Design of Competitive Web Services Using QFD for Satisfaction of QoS Requirements

Gang WANG\textsuperscript{(a)}, Li ZHANG\textsuperscript{(b)}, Nonmembers, Yonggang HUANG\textsuperscript{(c)}, Student Member, and Yan SUN\textsuperscript{(d)}, Nonmember

SUMMARY It is the key concern for service providers that how a web service stands out among functionally similar services. QoS is a distinct and decisive factor in service selection among functionally similar services. Therefore, how to design services to meet customers’ QoS requirements is an urgent problem for service providers. This paper proposes an approach using QFD (Quality Function Deployment) which is a quality methodology to transfer services’ QoS requirements into services’ design attribute characteristics. Fuzzy set is utilized to deal with subjective and vague assessments such as importance of QoS properties. TCI (Technical Competitive Index) is defined to compare the technical competitive capacity of a web service with those of other functionally similar services in the aspect of QoS. Optimization solutions of target values of service design attributes is determined by GA (Genetic Algorithm) in order to make the technical performance of the improved service higher than those of any other rival service products with the lowest improvement efforts. Finally, we evaluate candidate improvement solutions on cost-effectiveness. As the output of QFD process, the optimization targets and order of priority of service design attributes can be used as an important basis for developing and improving service products.

key words: QFD, GA, fuzzy set theory, web service, QoS

1. Introduction

SOC (Service Oriented Computing), as a new computing paradigm, is changing the way of developing the information system. It can loosely couple web services provided by different providers so that enterprises response quickly to changes in business processes. Web services as a basic technology for SOC have become more frequently used.

QoS is an important aspect of web services. It is crucial to the success of a web service, especially in the situation of more than one web services with similar functions on the market. How to design a web service is a challenge for service providers to meet customers’ QoS requirements and make the web service more competitive than other services with similar functions in terms of QoS.

QFD (Quality Function Deployment) is one of the major quality methodologies, which transfers customers’ requirements into product characteristics during the design process. QFD, proposed in Japan [1], has now become an important quality design technology worldwide. It is an effective technology to meet customer requirements and win the market competition.

So we use QFD to design web services and trace customers’ QoS requirements into design attributes of web services. It can make the designed web services reflect QoS requirements by using QFD. The targets and order of priority of design attributes of web services are the output of QFD. Because the QFD process involves subjective and vague assessment, caused by the cognitive limit, fuzzy logic has been used for expressing degrees in linguistic terms. Fuzzy assessment with these terms is frequently intuitive and effective for decision-makers. During the process of determining the targets of design attributes of web services, we use GA (Genetic Algorithm), which can solve global optimization problems rapidly, to find the optimization target solutions. The solutions can make the designed web services more competitive than other services with similar functions in terms of QoS within the lowest efforts for the web service providers, such as on development cost.

The remainder of this paper is organized as follows. Section 2 discusses some related works about QoS of web services and QFD. Section 3 describes our approach to design web services using Fuzzy QFD to meet QoS requirements, including the determination of importance and technical targets of web service design attributes. Section 4 presents a web service system example to illustrate our approach and proposes a ROI decision analysis diagram to analyze the output solutions on cost-effectiveness. Finally, Sect. 5 concludes the paper.

2. Related Work

QoS is an important aspect of web service systems. With the increasing number of web services available on the Internet, web services with the same or similar function emerges inevitably. QoS becomes a decisive factor to distinguish the functionally same or similar web services. It affects the customers’ selection and the success of service providers. To facilitate customers’ service selection from functionally same or similar web services, QoS-based/aware web service selection approaches have been proposed. The web service selection approaches/ algorithms rank the functionally same or similar services based on QoS requirements of the customers and returns the service with the highest rank to the customers as stated by [2]. The QoS-based service selection approaches can be generally classified into two categories according to the number of tasks. One is the service selection for a single task whereas the other is the service selection for each constituent task of the composite process.
and search of an optimization combination of web services. Currently there are various QoS-based web service selection approaches. The comprehensive reviews of QoS-based web service selection techniques can be found in the surveys [2]–[4].

