A QOS-AREWARE APPROACH FOR WEB SERVICE SELECTION BASED ON PROBABILITY EVALUATION

Dessislava Petrova-Antonova
Faculty of Mathematics and Informatics, Sofia University “St. Kliment Ohridski”
125, Tzarigradsko shosse Blvd., 1113 Sofia, Bulgaria

ABSTRACT
The rising number of available web services with the same functionality hampers the web service discovery process and complicates clients while searching for a web service that best meets their requirements. That is why the web service description and discovery need to concern not only functional but also non-functional (QoS) properties of web services. This paper contributes to that challenge by proposing a QoS-aware approach for web service selection. The QoS data collected during web service executions is used for prediction of future values of QoS properties, based on probability evaluation. As a result, the web service with the largest probability to satisfy the requirements of the client is identified and recommended to him/her as the best one.

KEY WORDS
Probability, QoS, selection, web service.

1. INTRODUCTION
Service Oriented Architecture (SOA) provides a new generation of software architectures that attracts attention as a promising way for smooth integration of loose coupled software applications. The web services provide a popular technology that overcomes the limitations of traditional middleware and can be used to implement the SOA model. The current web service architecture is based on Web Service Description Language (WSDL) and Universal Description Discovery and Integration (UDDI) standards that support functional web service description, publication and discovery. Due to the rising number of web services that share the same functionality, often clients are complicated while searching for a web service that best meets their requirements. That is why many research efforts focus on non-functional, also called Quality of Service (QoS), properties of web services.

The QoS can be applied over the whole process of web service description, publication and discovery. In this paper, it is utilized to differentiate a single web service, which best meets client requirements among multiple web services with the same functional properties. The QoS data collected during web service executions is used for prediction of the future values of QoS properties, based on probability evaluation. Thus, the client will be able to select a web service with the largest probability of having QoS properties that are as close as possible to the preliminary defined requirements.

The remainder of this paper is structured as follows. In Section 2, some background information related to QoS properties and discrete random variables is presented. In Section 3, a QoS-aware approach for web service selection is described in details. In Section 4, an illustrative example of the proposed approach is given. Section 5 introduces related work. Finally, Section 6 concludes the paper.

2. BACKGROUND
This section presents a preliminary knowledge, including QoS specification and some aspects of the theory of discrete random variables. Section 2.1 provides several classifications of QoS properties, while in Section 2.2, the probability functions applied in the proposed approach for web service selection are shown.
2.1. QoS properties

QoS properties present the non-functional characteristics of web services. They have significant role in web service discovery process, helping clients to select a web service that meets best their requirements.

Various classifications of QoS properties exist in literature. The QoS properties can be distinguished on the basis of their values. If the client requires the values of QoS properties to be as higher as possible, then these QoS properties can be classified as increasing or positive. Otherwise, they are classified as decreasing or negative. Examples for positive QoS properties are Availability and Throughput. In contrast, Response time and Cost are negative QoS properties. Such classifications are given by Tran et al (2009) and D’Mello et al (2008).

As not all of the QoS properties can be quantified, Tran et al (2009) classify the QoS properties as unit and unitless. For example, QoS properties, such as Supported standards, Authentication method and Data encryption are unitless, because they do not have numeric values. Similarly, Kritikos and Plexousakis (2009) distinguish between measurable and unmeasurable QoS properties. They propose a categorization of QoS properties dividing them in two sets, namely domain-independent and domain-dependant. The domain-independent QoS properties refer to the technical aspects of web services without respect to application domain. They are organized in seven categories, as follows: Performance, Dependability, Integrity, Security, Network and infrastructure related, Cost, and Other. The last category is created to include the QoS properties that are not covered by any other category. The domain-dependant QoS properties rely on the application domain of web services. For example, the domain-dependant QoS properties for the application domain of Vehicle Traffic Monitoring are covered area, routes set, detail level, accuracy, completeness, validity, timeliness, and coverage. They characterized the data processed and produced by the monitoring system.

2.2. Discrete random variables

The probability theory finds applications in every area of scholarly activities as well as in daily experience like weather and stock market prediction. In any experiment there are a finite number of possible outcomes. For example, the values of QoS properties recorded in some history log at a given time can be seen as such possible outcomes. Each outcome can be associated with a number according to the specific rule of association called random variable (Devore 2008). Normally, random variables are denoted with uppercase letters and their values are denoted with lowercase letters. Each random variable can be discrete or continuous. The possible values of discrete random variable form a finite set or infinite sequence having a first element, a second element, and so on. The possible values of continuous random variables consist of an entire interval on the number line (Devore 2008). In our sample a given QoS property such as Response time can be represented with a discrete random variable $X$ with several possible values $x$ that correspond to the recorded values in the history log.

