Service Selection based on Customer Rating of Quality of Service Attributes

Abhishek Srivastava, Paul G. Sorenson
Department of Computing Science
University of Alberta
Edmonton, Canada
{*sr16, paul.sorenson*}*@ualberta.ca*

***Abstract*—Selecting the optimal service from a set of functionally equivalent services is non-trivial. Previous research has addressed this issue making use of Quality of Service (QoS) attributes of the candidate services. In doing this, researchers have however assumed that the customers' preference of the various QoS attributes varies linearly with the actual attribute values. In this work, we put forward a technique that overcomes this restriction and compares functionally equivalent services on the basis of the customers' perception of the QoS attributes rather than the actual attribute values. We utilize the 'mid-level splitting' method to track the customer's preference vis-a-vis the actual attribute values. Further, we utilize the 'Hypothetical Equivalents and Inequivalents Method' to assign weights, reflecting the importance, to the attributes on the basis of the customer preference. The whole procedure is demonstrated using a simple running example.**

I. Introduction

Major corporations the world over are realizing the efficacy of delivering their expertise as services rather than complete end products [1] [2]. Services as the delivery mode in business has a number of advantages. Often the risk factors involved in entering a new domain are relatively few owing to smaller capital investment upfront. The advent of cloud computing platforms, such as Amazon's 'EC2' [3], and 'RightScale' [4] make this risk even smaller, and also enable small-and-medium-sized players to enter the service sector. The 'margin for error' is much wider in the case of services as compared to products, which means that organizations dealing in services have greater liberty of recovering from mistakes and modifying the performance and quality of services. Thus, service providers can swiftly act upon a customer's response and feedback and make corrective changes. Furthermore, the penetration of the Internet in the daily lives of people acts as a foundational and inexpensive delivery medium for services. The service providers therefore package their respective expertise as web services which can be easily transported over the Internet. The customer also stands to gain from the service-oriented structure of the market. Customers have many more options from which to choose to get their work done. With services, customers also have a relative cost advantage as they have the liberty to use services on an 'as-needed' basis. Customers may also request customized services from the providers which is difficult in a product-oriented environment.

A high-level view of the service-oriented environment is shown in Figure 1. The service requester (customer) is the entity looking to accomplish some activity. The customer therefore gets in touch with a 'broker' which may be a registry of services or an agent that is capable of matching requirements with services. The broker then suggests prospective service providers to the customer in an effort to 'bind' a customer and a provider in a mutually beneficial fashion. A good detailed exposition of the service-oriented environment is available in [5].

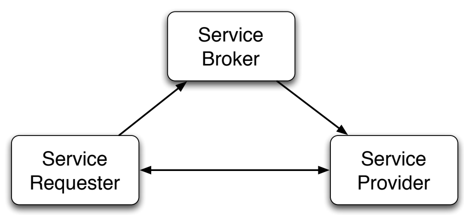

Figure 1. The basic model of an SOA

An issue that often crops up in this scenario is that the broker returns more than one service that is capable of catering to the functionality requirements of the respective customer. Selecting the best service from this set of functionally equivalent services is however non-trivial. Substantial research has been done in the past on making this selection. Most of the research is focused on utilizing various non-functional attributes of the candidate services for comparison and selecting the one that matches the requirements of the customer the most [6] [7] [8] [9].

In our research, we choose to look at the issue of service selection as part of the larger task of putting together individual services to form a composite application. Every functionality of this composite application may be realized by using one of a number of potential services. We assume that the discovery of these potential services on the basis of the functionality requirements has already been done. The part that we are interested in is selecting the 'one' service from these functionally equivalent services that aligns best with the requirements of the customer. Our representation of the domain of services from which the application is formed is shown in Figure 2. Each horizontally aligned set of services represents one level of functionality. The services in the set are functionally equivalent and capable of catering to a specific functionality. One service would need to be selected from each level to form the composite application. For example, in Figure 2, the composite application could be made up of services: $S1$-$S4$-$S5$-$S9$-$S10$. Another composition might be $S1$-$S2$-$S7$-$S8$-$S10$.