However, as the paper [5] pointed out, current research on QoS of web services focuses on service users and the implementation stage, which means focusing on how to find the optimal services satisfying QoS requirements among existing web services on the network from the customer’s perspective. It is also important how to design a web service to guarantee QoS requirements before making it available to the customers, especially for service providers. The paper [6] proposed a methodology WSMoD (Web Services Modeling Design), extending MDA (Model Driven Architecture) to address the QoS-based web service design explicitly. It adds a specific methodological step and extends PIM (Platform Independent Model) to define QoS/non-functional requirements. The research [5] incorporates QoS into the web service design at the development stage. It introduces a QoS meta-model by extending UML QoS profile to support QoS specifications, and extends WSDL (Web Service Description Language) to include the proposed QoS meta-model. These approaches could assist web service providers to better understand what QoS of web services represents including QoS properties, their descriptions and dimensions, and they could ease web service providers to specify QoS. But how to design web services according to QoS metrics and how to determine service design specifications to meet QoS requirements are not addressed in these papers.

There are few web service design methods about how to determine target values of web service design attributes so that designed services have the ability to meet QoS requirements. Service design attributes are technical characteristics of services such as programming languages, data access protocols and database capacity which affect QoS. Only some design attributes of web services about network and communication e.g. bandwidth are analyzed in several papers [7]–[9]. They are not systematic to transfer QoS requirements into design attributes explicitly and quantitatively.

QFD, a quality methodology to transfer user demands into design quality, has been widely used in industries such as construction and aviation [1], [10], and also used in software engineering [11], [12], [24]. But researches on using QFD for developing emerging SOC systems are few [13]. Fuzzy theory has been applied to QFD to deal with a variety of qualitative and fuzzy input [14], [15]. GA has been applied to optimization of technical characteristics in QFD [16].

So we propose an approach to develop SOC-based web service systems using QFD. Based on technical information on competitive products, the approach searches optimal solutions of target values of web service design attributes in order to improved web services more competitive than any other service products within the lowest improvement efforts such as on cost. The solutions provide web service providers basis and support for development and improvement of their web service products.

3. Design of Web Service Systems Using Fuzzy-QFD

We use HoQ (House of Quality) [17], the core graphic tool of QFD, to represent the relationship between QoS requirements and web service design attributes. Figure 1 shows a HoQ of web service systems. It is a large matrix, which generally has five different components:

1. WHATS matrix and importance of customers’ requirements: customers’ QoS requirements of web services named as VOC (Voice of the Customer). Collection of VOC is a relatively independent process. QoS properties of web services have been identified in [18], [19].
2. HOWS matrix and direction of improvement: product features to meet customers’ requirements. Here is web service design attributes affecting QoS. Direction of improvement means if “more is better” or “less is better” for each design attributes.
3. Interrelationship matrix: the impact of web service design attributes on customers’ QoS requirements. It reflects the mapping from QoS requirements to the design attributes.
4. Technical competitive evaluation matrix: to evaluate the technical performance of similar web services on the market in order to determine technical targets of our web service products.
5. Importance and targets of product features: the order of priority and target of web service design attributes to improve. They are the important outputs of HoQ. They are determined based on the importance of each property of QoS requirements, the relationship in the interrelationship matrix and technical competitive evaluation.

Fig. 1 HoQ used for web service design.
The steps using FQFD (Fuzzy QFD) to design web service to satisfy the customers’ QoS requirements are described as following:

1. Identify QoS requirements and determine the importance of each property of QoS requirements.
2. Identify web service design attributes directly related with QoS requirements.
3. Establish the impacts of design attributes of web services on QoS requirements in the interrelationship matrix and determine the importance of design attributes.
4. Evaluate the technical performance of competitive web services.
5. Determine the target values of web service design attributes to be improved based on the importance of each property of QoS requirements, the relationship in the interrelationship matrix and technical and cost constraints.