The random variable can be described by a probability mass function (PMF), which captures the probabilities of the values that the random variable can take. If $x$ is any possible value of discrete random variable $X$, the PMF of $x$, denoted $p_X(x)$ is the probability of the event $\{X = x\}$ consisting of all outcomes that give rise to a value of $X$ equal to $x$ (Bertsekas and Tsitsiklis 2000):

$$p_X(x) = P(\{X = x\})$$

For example, let the experiment consist of six independent measurements of the Response time of a particular web service and let $X$ be the value measured. Let the event $\{X=1.8\}$ is happened one time, the event $\{X=2\}$ is happened four times, and the event $\{X=3\}$ is happened one time. Then the PMF of $x$ is:

$$p_X(x) = \begin{cases} 
\frac{1}{6} & \text{if } x = 1.8 \\
\frac{2}{3} & \text{if } x = 2 \\
\frac{1}{6} & \text{if } x = 3 \\
0 & \text{otherwise}
\end{cases}$$
Sometimes the information that the PMF gives for a given random variable \( X \) is needed to be summarized in a single representative number. This is accomplished by an expected value of \( X \), denoted \( E(X) \) or \( \mu_X \):

\[
E(X) = \mu_X = \sum_x x p_X(x)
\]  
(2)

For the example above, the expected value of the Response time is calculated as follows:

\[
E(X) = \mu_X = 1.8 \cdot \frac{1}{6} + 2 \cdot \frac{2}{3} + 3 \cdot \frac{1}{6} = 0.3 + 1.33 + 0.5 = 2.13
\]

For a given value \( x \) it is often necessary to compute the probability that the value of the random variable \( X \) will be at most \( x \). This can be accomplished by a cumulative distribution function (CDF), denoted \( F_X \). The CDF of a discrete random variable provides the probability \( P(X \leq x) \) (Bertsekas and Tsitsiklis 2000):

\[
F_X(x) = P(X \leq x) = \sum_{x \geq x} p_X(k)
\]  
(3)

Since the two events \( \{X \leq x\} \) and \( \{X > x\} \) are mutually exclusive, the sum of their probabilities is equal to 1. Therefore:

\[
P(X > x) = 1 - F_X(x)
\]  
(4)

For the example above, the probability that the Response time will be at most 2 sec is calculated as follows:

\[
F_X(x) = P(X \leq 2) = \frac{1}{6} + \frac{4}{6} + \frac{5}{6}
\]

The probabilistic models usually concern several random variables. For example, in the context of web services, the QoS requirements of the clients consist of definitions for several QoS properties that can be seen as random variables belonging to the same experiment. The joint CDF of two random variables is as follows:

\[
F_{X,Y}(x,y) = P(X \leq x, Y \leq y) = F_X(x)F_Y(y)
\]  
(5)

For example, if the probability that the Response time is at most 2 sec is \( \frac{5}{6} \) and the probability that the Throughput is at most 160 kbps is \( \frac{2}{5} \), then the joint CDF of these two QoS properties is calculated as follows:

\[
F_{X,Y}(x,y) = P(X \leq 2, Y > 160) = P(X \leq 2) \cdot (1 - P(Y \leq 160)) = \frac{5}{6} \cdot \frac{3}{5} = \frac{1}{2}
\]

The equation 5 can be used for calculation of probability that the QoS properties of a particular web service will satisfy the client’s requirements. Such calculations can be accomplished for all candidate web services that meet functional requirements of the client. After that the client will be able to select a web service with a maximum probability of having QoS properties that are as close as possible to the preliminary defined requirements. The differentiation among web services with the same probability to meet client’s requirements can be accomplished by comparison of their expected values of QoS properties calculated according to equation 2.

### 3. A QoS-Aware Approach for Web Service Selection

This section presents a QoS-aware approach for web service selection based on probability evaluation. The approach consists of eight steps that are described bellow in detail.

Let the candidate web services, namely those that satisfy functional requirements of the client, form the set \( S \):

\[
S = \{S_1, S_2, ..., S_i, ..., S_n\}, i = 1 + n
\]

where \( n \) is the number of candidate web services.

