Flexible Tradeoff Based Real Time Service Selection Process Based On Priorities and User Preferences

1Adlin Suji.K and 2Dr.S.Sujatha
1Assistant Professor, Department of MCA, Dhanalakshmi Srinivasan Engineering College, Perambalur, India - 621 212.
2Professor, Department of MCA, BIT Campus, Anna University, Trichy, India-620 024

ARTICLE INFO
Article history:
Received 3 September 2014
Received in revised form 30 October 2014
Accepted 4 November 2014

Keywords:
Service Selection; Tradeoff matrix; Prioritized selection; User Preference; AHP

ABSTRACT
Service selection is one of the major processes involved in building any online system. The availability of a huge number of web services complicates this selection process. The objective of the current paper is to provide an effective mechanism that retrieves the appropriate web services faster. These services hoisted by third party may not contain all the required quality attributes. Hence we also introduce a tradeoff mechanism that tends to provide appropriate tradeoffs in attributes during the selection process. QWS 2.0 dataset was used for analysis. Results show appropriate retrievals and low retrieval times.

INTRODUCTION
Service selection has become one of the major functionalities while building a system. This holds good for any type of online application due to the fact that most of the application designers prefer to utilize the available services rather than build their own custom services. This decision could be considered uncomplicated if the number of web services available are minimal. But due to the advancement in technologies and increased requirements, the number of web services available is huge and hence cannot be solved by traditional selection methods. Further, the requirement of users does not come in terms of a single value. Instead, the demands come in terms of quality values for each attribute provided by the service. Hence this becomes a Multi Criteria Decision Making (MCDM) problem. The requirement of any system solving this problem is that the result should satisfy all the demands of the user, and sometimes if the rigid requirements are not met, the system should also be flexible to introduce tradeoffs in the selection process. Further, time also proves to be a major constraint in such a system. A system that performs these processes faster and with best possible accuracy is the need for the current requirement scenario.

Several methodologies were proposed in literature that deals with web service selection and orchestration. An auction based approach dealing with quality aware service selection for service based systems is presented by Qiang et al. (2014). It deals with providing solutions for complimentary between services and competition among service providers. It provides approaches that provides solutions for providing offers to customers depending on the multi dimensional quality of services. A Quality of Experience (QoE) driven service selection process is presented by Bipin Upadhyaya et al. (2014). This method considers non functional attributes and provides them with equal importance alongside the QoS properties. An end to end QoS mapping and Aggregation for selecting services in a cloud is presented by Raed Karim et al. (2014). This paper stresses the use of non functional attributes along with their functional counterparts for enabling efficiency in the service selection process. It provides mapping rules to map user’s QoS requirements with service levels to perform an effective matchmaking process. a similar quality based service selection is presented by Neerja Negi and Satish Chandra (2014). This method uses a service negotiator for the selection process. A stability analyzer calculates the stability score of the web service and the services are ranked. Ranking is performed using AHP and TOPSIS.

A preference based semantic selection of web services is presented by Raluca Iordache et al. (2014). This method uses the conventional service selection process, but tends to incorporate tradeoffs in the selection process. a rigid rule cannot provide an effective selection mechanism, hence the flexibility introduced here tends to improve the overall selection process.

Corresponding Author: Adlin Suji.K, Assistant Professor, Department of Computer Applications, Dhanalakshmi Srinivasan Engineering College, Perambalur - 621 212.
Tel: +91-9442425880, E-mail: suji.adlin@gmail.com
A network aware QoS prediction for service composition is presented by Xinyu Wang et al. (2013). It uses the geolocation of the user to provide effective predictions. According to X. Wang, geographically similar routes tend to have similar latencies and bandwidth similarities. Hence this is also a prediction based approach that tends to improve the prediction efficiency and overall composition efficiency. A similar location based method is proposed by Xi Chen et al. (2014). In contrary to the previous approach, X. Chen uses the location information in their service selection process rather than in their routing process, a service is selected by the user based on their location information. User’s access details along with their location preferences are stored in the system. This data is then clustered and personalized preferences are obtained using region analysis.