After the establishment of HoQ through qualitative and quantitative analysis, the HoQ translates “what to do” into “how to do”, which means technical targets of web services are determined to meet QoS requirements.

The rest of this section includes 2 subsections: (1) Determining Importance of Design Attributes based on Fuzzy Set Theory and (2) Determination of Target Solutions of Design Attributes with the Fewest Improvement Efforts Using GA. The first subsection describes how to transfer customers’ prioritized QoS requirements into the importance of web service design attributes using fuzzy set theory. The second subsection describes how to find an optimization target solutions of web service design attributes for the improvement to meet QoS requirements and more competitive than other similar web service products on the market with the fewest efforts on the aspect of cost, time and so on. It uses GA to search the target solutions based on the information of the importance of design attributes, technical competitive evaluation and some other constraints.

3.1 Determining Importance of Design Attributes Based on Fuzzy Set Theory

A variety of input needs the subjective evaluation in many cases in the process of QFD caused by the incomplete information by nature. For example, interrelationship matrix of QoS requirements and design attributes of web service are usually completed by human’s judgments with concepts, such as strong, medium, weak relation. These language concepts have vagueness, it is not easy to express as exact numbers. Fuzzy set theory shows great vitality in dealing with vagueness of the language with the membership function to change vague, subjective and qualitative information into quantitative description, so we apply fuzzy set theory to deal with qualitative and vague language in QFD.

First, we should determine the amount of linguistic variables and the corresponding fuzzy numbers. The amount reflects the degree of uncertainty and should be not too big, which means the judgments should be not divided into many degrees because that confuses users’ selection, the number of linguistic variables is generally not more than 11 [23]. Linguistic variables are quantified using TFNs (Triangular Fuzzy Numbers) in this paper because they are widely used and easy to use.

The customers and QFD team use the defined linguistic variables to establish the level of importance of each service QoS property and the interrelationship between QoS properties and design attributes.

With more than one design attributes of web services, we need to determine the importance or order of priority of web service design attributes to be improved.

The importance of web service design attributes is assessed according to the importance of QoS properties of web services and the strength of relationship between QoS properties and web service design attributes in what-how matrix.

It is calculated by the following formula:

\[
\omega_{hj} = \sum_{i=1}^{M} \omega_i \otimes r_{ij}
\]  

where \(\omega_{hj}\) is the weight of the \(j\)th design attribute, \(\omega_i\) is the weight of the \(i\)th QoS property, \(r_{ij}\) is the strength of the relationship between them, \(\omega_{hj}\) is a fuzzy number, \(M\) is the account of QoS properties.

Providing \(\omega_i = (a_i, b_i, c_i)\), \(r_{ij} = (a_{ij}, b_{ij}, c_{ij})\), \(\omega_{hj} = (ah_j, bh_j, ch_j)\),

According to the formulae of addition and multiplication of two fuzzy numbers in [20]:

Then

\[
ah_j = \sum_{i=1}^{M} a_i \times a_{ij}, bh_j = \sum_{i=1}^{M} b_i \times b_{ij}, ch_j = \sum_{i=1}^{M} c_i \times c_{ij}
\]  

In this paper, we use the centroid method, one of popular methods to rank fuzzy numbers, to get the order of priority of web service design attributes to be improved. It is calculated as following:

\[
E(\tilde{A}) = \int [x \times \mu_{\tilde{A}}(x)]dx / \int [\mu_{\tilde{A}}(x)]dx
\]

where \(E(\tilde{A})\) is the exact value equivalent to fuzzy number \(\tilde{A}\). For a TFN \(\tilde{A} = (a, b, c)\), it is

\[
E(\tilde{A}) = (a + b + c)/3
\]

3.2 Determination of Target Solutions of Design Attributes with the Fewest Improvement Efforts

After determining the order of priority of web service design attributes, the targets of web service design attributes need to been determined, which means how to alter values of web service design attributes to improve QoS of web services as efficiently as possibly within limited development resources.
But how are web service design attributes altered quantitatively? It is impossible to increase and decrease target values of web service design attributes unlimitedly under limited development resources such as development cost. Actually, web service providers just want to provide more competitive products than other similar services. A web service is more competitive if the value of every web service design attribute is better than that of any other similar web services, but it is impractical because of development resource constraints. So determination of improvement target of design attributes is a complex multi-variable, multi-objective decision-making process. It needs balancing various contradictions and conflicts such as between customer demands and product development cost and time.