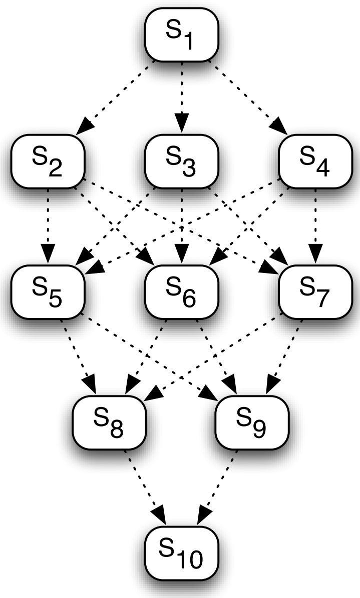

Figure 2. Service domain representation

In this paper, we present a technique to make the non-functional characteristics of the potential services, referred together as its 'Quality of Service (QoS)' attributes, as the factor responsible for service selection. The main issue that would be addressed here would be to compare functionally equivalent services on the basis of the collective score of all the QoS attributes. Although work has already been done in this field (refer to section 2), the novel aspect of this technique is that it would do the comparison among the candidate services on the basis of the preference of individual customers, rather than the actual QoS attribute values.

The remainder of this paper is structured as follows. Section 2 consists of a quick look at some work done in the past similar or related to our work. Section 3 is a discussion of various methods of comparing entities on the basis of their collective attribute ratings. Section 4 introduces the 'mid-level splitting' method, as applied to service selection, to capture the preferences of individual customers. Section 5 is a discussion on 'hypothetical equivalents and inequivalents method' to assign suitable weights to the individual QoS attributes of the service, again depending upon the preferences of individual customers. Finally, section 6 concludes the paper.

II. Related work

Service selection on the basis of QoS attributes has received appropriate attention in previous research. Shuping Ran's paper is a pioneering work in this field [6]. In the work, Ran suggests the use of an external 'QoS certifier' that certifies the QoS claims made by service providers on their respective services. The QoS values may then be incorporated into the 'UDDI' registry [10] to facilitate more appropriate service selection. This suggestion has an important restriction that the QoS claims made by service providers are very dynamic and to have a certifier continuously assess these claims can become prohibitively expensive. The work was nevertheless significant in the sense that it gave a new direction to approaching the issue of service selection.

Closer to our work, Godse *et al.* present a technique for service selection on the basis of the collective ratings of various QoS attributes [11]. The QoS attributes are grouped into clusters based on type similarity, and these clusters are the leaf nodes of a hierarchy. The hierarchy comprises the goal *i.e.* service selection as the root, and the factors leading up to the QoS attributes as the intermediate levels. The elements at each level are compared pair-wise to each other, and through a mathematical procedure, the pair-wise comparisons lead to a 'local' weight being assigned to each element. The element subsequently is assigned a 'global' weight by multiplying its local weight by that of its parent. The global weight thus calculated of each QoS attribute, along with the degree to which the attribute is present in the candidate service, contributes to the ranking of the services and their subsequent selection.

Liu *et al.* present an 'extensible QoS model' wherein the decision on which QoS attributes are to be considered for service selection is flexible [12]. The attributes considered would depend on the service domain in question or possibly on the basis of customer preference. After the attributes to be considered are decided, the values of these attributes in the candidate services are arranged in the form of a matrix. This matrix is then made to undergo successive stages of normalization which finally yields values reflecting the extent to which the relevant QoS attributes are present in each candidate service. Wang *et al.* have a similar approach to service selection with a few additions. [13]. First, their method takes into consideration linguistic expression of attributes such as 'slightly low', 'very low' *etc.* Second, their normalization procedure is different and the final result expresses the degree of the presence of attributes in the various candidate services as a value falling in the fixed range of 0 to 1.

Al-Masri *et al.* introduce the 'web service relevancy function (WsRF)' in their work [14]. WsRF(ws_i) is a measure of the relevance of a web service ws_i to the requirements of the concerned customer. WsRF is calculated by first calculating the 'distance' of each QoS attribute of the concerned service from the best respective value in the domain, and subsequently multiplying this distance with the weight (suggested by the customer) assigned to the attribute. These products are then summed over all the QoS attributes of the service to give the WsRF value.

We feel each of these techniques are relevant but have an important restriction. They are all based on the premise that the expectation of the customer with regard to each QoS attribute is directly proportional to the actual value of the respective attribute. For example, it is assumed that an improvement in the accuracy value of a service from 60% to 69% is equally significant for all customers as an improvement in accuracy from 90% to 99%. This is usually not true. In our work, we present a technique to capture the variation between the customer expectation of an attribute and the actual attribute values. Liu *et al.* do incorporate the customer expectations in their technique but the role of the customer in their method is restricted to the selection of attributes.