Let the QoS properties, whose values are stored in history log, form the set \( Q \):
\[ Q = \{q_1, q_2, \ldots, q_l\}, j = 1 + l \]

where \( l \) is the number of QoS properties.

**Step 1 Data extraction from history log.**

As mentioned in Section 2.2 each QoS property of web services can be represented with a discrete random variable with several possible values that correspond to the recorded values in the history log. Thus, the values of QoS properties of a particular web service form the following matrix:

\[
V_{i} = \begin{bmatrix}
q_1 & q_2 & \cdots & q_l \\
x_{11} & x_{12} & \cdots & x_{1l} \\
x_{21} & x_{22} & \cdots & x_{2l} \\
\vdots & \vdots & \ddots & \vdots \\
x_{k1} & x_{k2} & \cdots & x_{kl}
\end{bmatrix}
\]

where \( x_{kl} \) is the \( k \)-th value of QoS property \( q_l \) obtained from history log and \( k \) is the number of values obtained.

**Step 2 Extraction of events and priorities from client’s requirements.**

The QoS requirements of the client are considered as events. They form the following vector:

\[ R = [r_1 \ r_2 \ \cdots \ r_l] \]

where \( r_i = \{q_i \leq y_i\} \) or \( r_i = \{q_i > y_i\} \).

The client also specifies a priority of each QoS property from his/her perspective. The vector \( W \) presents these priorities and is defined as follows:

\[ W = [w_1 \ w_2 \ \cdots \ w_l] \]

where \( w_i \) is the priority of QoS property \( q_i \). The priorities are integer values from \( l \) to the number \( n \) of QoS properties, so that \( l \) is the maximum priority and \( n \) is the minimum priority.

**Step 3 Calculation of PMF for each value of each QoS property for all candidate web service.**

The probabilities of the values that the QoS property takes are calculated according to equation 1. They are presented with the following vector:

\[ P_{x_{ij}} = [p(x_{1i}) \ p(x_{2i}) \ \cdots \ p(x_{mi})], j = 1 + l \]

where \( p(x_{mi}) \) is the probability that the value of QoS property \( q_i \) will be \( x_{mi} \). Note that \( m \) is the number of unique values obtained from history log for a given QoS property.

**Step 4 Calculation of CDF for each QoS property for all candidate web service.**

The CDF of each QoS property is calculated according to equation 3. In case of \( r_i = \{q_i \leq y_i\} \) the CDF of QoS property \( q_i \) is as follows:

\[ F_{q_i}(y_i) = P(q_i \leq y_i) \]

In case of \( r_i = \{q_i > y_i\} \) the CDF of QoS property \( q_i \) is as follows:

\[ F_{q_i}(y_i) = 1 - P(q_i \leq y_i) \]

The CDFs calculated for each QoS property of all candidate web services form the following matrix:

\[
F_q = \begin{bmatrix}
F_{q_1} & F_{q_2} & \cdots & F_{q_l} \\
S_1 & F_{11} & F_{12} & \cdots & F_{1l} \\
S_2 & F_{21} & F_{22} & \cdots & F_{2l} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
S_n & F_{n1} & F_{n2} & \cdots & F_{nl}
\end{bmatrix}
\]

**Step 5 Calculation of joint CDF for all candidate web services.**

Finally, for each web service the joint CDF is calculated according to equation 5. The result is a vector \( F \) of probabilities:

\[ F = [F_{S_1} \ F_{S_2} \ \cdots \ F_{S_n}] \]

where \( F_{S_n} \) is the probability that web service \( S_n \) will satisfy the client’s requirements.
Step 6 Finding of web service with the maximum probability to satisfy client’s requirements.
The web service with the maximum probability to satisfy the client’s requirements is then proposed to the client. Here, several well known algorithms for finding largest element in an array are applicable.

Step 7 Calculation of expected values of each QoS property for all candidate web services.
Suppose that there are several web services with the same probability to satisfy client’s requirements. They form the set $S'$, which is the subset of $S$:

$$S' = \{S_1, S_2, \ldots, S_{i'}, \ldots, S_t\}, i = 1 + t$$

where $t$ is the number of web services that have probability to satisfy client’s requirements.