We propose a method to determine the improvement targets so that the improved web service is more competitive than any others with similar functions on the market in terms of QoS requirements with less efforts such as on costs or time.

3.2.1 Introduce of TCI to Measure Technical Competitive Capacity of Web Services

Definition of TCI. To measure technical performance or competitive capacity of web services and make them compared, we define

Technical Competitive Index of web services

\[
TCI(W_i) = \sum_{j=1}^{N} \omega h_j \times nx_j
\]

where \(W_i\) is the web service provided by the \(i^{th}\) competitive providers, \(\omega h_j\) is the weight of the \(j^{th}\) web service design attribute, \(nx_j\) is the normalized value of the \(j^{th}\) web service design attribute of \(W_i\). The higher \(TCI(W_i)\) is, the more competitive \(W_i\) is.

Different web service design attributes have different dimensions and data types of numeric or ratio. Ratio data changes from 0 to 100%. Numeric data may change in different domains. For example, the two web service design attributes of response time and bandwidth are numeric. Bandwidth is positive correlation to QoS of web services, which means the higher the bandwidth is, the better QoS is. On the other hand, response time is negative correlation with optimization objective, which means the lower response time is, the better QoS is. To remove these effects, web service design attributes should be normalized to make different properties into the same extent of 0 to 1.

For the positive correlative web service design attributes, this paper uses the following normalization function

\[
np_{ij} = (p_i - p_{ij}^{\text{min}})/(p_{ij}^{\text{max}} - p_{ij}^{\text{min}})
\]

For the negative correlative web service design attributes, this paper uses the following normalization function

\[
np_{ij} = (p_{ij}^{\text{max}} - p_i)/(p_{ij}^{\text{max}} - p_{ij}^{\text{min}})
\]

where \(p\) represents a web service design attributes, \(p_{ij}\) is the value of the \(i^{th}\) web service design attribute of the \(j^{th}\) competitive web service, \(p_{ij}^{\text{max}}\) is the maximum of the attribute \(p\) in the competitive web services and \(p_{ij}^{\text{min}}\) is the minimum.

Validation of TCI to Measure Web Service Competitive Capacity. Customers’ satisfaction determines the success or failure of products. CSI (Customer Satisfied Index) is a usual indicator of how successful a product is provided to the marketplace. CSI\((W_i)\) is calculated by combining values of QoS attributes with the corresponding weights. The higher CSI\((W_i)\) is, the more satisfied customers with \(W_i\), which means \(W_i\) is used by more customers and \(W_i\) is more competitive on the market. TCI indicates the technical capacity of a product from the technical prospective. Equation (8) shows the relationship between TCI and CSI, and shows how the technical capability of a product links customers’ needs to satisfy their requirements. This is the reason why we use QFD to design web services. It can trace customers’ QoS requirements into design attributes of web services. From Eq. (8), we know CSI\((W_i)\) is higher and \(W_i\) is more competitive when TCI\((W_i)\) is higher.

\[
TCI(W_i) = \sum_{j=1}^{N} [\omega h_j \times nx_j]
\]

\[
= \sum_{j=1}^{N} \left[ \omega_i \times r_{ij} \right] \times nx_j
\]

\[
= \sum_{i=1}^{M} [\omega_i \times \sum_{j=1}^{N} (r_{ij} \times nx_j)]
\]

\[
= \sum_{j=1}^{M} [\omega_i \times n_{ij}] = CSI(W_i)
\]

where \(\omega_i\) and \(r_{ij}\) are the same meanings in Eq. (1), \(n_{ij}\) is the value of the \(i^{th}\) QoS property, contributed by web service design attributes.