Further, in our work, we use a technique to assign weights to the QoS attributes that would be again based on the preference of the customer. Most of the related work discussed here assign these weights in an intuitive manner which is often inconsistent. Godse *et al.* do have a systematic method of weight assignment but their method does not incorporate the customer as directly as ours does.

III. Entity comparison on the basis of attributes

In this section we will discuss the basic strategies adopted for comparison between entities on the basis of their characteristic attributes. The intention is to gradually move from the most generic approach to our approach which is the main subject of this paper.

A simple method of comparison of entities on the basis of the respective attributes is pair-wise comparison [15]. In simple terms, pair-wise comparison involves picking out any two entities from the group and comparing each attribute of the two entities. The entity that 'wins' on a larger number of attributes is the winner. The winner is then similarly compared with a third entity, and so on. An example of this is shown in Figure 3. A, B, and C are three services with attributes: Accuracy, Response-time and Security. A, and B are first compared pair-wise, one attribute at a time. B is found to be better than A in terms of Response-time, and Security, whereas A is found to be better in terms of Accuracy. B thus 'beats' A, 2-1, and goes on to be compared with service C. Although this method is very simple, it has a few drawbacks. This comparison technique may give rise to cyclic results: we could have a situation where $B > A > C > B$. The degree of difference between the attributes is not taken into account in this method. Also, it is assumed that all the attributes have equal importance.

The drawback of getting cyclic results may be easily eliminated by using a simple ranking procedure. Here, as shown in Figure 4, the services are all compared *together* rather than pair-wise and the comparison is again on the basis of each attribute taken separately. The services are ranked relative to each other, and the service with the smallest total rank is selected. Although the problem of cyclic results are eliminated, the other two drawbacks: degree of difference among the attributes not being considered, and equal importance being given to attributes, remain.

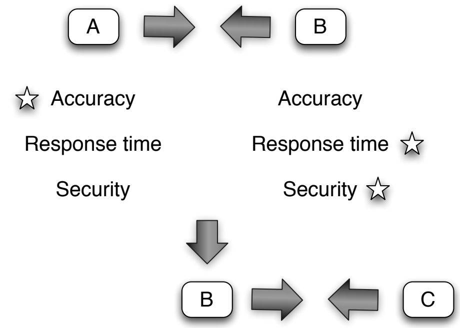

Figure 3. Pairwise comparison

	Accuracy	Response time	Security	Total
A	1	2	3	6
B	2	1	1	4
C	3	3	2	8

Figure 4. Comparison based on rankings

The method of 'normalized rating' of attributes helps overcome the drawback of the degree of difference between the attributes being overlooked [16]. In this technique, the actual values of the various attributes are considered. These values are then normalized by assigning a value of 1 to the service with the largest value of that attribute and a value of 0 to the service with the smallest attribute value. All the other services are assigned values between 0 and 1, in a way proportional to their respective attribute values. This is done separately for each attribute, as shown in Figure 5. The total of these normalized ratings for each service is then calculated, and the service with the largest total is considered best.

This technique however suffers from the important drawback that it assumes that the preference of the customer varies linearly with the variation of the attribute value. Repeating the same example as the previous section, it assumes that for a customer an improvement in Accuracy from 60% to 69% (say) is as significant as an improvement