**Tradeoff Based Prioritized Service Selection Process:**

A tradeoff based service selection process is proposed in this paper which uses a tradeoff incorporated elimination method to deliver the best possible service to the user. Research shows that using functional attributes alone for service determination is not sufficient. Non functional attributes should also be used for better efficiency. Hence our method of service selection uses both functional and non functional attributes for service selection. The proposed method is divided into two phases. The initial phase deals with service attribute determination and feature selection, while the next phase deals with evaluating the user preferences and tradeoff incorporation in the service selection process.

**Dataset Description:**

QWS 2.0 dataset (http://www.uoguelph.ca/~qmahmoud/qws/#Download_0) Al-Masri et al. (2007a, b) was used for evaluation of our method. The QWS dataset consists of 365 real web service implementations that currently exist on the Web. These services were collected using Web Service Crawler Engine (WSCE). Each service was tested over a ten-minute period for three consecutive days. The attributes available in the QWS dataset are described in the Table 1.

**Feature Selection:**

The process of service selection begins with attribute analysis. Services hoisted by different vendors tend to have common attributes. These tend to be the functional attributes of the hoisted service. The non functional attributes which are considered to the contributors in the current scenario should also be considered for analysis. Hence the number of attributes to be used tends to increase with time. This proves to be a downside when considering the process in terms of time. Removal of certain attributes tends to improve the performance of the system and it also tends to increase the accuracy of the overall selection process. This is performed in the initial phase.

Weka 3.7 was used for the process of attribute evaluation. Weka uses two algorithms for the process of Attribute Selection; the Attribute Evaluator and the Search Method. The evaluator search combination used in the current scenario is the ReliefF Attribute Evaluator and the Ranker method.

The ReliefF method Kenji Kira et al. (1992), Igor Kononenko et al. (1994), Marko Robnik-Sikonja et al. (1997) evaluates the worth of an attribute by repeatedly sampling an instance and considering the value of the given attribute for the nearest instance of the same and different class. The advantage of this approach is that it can operate on both discrete and continuous class data. The ranker method provides ranks to the attributes using their individual evaluations.

The result provided by this method returns all the attributes and their corresponding ranks ($R$). A user provided threshold ($\tau$) is considered as the baseline for the selection criteria. All the attributes whose ranks fall above $\tau$ are considered for evaluation, while those that fall below the threshold are eliminated.

**Table 1: QWS Attribute Description.**

<table>
<thead>
<tr>
<th>ID</th>
<th>Parameter Name</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Response Time</td>
<td>Time taken to send a request and receive a response</td>
<td>ms</td>
</tr>
<tr>
<td>2</td>
<td>Availability</td>
<td>Number of successful invocations/total invocations</td>
<td>%</td>
</tr>
<tr>
<td>3</td>
<td>Throughput</td>
<td>Total Number of invocations for a given period of time</td>
<td>invokes/second</td>
</tr>
<tr>
<td>4</td>
<td>Successability</td>
<td>Number of responses / number of request messages</td>
<td>%</td>
</tr>
<tr>
<td>5</td>
<td>Reliability</td>
<td>Ratio of the number of error messages to total messages</td>
<td>%</td>
</tr>
<tr>
<td>6</td>
<td>Compliance</td>
<td>The extent to which a WSDL document follows WSDL specification</td>
<td>%</td>
</tr>
<tr>
<td>7</td>
<td>Best Practices</td>
<td>The extent to which a Web service follows WSDL Basic Profile</td>
<td>%</td>
</tr>
<tr>
<td>8</td>
<td>Latency</td>
<td>Time taken for the server to process a given request</td>
<td>ms</td>
</tr>
<tr>
<td>9</td>
<td>Documentation</td>
<td>Measure of documentation (i.e. description tags) in WSDL</td>
<td>%</td>
</tr>
<tr>
<td>10</td>
<td>WsRF</td>
<td>Web Service Relevancy Function: a rank for Web Service Quality</td>
<td>%</td>
</tr>
<tr>
<td>11</td>
<td>Service Classification</td>
<td>Levels representing service offering qualities (1 through 4)</td>
<td>Classifier</td>
</tr>
<tr>
<td>12</td>
<td>Service Name</td>
<td>Name of the Web service</td>
<td>None</td>
</tr>
<tr>
<td>13</td>
<td>WSDL Address</td>
<td>Location of the Web Service Definition Language (WSDL) file on the Web</td>
<td>None</td>
</tr>
</tbody>
</table>
Let $S(a,R)$ be the service with attribute $a$ and rank $R$, then the probability that a particular attribute is selected for processing in the next level is

$$P_{aa} = \begin{cases} 1 & \text{if } R \geq \tau \\ 0 & \text{Otherwise} \end{cases}$$