Both CSI and TCI can indicate QoS of web services. But CSI assesses degree of web services to meet QoS requirement from the customer’s view while TCI assess market competitiveness of web services from a technical view.

After the definition of TCI as the competitive measurement index, the determination process is to find targets of web service design attributes in order that the improved web service has higher TCI than any other competitive web services in QoS with the lowest efforts such as development cost and time.

Mathematical model of the problem is:

\[
R(x_{0,1}, x_{0,2}, \ldots, x_{0,N}) = [R_1(x_{0,1}, x_{0,2}, \ldots, x_{0,N}),
\]

\[
R_2(x_{0,1}, x_{0,2}, \ldots, x_{0,N}), \ldots, R_L(x_{0,1}, x_{0,2}, \ldots, x_{0,N})]\]

(9)

\[
\text{min} R(x_{0,1}, x_{0,2}, \ldots, x_{0,N})
\]

(10)

s.t. (subject to)

\[
TCI(W_0) \geq G, 0 < i \leq K, \text{max}[TCI(W_i)] < G \leq 100
\]

(11)

\[
\text{min}_{0 \leq i \leq K} x_{i,j} \leq x_{0,j} \leq \text{max}_{0 \leq i \leq K} x_{i,j}
\]

(12)
where there are K competitive web services, N web service design attributes and a vector of L optimization objectives, i = 0 means the web service to be improved. The vector \(R(x_{0,1}, x_{0,2}, \ldots, x_{0,N})\) is the efforts of the implementation of improvement. G is the TCI goal, determined by \(W_0\) provider. G must be higher than \(\max[TCI(W)]\) to make sure TCI of improved \(W_0\) higher than other competitors. The upper limit of G is 100 when each positive correlative web service design attribute \(x_{0,j}\) takes the maximum \(\max_{0<i<K}x_{i,j}\) and each negative correlative web service design attribute \(x_{0,j}\) takes the minimum \(\min_{0<i<K}x_{i,j}\). Constraint (11) means the improved web service \(W_0\) must be better and more competitive than similar web services in terms of TCI. Constraint (12) is upper and lower limits of \(x_{0,j}\). Although values of web service design attributes are valid in positive numbers, they make search space large and reduce the search efficiency. So we choose \(\min_{0<i<K}x_{i,j}\), the minimum value of \(x_{i,j}\) as the lower and choose \(\max_{0<i<K}x_{i,j}\) as the upper. In the search space limited by Constraint (12), at least one solution higher than \(\max[TCI(W)]\) must exist in the case that the value of \(x_{0,j}\) is \(\max_{0<i<K}x_{i,j}\) if \(x_{0,j}\) is positive correlation to TCI or \(x_{0,j}\) is \(\min_{0<i<K}x_{i,j}\) if \(x_{0,j}\) is negative correlation with TCI.

To let TCI of improved \(W_0\) higher than those of any other competitors, determination of improvement targets of web service design attributes with the lowest efforts is constraint satisfaction problem. There are several algorithms for these problems. Many of traditional search algorithms are single point search algorithms, namely, by defining some rules of search, the solution of the problem changes from the current solution (point) to another better solution (point) according to the rules. For more than one peaks in the search space, these traditional point to point search algorithms are often caught up in the local optimal solution of a single peak while GA begins with many points named as individuals and assesses multiple solutions in the search space simultaneously, making GA have good performance in global search and reducing the risk of trapping in local optimal solutions [21]. GA can solve global optimization problems rapidly and widely used in constraint satisfaction problems. So this paper adopts genetic algorithm (GA) to search for optimization target solutions.

3.2.2 Search for Optimization Target Solutions of Design Attributes Using GA

The optimization objective is to find optimal solutions of improvement targets of web service design attributes which have the best trade-off, known as a Pareto set. It indicates best balance between improvement targets and development resources such as cost and time.