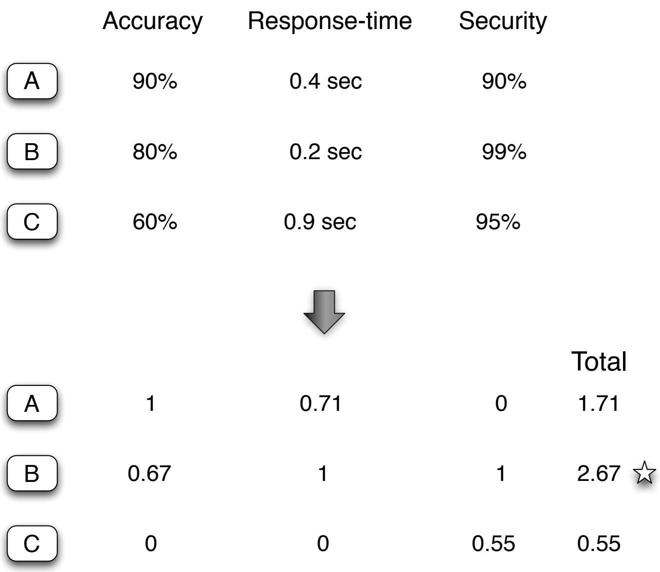

Figure 5. Normalized rating

from 90% to 99%. This is usually not true, and also varies from one customer to another. Presenting a technique that captures the preference of customers would be our first task in this paper, and will be dealt with in section 4. The other drawback of this method is that it still does not address the issue of equal importance being given to each attribute in the ranking. We would be presenting a technique to do this is section 5.

IV. Mid-level splitting method

To capture the preferences of customers in ranking services on the basis of their respective attributes, we adopt the 'mid-level splitting' technique. We borrowed this technique from the following paper on operations research [17].

The mid-level splitting technique involves, splitting the attribute value range at its mid-point and querying the customer on which of the two halves is more relevant. Depending on the response of the customer, the range of interest of the attribute values is shifted and this new range is further split at its mid-point. This process is continued recursively until the customer feels that each of the halves is equally relevant in which case the current point of splitting is considered to be the customer's preference equivalent of the mid-point of the original range. In a similar way, sufficient customer equivalents of points in the value range of the respective attribute are computed and an approximate curve between the customer preference and the actual attribute values is obtained.

It would be much easier if we used a simple example to explain this technique. Lets use the Accuracy attribute from the service example of the previous section. The best Accuracy value among the services available is 90%, and the worst Accuracy value is 60%. The customer has to be content with this range of Accuracy, thus, the upper limit of the customer preference, 1, is set at 90% and the lower limit is set at 60% (customer rating: $1 \Rightarrow 90\%$, and customer rating: $0 \Rightarrow 60\%$). To find the 0.5 value of the customer preference, we split the range at it mid-point, *i.e.* at 75%, and ask the customer the following question,

Which range in Accuracy is more significant for you, 60% to 75% or 75% to 90%? The increase in price of services if the first range is chosen is c_{11} to c_{12}, and for the second range it is c_{21} to c_{22}.

It should be noted at this point that we are in the process of assessing the dependence of price on individual service attributes, and in future literature dwell upon this topic. For now, we will simply use c_{ij}

Suppose the customer in response says: *75% to 90%*, then the range of interest becomes 75% to 90% and the range is split at its mid-point *i.e* 82.5%. The user is then asked the following question,

Which range in Accuracy is more significant for you, 75% to 82.5% or 82.5% to 90%? The increase in price ...

This process is continued recursively until the customer says both the ranges are equally significant. Suppose at this point the customer does say that an Accuracy range of 75% to 82.5% is equally significant as 82.5% to 90%. In this case, the 0.5 for this customer is set at 82.5% (thus customer rating: $0.5 \Rightarrow 82.5\%$). Now that the customer preference equivalent of 0, 0.5, and 1 are known, we may compute the equivalent of 0.25 by recursively splitting the Accuracy range 60% to 82.5% (corresponding to 0, and 0.5 respectively), and we may compute the equivalent of 0.75 by recursively splitting the Accuracy range 82.5% to 90% (corresponding to 0.5 and 1 respectively). This is done until enough points are obtained to plot the curve between the customer preference and the actual values of the attribute. This curve is then used to obtain the customer rating equivalent of the attribute values of each of the other services that fall in between the two limiting values.

Hypothetically, lets assume that the customer above (whose preference rating values corresponding to 0, 0.5, and 1 are 60%, 82.5%, and 90% respectively), answers further queries such that his/her preference ratings corresponding to 0.25, and 0.75 are found to be 76.875, and 86.25 respectively. The approximate preference curve for this customer corresponding to the Accuracy attribute is shown in Figure 6. From this curve, the customer rating of the Accuracy attribute of the other services may be found.

In a similar manner, the curves for the other attributes: Response-time, and Security may also be traced for the customer. A set of hypothetical curves for these two attributes are shown in Figure 7.