The process of feature selection tends to reduce the overhead caused by unnecessary additional components that do not provide any impact in the final result. These attributes can be used as adjustable attributes (which will be described in the next section).

### User Preference Evaluation and Tradeoff Incorporation:

The web services available in the repository are usually provided with a service tag or name that identifies the type of service provided. Providing this service tag retrieves the services corresponding to it. Due to the huge availability of services in the service repository, the number of web services returned for a specific query is bound to be high. A short listing mechanism is mandatory to provide the user with the required result.

This short listing phase is done using two strategies. When a user requests for a service, he/she is also made to provide the QoE (Quality of Experience) parameters. QoE parameters Jonas Keppeler et al. (2014) are a combination of QoS parameters describing functional and non-functional requirements. A tradeoff matrix is obtained from the user in case of unavailability of a certain quality in an attribute. The tradeoff matrix in this paper divides the quality attributes in three sections; Fixed Constraints, Prioritized Constraints, Adjustable Constraints.

Fixed constraints, as the name implies, contains the attributes that are to be satisfied as specified by the user. They do not accept any tradeoffs in this section. If a fixed constraint is not satisfied (lower quality than required) by any of the services available in the repository, then the search result returns a null value or intimates the user to modify the constraint. If the repository contains services offering quality above or equal to the requirement, then those services are selected for the next phase (higher priority is given to the service that matches the requirement).

Let $C_i$ be the constraint provided for the $i^{th}$ attribute by the user, then the probability of selecting the service $P_{aa}$ is given by

$$P_{aa} = \begin{cases} 1 & \text{if } S_{qa} \geq C_i \\ 0 & \text{Otherwise} \end{cases}$$

Where, $S_{qa}$ is the quality of the attribute given by the service under consideration, and $C_i$ is the minimum requirement requested by the user.

Prioritized constraints contain a list of attributes in prioritized order. Analytic Hierarchy Processing (AHP) Saaty et al. (2008) is used for the process of prioritization. AHP is a structured analysis method for making complex decisions in a system. It has its basis in mathematics and psychology. Users are made to depict the importance of the attributes in a certain order by providing weights.

In this paper, the user is made to provide the weights using two methods; using pair wise comparison or as direct user assigned weights.

### Pair wise Comparison (Saaty et al., 2008):

The user is provided with every pair of attributes and is asked to rank them with respect to each other. This method is helpful if the user has a large number of attributes that they cannot keep track of, or if the user does not have intrinsic knowledge about each attribute. It becomes much easier to compare two attributes against each other rather than all the attributes. Comparisons are made on a 9 point scale and the comparison matrix depicting the priority set $P$ is as follows

$$P = \begin{bmatrix} w_1 / w_1 & w_1 / w_2 & \ldots & w_1 / w_s \\ w_2 / w_1 & w_2 / w_2 & \ldots & w_2 / w_s \\ \vdots & \vdots & \ddots & \vdots \\ w_s / w_1 & w_s / w_2 & \ldots & w_s / w_s \end{bmatrix} = \begin{bmatrix} 1 & s_{12} & \cdots & s_{1s} \\ s_{21} & 1 & \cdots & s_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ s_{s1} & s_{s2} & \cdots & 1 \end{bmatrix}$$

Where $W/W$ compares the attribute $x$ against $y$. $S_{xy}$ represents the comparison score of $x$ when compared with $y$. An attribute when compared to it will return a value of 1. These values are then integrated to provide the final attribute weights (WA).

