framework for semantic web services discovery. The presented frame work is used as ontology-based semantic web service discovery. It depends on user supplied context – specific mappings from user ontology to relevant domain ontologies. In this frame work, user query is transformed in to a query that can be processed by match making engine. The frame work also incorporates non functional requirements (QoS) in web service discovery mechanism. Personalized ranking criteria, specified as part of service request, ranks the discovered candidate services. This criterion enhances the traditional ranking approach.

A.Pradnya Khutade and B.Rashmi Phalnikar [4] presented a paper on QoS based web service discovery using object oriented concepts. The paper considers not only functional requirements, but also non-functional requirements, that describes the quality of service during web service discovery process. The proposed frame work uses the combination of UML and OWL. The idea is to use object oriented concepts such as inheritance, operation and polymorphism together with ontologies to discover appropriate web service.

Hong Xia Tong and Shensheng Zhang [5] presented a multi-attribute decision making algorithm for web services selection based on QoS. The proposed fuzzy multi attribute decision making algorithm selects the most appropriate web services with the highest degree of membership belonging to positive ideal solution. This selection is based on QoS, with static present.

Eyhab Al-Mastri and Qusay H.Mahmoud [6] presented a paper on discovering the best web services. The major challenges in web discovery are multiple / heterogeneous registries, services sharing the common functionalities and provision to customers to customize discovery process. The above issues are addressed in this paper. The paper uses Web Service Relevancy Function (WSRF) that measures the relevancy ranking of a particular web service based on QoS metrics and client preferences.

From the above, it is evident that several researches have been already done on web service ranking based on QoS constraints. But many of the above methods depend on the service providers to supply their QoS metrics. The problem with this arrangement is the reliability of the QoS metrics supplied by the service provider. It would be better to have third party authentication to manage the QoS information supplied by the service provider.

Here the duty of the third party is to check for the integrity of the QoS values and also to supply the missing metrics i.e. the values that are not provided by the service provider. Hence this paper aims to address the above issues and proposes a new mechanism to build web service repository by offering quality-driven discovery of web services. And also re-ranking of web services using non-functional QoS constraints. The paper is organized as follows: The paper starts by elaborate discussion about the need for appropriate web service selection in section 1. In section 2, provides related work for web service selection using QoS. The proposed method is presented in section 3 and section 4 provides results and discussion. Paper ends by providing a lucid conclusion about the proposed method.
2. Related Work

Generally web services may have similar functionalities but may have different non-functional properties. Hence it is necessary to consider both functional and non-functional properties during web service discovery process. Traditional UDDI standard does not include QoS properties[7]. To overcome this problem, one idea is to embed QoS information within the message. An API that can integrate QoS properties through user defined functions is needed. But UDDI provides very limited level of QoS support.

Another way to associate QoS properties during web service discovery process is to use Quality Requirements Language (QRL), which is a XML-based language that can be used for web service discovery. But QRL cannot be directly associated with WSDL. Another problem with QRL is that it does not provide information about controlling QoS properties[8]. Other methods include creation of middleware for improving web service composition, selection and monitoring of QoS properties for web services. These methods fail to provide desired accuracy.

3. Relevancy Function

Web service discovery mechanisms discussed in the above section lacks in certain requirements. The pitfalls are, (a) there is no provision to administer and maintain QoS properties in UBRs dynamically, (b) there are no provisions to estimate the reliability of the QoS values supplied by the service provider, (c) estimating QoS values is not open and transparent, (d) time periods for evaluating QoS parameters, (e) no specific data format for QoS values, and (f) upgradation of UDDI and WSDL to support QoS properties.

Hence we propose herewith a quality-driven method for ranking web services with QoS properties. Our method contains Web Service Crawler Engine (WSCE) that creep active UBRs.

A web service manager, which can measure QoS of web services are stored in Web Service Storage(WSS). Relevancy Function measures the relative ranking of each web service after getting client’s request. The QoS parameters are used to find the relevancy value of each web service[9]. Six QoS parameters have been selected to estimate the relevancy value and they are

- Throughput – Web service’s ability to handle maximum request at a given time
- Availability – Web service’s availability to give response
- Response time – The unit of time taken to send a request and to receive a response usually represented in milliseconds
- Interoperability – Compatibility of a web service to incorporate set of given standards
- Accessibility – Probability of the system to react normally for a request without any delay
- Cost – Price to be spent to execute a web service

The proposed method works as follows. Client sends their request and relevancy function computes the relevancy value and then ranks the web services. A web service with the highest relevancy value is treated as most suitable for that web service request. If two or more web services have the same relevancy value and functionalities, ranking can be done using the below principle.

Let $WS=\{ws_1, ws_2, ws_3, \ldots ws_j\}$ be the set of web services and $P=\{p_1, p_2, p_3, \ldots p_j\}$ be the set of QoS properties. The set of web services and relevant QoS attributes are represented in a matrix called relevancy matrix with web services in rows and QoS attributes in columns. QoS based computational algorithm is used to evaluate which web service is most relevant using QoS properties. To select the appropriate web service, $j$ criteria is used in relevancy function matrix.
Since QoS parameters are estimated in different units, it has to be normalized so that uniform distribution of QoS values can be obtained[10]. Normalization also allows measurements to be associated with QoS attributes and client can effectively tune their QoS criteria. For normalization of a QoS parameter \( p_j \),

\[
N(j) = \sum_{i}^{m} q_{m,j}
\]

Where \( N \) represents normalized value of \( p_j \), \( q_{m,j} \) represents actual value from relevancy function matrix. Each element of the relevancy matrix is normalized with maximum QoS value using

\[
h_{i,j} = \frac{q_{i,j}}{\max(N(j))}
\]

Where \( h_{i,j} \) represents difference between maximum QoS value and \( q_{i,j} \). Applying the above equation, we get a new weighted matrix as shown below

\[
E = \begin{bmatrix}
  q_{1,1} & \cdots & q_{1,j} \\
  \vdots & \ddots & \vdots \\
  q_{i,1} & \cdots & q_{i,j}
\end{bmatrix}
\]

Now the relevancy function value is again calculated for the above matrix by using the below formula

\[
RF(\text{ws}_j) = \sum_{i=1}^{N} h_{i,j}
\]

Where \( \text{ws}_j \) represents set of web services with same functionality, \( h_{i,j} \) represents corresponding web service and QoS criterion, and \( N \) represents number of web services on a given set.