In a realistic situation however, especially when the services have a large number of attributes it would not be practicable to expect customers to respond to such a large number of queries. A more practical approach would

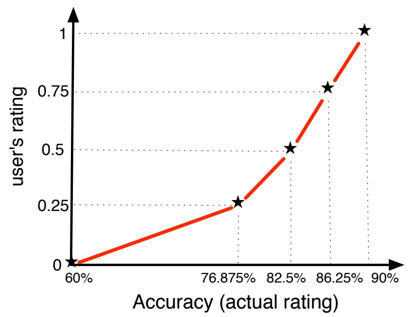

Figure 6. Accuracy (user's vs. actual rating)

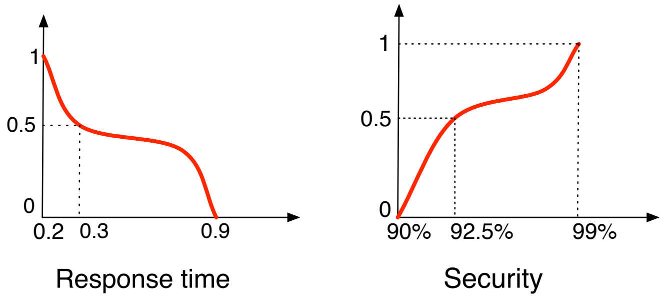

Figure 7. user's vs. actual rating

thus be to present the customer with a number of sample preference curves which could perhaps have been collected from customers in the past. The customer could be given the option to choose one of these curves. Only if he/she is not satisfied, a new specific curve could be formed for that customer following the method outlined above.

V. Weight Assignment to Attributes

The drawback of equal importance being given to all the attributes may be overcome by assigning appropriate weights to the attributes. An attribute that has a higher weight is one that is of greater importance to the customer. The issue then is: how do we determine these weights?

A simple approach to weight assignment would be the 'intuitive' approach. Here, the customer is asked a set of questions regarding his/her expectations of the service, and depending on the responses, an experienced personnel takes an informed decision on the weight assignment. We however feel this technique lacks consistency and is substantially dependent on human judgement.

We therefore chose to borrow the *Hypothetical Equivalents and Inequivalents Method (HEIM)* for assignment of weights to the attributes [18]. In this method, a number of hypothetical services are created by using random combinations of attribute values. Various techniques may be used for these hypothetical attribute combinations. For demonstration, we will use a very simple technique wherein we will be using the attribute values corresponding to a user rating of 0, 0.5, and 1. The user rating values for the three attributes are randomly combined to form a set of ten hypothetical services shown in Figure 8. The actual attribute values corresponding to 0, 0.5, and 1 are obtained from the curves in Figure 6 (for Accuracy) and Figure 7 (Response-time, and Security) as shown below:

	Accuracy	Response time	Security
HS1	1	0.5	1
HS2	0.5	0	0.5
HS3	0	1	0
HS4	1	0.5	0.5
HS5	0.5	0	0
HS6	0	1	1
HS7	1	0.5	0
HS8	0.5	0	1
HS9	0	1	0.5
HS10	1	0	0

Figure 8. Hypothetical services

$Accuracy$ (Figure 6)
$0 \Rightarrow 60\% : \;\; 0.5 \Rightarrow 82.5\% : \;\; 1 \Rightarrow 90\%$
$Response\text{-}time$ (Figure 7)
$0 \Rightarrow 0.9\ sec : \;\; 0.5 \Rightarrow 0.3\ sec : \;\; 1 \Rightarrow 0.2\ sec$
$Security$ (Figure 7)
$0 \Rightarrow 90\% : \;\; 0.5 \Rightarrow 92.5\% : \;\; 1 \Rightarrow 99\%$

The hypothetical services with the actual values of the attributes are shown in Figure 9.

Each attribute is then assigned an unknown weight w_i whose value needs to be calculated. These weights are such that,

$$\sum w_i = 1 \tag{1}$$

where i are the different attributes.