The implementation of genetic algorithms specific for optimal solutions of improvement targets includes the following steps: (1) genome coding, (2) population initialization, (3) fitness evaluation, (4) genetic manipulation including crossover and mutation operators, and (5) selection operator.

**Genome Coding and Population Initialization.** To use the GA to search for optimization solutions of the targets of design attributes, we first need to design a genome in GA to represent the problem.

Each gene \(g_i\) represents a web service design attribute \(t_i\). The \(g_i\) takes the integral value if the design attribute \(t_i\) is discrete while the \(g_i\) takes the double value if the design attribute \(t_i\) is continuous. The range of each web service design attribute is limited by Constraint (12). Initial population is generated, in which each gene in a genome is set randomly within its range.

**Fitness Evaluation.** To measure the degree of an individual near optimal solutions, we need to define a fitness function. Because the optimization objective is to find optimization target solutions of design attributes which have higher technical performance than any other competitive web services within the fewest development efforts, we make \(R(x_{0,1}, x_{0,2}, \ldots, x_{0,N})\) in Eq. (10) as fitness function to evaluates the development resources the solutions use, and we take \(TCI(W) \geq \max[TCI(W)]\) in Eq. (11) as the constraint. Because every element such as cost and time in the vector \(R(x_{0,1}, x_{0,2}, \ldots, x_{0,N})\) plays a negative role for web service providers, the fitness function \(R(x_{0,1}, x_{0,2}, \ldots, x_{0,N})\) should be minimized. If the length of the vector \(R(x_{0,1}, x_{0,2}, \ldots, x_{0,N})\) is greater than 1, the problem become multi-objective optimization, and we adopt the algorithm used in GA described in [22].

The selection function is the most used roulette selection, and the crossover function is the standard two-points crossover [21] while the mutation function randomly selects a web service design attribute \(t_i\) (i.e., a gene in the genome) and then replaces the old value of \(t_i\) with a random number selected uniformly from the range for design attribute \(t_i\).

The optimization solutions of web service design attribute targets produced by the GA will have the fewest efforts for improvement.

4. Case Study and Result Analysis

We will follow the steps of establishment and decision making of HoQ by using an example of house of quality of the web service system [13], to illustrate the application of Fuzzy-QFD to design of SOC based web service systems, including determination of importance and target values of web service design attributes. We analyze the validity of target solutions which means whether the solutions have the ability of the fewest efforts to improve web services more competitive than any others. We analyze candidate target solutions and provide a method to determine the target solutions to get the maximum ROI (Return on Investment).

4.1 Step 1 and 2: Identification of QoS Proprieties and Design Attributes of a Web Service System

There are various QoS models of web services and no consensus on the QoS model. Here we use as example six quality proprieties and ten typical web service design at-
Table 1 QoS properties of web services.

<table>
<thead>
<tr>
<th>QoS Requirements</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1: Interoperability</td>
<td>It refers to the ability of a collection of communicating entities to share specific information and operate on it according to an agreed-upon operational semantics.</td>
</tr>
<tr>
<td>R2: Usability</td>
<td>Measuring the experience quality of users in employing services to accomplish the goals in terms of task time, ease-of-use and so on.</td>
</tr>
<tr>
<td>R3: Adaptability</td>
<td>The ease with which a service system may be modified to adapt to changed requirements such as on users’ business processes.</td>
</tr>
<tr>
<td>R4: Performance</td>
<td>It is generally related to response time, throughput, or timeliness (ability to meet deadlines, i.e., to process a request in a deterministic and acceptable amount of time).</td>
</tr>
<tr>
<td>R5: Reliability</td>
<td>The ability of a service system to perform its required functions under stated conditions for a specified time interval.</td>
</tr>
<tr>
<td>R6: Scalability</td>
<td>The ability of changing the computing capacity of web service providers’ system to process variable users’ requests.</td>
</tr>
</tbody>
</table>

Table 2 Design attributes of web services.