4. Results and Discussion

To test the effectiveness of the proposed method, actual implementation such as XMethods.net, XMLLogic and StriceIron.com are used. Six services were chosen which share the same functionality of validating an email address request. All the QoS parameters which were discussed above are used to test the QoS metrics. The QoS parameters are Response Time (RT), Throughput Time(TP), Availability (AV), Accessibility (AC), Interoperability (IA) and Cost (C). Table1 shows the average QoS values tested from two different networks.

<table>
<thead>
<tr>
<th>ID</th>
<th>Service Provider</th>
<th>RT</th>
<th>TP</th>
<th>AV</th>
<th>AC</th>
<th>IA</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>XML Logic Validate Email</td>
<td>716</td>
<td>5.64</td>
<td>80</td>
<td>82</td>
<td>78</td>
<td>1.1</td>
</tr>
<tr>
<td>2</td>
<td>XWebservices XWeb EmailValidation</td>
<td>1094</td>
<td>1.86</td>
<td>81</td>
<td>79</td>
<td>99</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>StrickIron Email Verification</td>
<td>706</td>
<td>11</td>
<td>96</td>
<td>97</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>StrickIron Email Address Validator</td>
<td>907</td>
<td>10.12</td>
<td>98</td>
<td>94</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>CDYNE Email Verifier EmailValidator</td>
<td>905</td>
<td>11</td>
<td>90</td>
<td>89</td>
<td>93</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>Webservicex Validate Email</td>
<td>1226</td>
<td>3.96</td>
<td>85</td>
<td>82</td>
<td>98</td>
<td>2.8</td>
</tr>
</tbody>
</table>
To find the appropriate web service, it is necessary to optimize the values for each QoS parameter as few metrics are given in percentage where as few in probability. Hence, the results need to be normalized. Table 2 gives the relevancy function value for each web service

<table>
<thead>
<tr>
<th>ID</th>
<th>Service Provider</th>
<th>Relevancy function Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>XML Logic Validate Email</td>
<td>3.6588</td>
</tr>
<tr>
<td>2</td>
<td>XWebservices XWeb Email Validation</td>
<td>3.2116</td>
</tr>
<tr>
<td>3</td>
<td>Strikelron Email Verification</td>
<td>4.6053</td>
</tr>
<tr>
<td>4</td>
<td>Strikelron Email Address Validator</td>
<td>4.1905</td>
</tr>
<tr>
<td>5</td>
<td>CDYNE Email Verifier</td>
<td>3.9196</td>
</tr>
<tr>
<td>6</td>
<td>Webserivces Validate Email</td>
<td>4.2629</td>
</tr>
</tbody>
</table>

To demonstrate our proposed method of ranking technique, we will consider four scenarios with different combinations of QoS requirements in each scenario. In the first scenario, response time (RT) is taken as an important QoS parameter. From table 1, it is clear that web service number three is having fastest response time of 0.706. In the second scenario, throughput value is taken as an important QoS parameter. From table 2, it is evident that again web service number three has highest throughput value i.e. 12.

In third scenario, response time and throughput are considered to be important QoS parameters and again web service number three is having less response time and high throughput value. And in the last scenario, both response time and cost are considered as important QoS criterion. In most of the cases, response time is the dominant QoS parameter than the cost parameter. Table 3 represents the rank of the four scenarios.

<table>
<thead>
<tr>
<th>ID</th>
<th>Test Scenario 1</th>
<th>Test Scenario 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RF Rank</td>
<td>RF Rank</td>
</tr>
<tr>
<td>1</td>
<td>0.535 2</td>
<td>0.445 3</td>
</tr>
<tr>
<td>2</td>
<td>0.355 5</td>
<td>0.295 6</td>
</tr>
<tr>
<td>3</td>
<td>0.886 1</td>
<td>0.819 1</td>
</tr>
<tr>
<td>4</td>
<td>0.425 3</td>
<td>0.345 4</td>
</tr>
<tr>
<td>5</td>
<td>0.425 4</td>
<td>0.328 5</td>
</tr>
<tr>
<td>6</td>
<td>0.315 6</td>
<td>0.476 2</td>
</tr>
</tbody>
</table>

From table 3, web service number three is ranked as first since its relevancy function value is higher in both test scenarios. This is because the QoS parameters such as response time, throughput, average delay, etc are very good. The web service number six is ranked as sixth in first test scenario and second in fourth test scenario. The reason behind this drastic change is that QoS parameters of web service number six is good except response time. Since fourth test scenario considers cost along with response time, it has been ranked as second instead of sixth.

5. Conclusion

In this paper, ranking of web services based on QoS parameters is presented. The objective is to find the best available web service for the given user request. Six dominant QoS parameters such as response time, throughput, availability, accessibility, interoperability and cost are considered for reranking process. It is obvious from the experimental results that the use of non-functional QoS parameters during web service discovery process improves the probability of selecting highly appropriate web services. The proposed method uses Relevancy Function, which acts as a basis for ranking web services. Since web services are easy and inexpensive, they can be used for controlling elements in smart grid environment.
References