Using these unknown weights, an expression for the total weight of each hypothetical service is calculated in the manner shown in Figure 10 . The total weight for each service is the summation over all attributes, the product of the customer rating of each attribute and the unknown weight w_i assigned to that attribute, as shown below,

	Accuracy	Resp. time	Security
HS1	90	0.3	99
HS2	82.5	0.9	92.5
HS3	60	0.2	90
HS4	90	0.3	92.5
HS5	82.5	0.9	90
HS6	60	0.2	99
HS7	90	0.3	90
HS8	82.5	0.9	99
HS9	60	0.2	92.5
HS10	90	0.9	90

Figure 9. Hypothetical services (actual ratings)

$$W_{service} = \sum w_i \cdot \mathit{customer\ rating}_i \qquad (2)$$

	w_1	w_2	w_3	
	Accuracy	Response time	Security	Total
HS1	1	0.5	1	$w_1+0.5*w_2+w_3$
HS2	0.5	0	0.5	$0.5*w_1+0.5*w_3$
HS3	0	1	0	w_2
HS4	1	0.5	0.5	$w_1+0.5*w_2+0.5*w_3$
HS5	0.5	0	0	$0.5*w_1$
HS6	0	1	1	w_2+w_3
HS7	1	0.5	0	$w_1+0.5*w_2$
HS8	0.5	0	1	$0.5*w_1+w_3$
HS9	0	1	0.5	$w_2+0.5*w_3$
HS10	1	0	0	w_1

Figure 10. Hypothetical services (with weights)

The hypothetical service table in Figure 9 is then put before the customer, and the latter is asked to compare as many services he/she can. The comparison may be such that the customer rates two or more hypothetical services as being equally good for his/her purpose, or he/she may rate one service as being better of worse than another. This comparative rating of the customer is captured in the form of mathematical equalities and inequalities between the service total weight expression. Suppose the customer says that hypothetical service $HS2$ is more suited to his/her requirements than $HS3$, $HS6$ is better than $HS5$, and $HS10$ is worse than $HS3$. These responses may be captured in the following expressions (the weights for services are obtained from the last column of the table in Figure 10):

$$\begin{aligned} &0.5 * w_1 + 0.5 * w_3 > w_2 \\ &w_2 + w_3 > 0.5 * w_1 \\ &w_1 < w_2 \end{aligned} \qquad (3)$$

Having obtained these expressions, we have abstract information on the importance that the customer places on the various attributes. To explicitly express this information in the form of weights, we attempt to optimize the following expression,

$$Minimize(1 - \sum w_i)^2 \qquad (4)$$

This is based on equation (1) that the sum of the weights assigned to the attributes should be 1. The equalities and inequalities in (3) are the constraints in this optimization. Various techniques may used to perform this optimization and obtain the values of w_i [19] [20].

We utilized the ‘Solver’ toolkit [21] that comes built-in with Microsoft Excel (2004) for solving the optimization problem given the constraints of equation (3). We obtained the following values for the weights corresponding to the three attributes: $w_1 = 0.67$, $w_2 = 0.33$, and $w_3 = 0$.

These weight values are then used along with the user rating curves to calculate the total weight of each of the available services. To do this, first the user rating equivalent of each of the attribute values of the three services of our example: A, B, and C, are calculated from the rating curves. The attribute values in Figure 5 are traced on the user rating curves as shown in Figure 11.

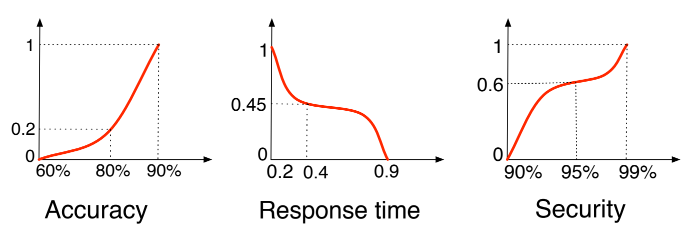

Figure 11. User equivalent of service attribute values

The user rating equivalent of each of the attributes of services A, B, and C are shown in Figure 12. The total weight of each of the services is then calculated using equation (2). For example, for service B:

$$\begin{aligned} Total\text{-}weight &= 0.2 * 0.67 + 1 * 0.33 + 1 * 0 \\ &= 0.46 \end{aligned}$$

The total service weights for all available services are shown in Figure 12. The service with the highest weight, A is therefore found to be the most optimal for the customer in terms of QoS and is selected.

	w_1 = 0.67	w_2 = 0.33	w_3 = 0	
	Accuracy	Response time	Security	Total
A	1	0.45	0	0.82 ☆
B	0.2	1	1	0.46
C	0	0	0.6	0

Figure 12. Total service weight

VI. Conclusion

In this paper, we present a method to compare functionally equivalent services on the basis of their QoS attributes. In doing this, we take into account the fact that the improvement in each attribute value does not necessarily have a linear relationship with the increase in the customer preference of the respective attribute. The main contribution of this work is therefore to capture the relationship between the customer rating of each attribute and the actual attribute value. The service comparison is done on this basis and is specific to individual customers.