<table>
<thead>
<tr>
<th>Web Service Design Attributes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1: Languages</td>
<td>Number of programming languages supporting for developing web services</td>
</tr>
<tr>
<td>T2: Data Types</td>
<td>Data types supporting for web service development such as plain data, media data</td>
</tr>
<tr>
<td>T3: Data Access Protocols</td>
<td>Protocols provided for services such as SOAP, REST</td>
</tr>
<tr>
<td>T4: Services</td>
<td>Number of services provided by the service platform</td>
</tr>
<tr>
<td>T5: Database Capacity (Billion Objects)</td>
<td>Storage Capacity of database used for web services</td>
</tr>
<tr>
<td>T6: Downtime Per Week (Mins)</td>
<td>Probability of web service platform in downtime</td>
</tr>
<tr>
<td>T7: Bandwidth (MB/s)</td>
<td>Average rate of data transfer of web service platform</td>
</tr>
<tr>
<td>T8: Maximum Users</td>
<td>Maximum users supported by the service platform</td>
</tr>
<tr>
<td>T9: Query Response Time (s)</td>
<td>Time from sending a request to receiving a response</td>
</tr>
<tr>
<td>T10: Download Speed (MB/s)</td>
<td>Average rate of data download</td>
</tr>
</tbody>
</table>

The impact relationships are rated mainly by the QFD team. The members come from different departments of a company providing web services. The way to obtain the relationships is introduced in [24]. Because the subjective evaluation is involved which has ambiguity, fuzzy set theory is used to represent fuzzy options. We mentioned the amount of linguistic variables and the corresponding fuzzy numbers are determined first in Sect. 3.1.

4.2 Step 3: Establishment of the Impacts of Web Service Design Attributes on QoS Requirements and Determination the Importance

The impact relationships are calculated using Eq. (1) in Sect. 3.1 to calculate the importance of web service design attributes. The results are shown in Table 4. The results are TFNs. We use Eq. (4) to transfer the fuzzy numbers into crisp numbers to be used in TCI at the same time. The importance reflects the contributions of the design attributes to TCI. The crisp importance of design attributes are normalized in the range of [0, 1] by Eq. (13).

$$N(aw_h) = E(aw_h)\left(\frac{\sum_{j=1}^{N} E(aw_h)}{N}\right)$$ (13)
4.3 Step 4: Evaluation of Technical Performance of Competitive Web Services in Technical Competitive Assessment Matrix

After determining the importance of web service design attributes, QFD team should evaluate the technical performance of similar web services on the market from the technical view, and establish data of design attributes for further analysis and comparison. It is mainly done through market research.

Here we use evaluation data about 7 competitive web services [13] shown in technical assessment matrix of the HoQ for a web service system in Fig. 3.

The maximum and minimum values of each design attributes among competitive web services are shown in Table 5. They reflect state of arts of the competitive web services, and they are used to normalize design attributes and limit the search space of target solutions. “Direction” in Table 5 means improvement direction of design attributes.

Each design attributes are normalized in [0, 1] so that all of them are positively related to Technical Competitive Index. We normalize them by Eq. (6) and Eq. (7) and calculate TCI of competitive web services by Eq. (5). The results are shown in Table 6. C5 has the maximum TCI (76.87%) which indicate the most competitive among similar web services. So our improvement goal is to determine targets solutions which make the improved web service has higher TCI than TCI (C5) within the fewest improvement efforts.

4.4 Step 5: Determination Targets of Web Service Design Attributes with Fewest Efforts to Improve

We use the approach described in Sect. 3.2 to find the technical target solutions of design attributes of a web service $W_0$. The solutions have the fewest efforts to improve the technical performance so that the web service $W_0$ is more competitive than any other similar products on the market. The development efforts indicate the development resources to be used including cost, time, human resources and so on.