### Direct User Assigned Weights (Saaty et al., 2008):

The user can also assign weight values for attributes directly rather than pair wise comparison.

After the assignment of weights and the quality requirements for attributes, the weighted sum method (8,9) is used to calculate the priority score ($P_s$) for the query.
The remaining attributes come under the adjustable constraints. These attributes are ordered according with their quality requirements in decreasing order and are used for analysis. Tradeoffs are usually introduced in the prioritized and adjustable constraints.

**Process Flow:**

The services retrieved from the repository are initially subjected to the Fixed Constraints. All the services that meet the quality requirements specified by the fixed constraints are taken to the next level. Search is made for perfect matches rather than higher quality services. Perfectly matching services are given priorities over the others (both higher and lower). The total number of services present in this level is taken as the highest priority value. Priorities are provided in decreasing order, where the highest priority service gets the maximum value as its priority number.

The next phase deals with selecting the appropriate web service by incorporating tradeoffs. The priority score (P₁) is used to determine the next level of web services. Values provided by the user using AHP are considered as weights and the weighted sum method Fishburn et al. (1967), Triantaphyllou et al. (2000) is used to determine the priority score (P₁) of all the shortlisted web services. The next level of short listing is performed by considering the priority score provided by the user (P₁) as the median and selecting services that contain the score within the range of 10 points from P₁, i.e., P₁-10 to P₁+10. The priority values in this level start from 20 by default. If the total number of shortlisted services is < 20 then the priority values start from the total count of available web services. Since we select only the top n services (where n≤20), the maximum priority count added to the priority score of the best service (service that meets all the requirements) will be 20. For services that are equal to P₁ a priority score of 20 is added, the services that offer higher than the required quality occupy the next level, while the services that offer quality below the required quality occupy the final level in the reduction phase. Hence it is made certain that the weight provided at this point has lesser importance than the weights provided by the fixed constraints.

The final level of short listing is performed using the adjustable constraints. For every attribute satisfying the constraint, provided by the user, 2 is added to the priority score of the corresponding service, and for every attribute providing a higher quality than the requirement, 1 is added to the priority score of the service. The web service comprising the highest priority score is selected for delivery to the user.

**RESULTS AND DISCUSSION**

The process was applied on QWS 1.0 and QWS 2.0 datasets, containing approx. 370 and 2500 rows respectively. The results depicted in Figures 1 and 2 shows the service retrieval times and the number of services that were actually present in the service repository corresponding to the query. It has been found that time consumed is less than a millisecond for every transaction. The maximum time consumption for queries comes to around 300ms, which also proves to be the stabilization value.

![Fig. 1: Retrieval Time (QWS 1.0).](image-url)
Fig. 2: Retrieval Time (QWS 2.0).

It can be noted that as the number of services increase, the retrieval time tends to increase, but after a particular threshold, it gets stabilized. The increase in time is due to the increase in the processing due to the retrieval of large number of results from the web service repository, while the stabilization can be attributed to the second phase which shortlists the services based on bounds (-10 to +10). This reduces unnecessary processing to a great extent. These stabilization values can be altered depending upon the system requirement. In case of time critical systems, this boundary can be reduced to provide faster results. It can also be fine-tuned according to user preference if the user has a good knowledge about the repository they are operating on. Further, the experiments revealed that our approach satisfied the requirements for services positively for 87.3% of the requests.

Conclusion:
The proposed methodology provides a promising scope for the service selection process. Due to the appropriate shortlisting of services at checkpoints, the system eliminates unnecessary services and concentrates on services that matter the most. Currently this system supports only expert users, with absolute knowledge of the service repository and the service names available. Working of our current system can be improved by incorporating semantic based service retrieval in the second phase. This will also facilitate novice users without any knowledge of the system to utilize the process efficiently. The service selection process is actually a part of the service composition process. Hence our future work will deal with incorporating this technology combined with semantic based retrieval as a part of web service choreography.

REFERENCES

Bipin Upadhyaya, Ying Zou, Joanna Ng, Tinny Ng, Diana Lau, 2014. Towards Quality of Experience driven Service Composition, IEEE 10th World Congress on Services, 978-1-4799-5069-0/14.