We further address the issue of unequal importance being given to different QoS attributes by different customers. We use the *Hypothetical Equivalents and Inequivalents Method* to assign weights to individual attributes based again on the respective customer preference. This is an attempt to replace the rather inconsistent method of depending on expert judgement to make these weight assignments.

We however concede that the proposed technique may sometimes become tedious for the customer. It is therefore suggested that the customer first be shown a set of canonical curves to choose from. Only when the customer is not satisfied and wants customized curves for any or all the attributes should the proposed technique be used and the customer be made to go through the set of question-answers.

References

[1] Patrizia Battilani and Francesca Fauri, *The rise of a service-based economy and its transformation: the case of Rimini*, Rimini Centre for Economic Analysis, Working Paper Series, 2007.

[2] Faiz Gallouj, *Innovation in the service economy: the new wealth of nations*, Edward Elgar Publishing, 2002.

[3] http://aws.amazon.com/ec2/

[4] http://www.rightscale.com

[5] Mike P. Papazoglou, *Service-Oriented Computing: Concepts, Characteristics and Directions*, Proceedings of the Fourth International Conference on Web Information Systems Engineering, pp. 3-12, 2003.

[6] Shuping Ran, *A model for web services discovery with QoS*, ACM SIGecom Exchanges, pp. 1-10, 2003.

[7] Zhengdong Gao and Gengfeng Wu, *Combining Qos-based service selection with performance prediction*, IEEE International Conference on e-Business Engineering (ICEBE), pp. 611-614, 2005.

[8] Natallia Kokash, *Web service dicovery with implicit QoS filtering*, Proceedings of the IBM PhD Student Symposium, in conjunction with the International Conference on Service Oriented Computing (ICSOC), pp. 61-66, 2005.

[9] V. Deora, J. Shao, G. Shercliff, P.J. Stockreisser, W.A. Gray, and N.J. Fiddian, *Incorporating QoS specifications in service discovery*, IWeb Information Systems (WISE), pp. 252-263, 2004.

[10] Francisco Curbera, Matthew Duftler, Rania Khalaf, William Nagy, Nirmal Mukhi, and Sanjiva Weerawarana, *Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI*, IEEE Internet Computing, pp. 86-93, 2002.

[11] Manish Godse, Rajendra Sonar, and Shrikant Mulik, *The Analytical Hierarchy Process Approach for Prioritizing Features in the Selection of Web Service*, Sixth European Conference on Web Services, pp. 41-50, 2008.

[12] Yutu Liu, Anne H.H. Ngu, and Liangzhao Zeng, *QoS Computation and Policing in Dynamic Web Service Selection*, International World Wide Web conference, 2004.

[13] Xia Wang, Tomas Vitvar, Mick Kerrigan, and Ioan Toma, *A QoS-aware Selection Model for Semantic Web Services*, International conference on service oriented computing, pp. 390-401, 2006.

[14] Eyhab Al-Masri and Qusay H. Mahmoud, *Discovering the Best Web Service*, International World Wide Web conference/Poster paper, 2007.

[15] Filippo A. Salustri, *Pairwise Comparison*,http://deed.ryerson.ca/ fil/t/pwisecomp.html

[16] Miriam Fernndez, David Vallet, and Pablo Castells, *Probabilistic Score Normalization for Rank Aggregation*, 28th European Conference on Information Retrieval, pp. 553-556, 2006.

[17] Alison R. Callaghan and Kemper E. Lewis, *A 2-Phase aspiration-level and utility theory approach to large scale design*, Proceedings of Design Engineering Technical Conferences And Computers and Information in Engineering Conference, 2000.

[18] Michael Kulok and Kemper Lewis, *Preference consistency in multiattribute decision making*, Proceedings of Design Engineering Technical Conferences And Computers and Information in Engineering Conference, 2005.

[19] D. Luenberger, *Linear and Nonlinear Programming*, Addison-Wesley, Reading (MA), 1984.

[20] C. L. Hwang, J. L. Williams, and L. T. Fan, *Introduction to the generalized reduced gradient method*, Institute for Systems Design and Optimization, Kansas State University (Manhattan), 1972.

[21] http://www.solver.com/