Here we just consider cost to illustrate our approach. The mathematical model in Sect. 3.2 is instanced as following:

$$\min \sum_{j=1}^{N} [C(x_{i,j}) \times x_{i,j}] \quad \text{(14)}$$

subject to

$$TCI(W_0) \geq G, 0 < i \leq K, \max[TCI(W_i)] < G \leq 100 \quad \text{(15)}$$

$$\min_{0 \leq i \leq K} x_{i,j} \leq x_{i,j} \leq \max_{0 \leq i \leq K} x_{i,j} \quad \text{(16)}$$

where there are $K = 7$ competitive web services and $N=10$ web service design attributes, $W_0$ means the web service to improve. $C(x_{i,j})$ is the unit cost of implementation of improvement of the $i^{th}$ web service design attribute. The minimum and maximum values of each design attribute are shown in Table 5.

We let $G = 76.88$, a little higher (0.01) than $\max[TCI(W_i)] (76.87)$. The optimization target solution is shown in Table 7. To make TCI of the improved web service higher than other competitive web services, the fewest cost is 13094.19, and TCI of the improved web service is 76.89, 0.1 higher than the G that we set and 0.2 higher than $\max[TCI(W_i)]$.

S1 has the fewest cost to make TCI of the improved web service higher than other competitors, but may not have
better profitability of an investment. We introduce an index, named ROI (Return On Investment) to measure the return rate of target solutions.

The ROI formula is defined as following:

\[ ROI = \frac{TCI}{Total \ Cost} \]  

We assess other solutions with the goals increasing by the step size of 5. The total cost, improved TCI and ROI of these solutions are shown in Table 8 and Fig. 4.

The ROI decision analysis diagram is presented in Fig. 4. The improvement goal, the development cost and ROI are put together in the single diagram, giving decision makers an intuitive and detailed analysis of improvement solutions. ROI decision analysis diagram are provided to compare the efficiency of investments on different solutions. The optimization solutions and their comparison provide web service providers basis and support for decision-making of development and improvement of their web service products to win the market according to their technological and financial limitations. We use a web service system as an example to illustrate the approach and analyze the output of the approach to demonstrate its effectiveness, and the approach can be easily applied to practical situations.

**5. Conclusions**

This paper applies QFD to web service design, which traces customers’ QoS requirements into targets of web service design attributes and fulfills requirement driven web service design. It transfers the priority of necessity of QoS requirements into the priority of improvement of design specifications. Fuzzy set theory is used to deal with ambiguity and imprecision of subjective evaluation in the process of QFD. During the competitive evaluation and decision analysis of improvement target solutions, we introduce TCI to measure technical performance of web services and to compare them. Improvement target solutions with different goals are provided to make improved web service more competitive than any other similar web services on the market within the fewest efforts. ROI decision analysis diagram are provided to compare the efficiency of investments on different solutions. The optimization solutions and their comparison provide web service providers basis and support for decision-making of development and improvement of their web service products to win the market according to their technological and financial limitations. We use a web service system as an example to illustrate the approach and analyze the output of the approach to demonstrate its effectiveness, and the approach can be easily applied to practical situations.

**Acknowledgments**

This paper is supported by the Project of the State Key Laboratory of Software Development Environment under Grant No.SKLSDE-2012ZX-13.

**References**


Gang Wang was born in 1982, Ph.D. Candidate in the State Key Laboratory of Software Development Environment, University of BUAA, Beijing, China. His research interests are service-oriented software engineering, requirements engineering, and software architecture.

Li Zhang was born in 1968, Professor in the College of Computer Science and Engineering, University of BUAA, Beijing, China. Her research interests are software engineering, requirements engineering, software architecture structure, and process modeling and optimization.

Yonggang Huang was born in 1983, Ph.D. Candidate in the State Key Laboratory of Software Development Environment, University of BUAA, Beijing, China. His research interests are service-oriented software engineering, machine learning and pattern recognition.

Yan Sun was born in 1978, Ph.D. Candidate in the State Key Laboratory of Software Development Environment, University of BUAA, Beijing, China. His research interests are service-oriented software engineering, system architecture, formal verification, and model checking.