The expected value for each QoS property of those web services is calculated according to equation 2. These expected values form the following matrix:

$$E = \begin{bmatrix}
q_1 & q_2 & \cdots & q_t \\
S_1 & E_{11} & E_{12} & \cdots & E_{1t} \\
\vdots & \vdots & \ddots & \vdots \\
S_t & E_{t1} & E_{t2} & \cdots & E_{tt}
\end{bmatrix}, i = 1 + t$$

where $E_{ij}$ is the expected value of QoS property $q_j$ of web service $S_i$.

Step 8 Finding of web service with the best expected values of QoS properties.
The rows of the matrix $E$ are sorted according to the values of vector $F$, so that the web services with the maximum probability to satisfy the client’s requirements are shifted at the top of the matrix $E$. Next the columns of the matrix $E$ are sorted according to the values of vector $W$. Practically, each value of the vector $W$ shows the target position of a particular column in $E$ during sorting. Then the rows of the matrix $E$ are sorted according to the values of the first column. For any rows that have equal elements in a particular column sorting is based on the column immediately to the right. The sorting is in descending or ascending order depending on definition of the requirement for the QoS properties identifying the corresponding column. In case of $r_l=\{q_{l_1} \leq y_l\}$ the sorting is in ascending order. Otherwise, in case of $r_l=\{q_{l_1} \geq y_l\}$ the sorting is in descending order. As a result, the web service with the best expected values of QoS properties is shifted on the top of the matrix $E$ and is proposed to the client.

The pseudo code of the algorithm that implements the proposed approach is presented on Figure 1.

```
1  FOR EACH web service in S DO
2      Extract data from history log
3      Create matrix V
4  END FOR
5  FOR EACH QoS property DO
6      Extract an event from client’s requirement
7      Extract a priority from client’s requirement
8  END FOR
9  FOR EACH web service in S DO
10     FOR EACH QoS properties in P DO
11        FOR EACH unique value of QoS property DO
12            Calculate PMF
13        END FOR
14     Calculate CDF
15  END FOR
16  Calculate joint CDF
17  END FOR
18  Find web service with maximum joint CDF
19  IF there are web services with the same CDF THEN
20     FOR EACH web service in $S'$ DO
21         FOR EACH QoS properties in P DO
22             Calculate expected value
23         Put value in the matrix E
24     END FOR
25  END FOR
```
26 END IF  
27 Sort the matrix E  
28 Extract the web service with the best expected values

Figure 1. A pseudo code of the algorithm for QoS-aware web service selection

4. ILLUSTRATIVE EXAMPLE

In this section, an example that demonstrates the proposed approach is presented. Three web services, named $S_1$, $S_2$ and $S_3$, are considered. The QoS properties that will be used in the example are Successful execution rate ($q_1$), Reputation ($q_2$), Price ($q_3$), Availability ($q_4$) and Execution time ($q_5$).

The values of QoS properties of the web services $S_1$, $S_2$ and $S_3$ are obtained from history log used by Qi et al. (2010) to demonstrate a method for QoS-aware web service selection based on credibility evaluation.

According to Step 1 of the proposed approach, the values of QoS properties of each web service form the following matrices:

$V_{S_1} = \begin{bmatrix}
0 & 0.90 & 0.80 & 1.00 & 0.05 \\
0 & 0.95 & 0.80 & 0.95 & 1.00 \\
1 & 0.60 & 0.90 & 0.90 & 0.05 \\
0 & 0.95 & 0.80 & 4.00 & 0.60
\end{bmatrix}$

$V_{S_2} = \begin{bmatrix}
0 & 0.95 & 0.85 & 1.00 & 0.05 \\
0 & 0.90 & 0.85 & 0.95 & 1.00 \\
1 & 0.60 & 0.90 & 0.90 & 0.05 \\
0 & 0.95 & 0.80 & 4.00 & 0.60
\end{bmatrix}$

$V_{S_3} = \begin{bmatrix}
0 & 0.95 & 0.85 & 1.00 & 0.05 \\
0 & 0.90 & 0.85 & 0.95 & 1.00 \\
1 & 0.60 & 0.90 & 0.90 & 0.05 \\
0 & 0.95 & 0.80 & 4.00 & 0.60
\end{bmatrix}$

On the Step 2 the client’s requirements are extracted. The events that are obtained from the requirements form the following vector:

$R = \{q_1 = 0\} \quad \{q_2 > 0.8\} \quad \{q_3 < 14\} \quad \{q_4 > 0.85\} \quad \{q_5 < 5\}$

The priorities of QoS properties are presented as follows:

$W = [1 \ 3 \ 4 \ 5 \ 2]$  

According to Step 3, the probabilities of the values that the QoS property of each web service takes are calculated. They are used to compute the CDF of each QoS property of all candidate web services on the Step 4. As a result, the following matrix is defined:

$F_q = \begin{bmatrix}
0.875 & 0.750 & 0.750 & 0.625 & 0.625 \\
0.875 & 0.750 & 0.375 & 0.875 & 0.625 \\
0.750 & 0.750 & 0.625 & 0.750 & 0.500
\end{bmatrix}$

On the step 5 the joint CDF is calculated. The vector $F$ of probabilities is as follows:

$F = [0.192 \ 0.135 \ 0.132]$  

Finally, on the step 6 the web service with the maximum probability to satisfy the client’s requirements is obtained. As can be seen from the vector $F$ presented above this web service is $S_1$.

In order to present all steps of the proposed approach, suppose that web services $S_1$ and $S_2$ have the same probability to satisfy the client’s requirements. According to step 7, the expected value of QoS properties for those services need to be calculated. They are presented in the following matrix:

$E = \begin{bmatrix}
q_1 & q_2 & q_3 & q_4 & q_5 \\
0.850 & 12 & 0.869 & 5.075 \\
0.825 & 15 & 0.881 & 5.250
\end{bmatrix}$

The matrix $E$ is sorted according to the algorithm that is described in the step 8. As a result, the web service $S_1$ is proposed to the client.
5. RELATED WORK

QoS properties are crucial for selecting a web service that best meets the client’s requirements, especially when there are several web services with the same functionality. Therefore, many approaches for QoS-enabled web service selection exist.

Li et al (2010) propose a web service selection approach that is based on the weight of client’s satisfaction from QoS properties of a particular web service. Also, a mathematical model for definition of QoS constraints according to the preferences given by the client is designed. The main drawback of the approach is the inability to ensure that the service recommending algorithm is open, fair and trustworthy. Also, only measurable QoS properties are considered. In contrast, a web service selection mechanism, proposed by D’Mello et al (2008), deals with all types of QoS properties expressing those that are not measurable in terms of integer values. Here, the constraints QoS are presented in a weighted AND-OR tree.

Sha et al (2009) describe QoS based selection model. It is based on the overall QoS of web service that is obtained from a weighted sum of the normalized values of QoS properties. One of the drawbacks of the model is that it considers only measurable QoS properties that can be directly monitored. Also, the normalization of the values of QoS properties in the interval of [0,1] leads to losing valuable information.

Qi et al (2010) have designed a QoS-aware selection method based on credibility evaluation of the web services. The QoS properties are classified in two categories, i.e. negotiable and nonnegotiable. The values of nonnegotiable QoS properties are obtained from historical records of web service execution and cannot be modified by the provider. The negotiable QoS properties can be changed according to the client’s requirements. The credibility evaluation depends on whether the QoS property is negotiable or nonnegotiable. Li et al (2009) propose a web service selection algorithm, based on quantitative QoS prediction method applied to a dynamic environment. The QoS measurement of composite web services is modeled by structural equation in order to measure accurately the changes of QoS properties.

Shao et al (2007) have designed a collaborative filtering based approach that predicts QoS of unused web services taking into account the similarity among experience of clients. The similarity of clients is considered also in a web service selecting model proposed by Wang and Chen (2008). The model is based only on measurable QoS properties of client-side.

Xiong and Fan (2007) propose a QoS-aware selection method of web services based on adoption of fuzzy multiple criteria decision making. The subjective weights of QoS properties reflecting human rating and objective weights representing reliability of evaluation are used to form synthetic weights. Then the overall quality score of a particular web service is calculated using the synthetic weights of QoS properties. Similarly, Tran et al (2009) adopt Analytic Hierarchy Process (AHP) method as an underlying mechanism for QoS-based web service ranking.

Xu et al (2007) present QoS-based web service selection approach that is based on ratings indicating the level of client’s satisfaction with a web service following certain interaction with it. A drawback of the approach is the assumption that the ratings are objective and valid. Thus, the web service selection process becomes untrustworthy. Al-Masri and Mahmoud (2007) define web service relevancy function in order to measure the relevancy ranking of a particular web service. The function calculates the distance between a particular QoS value and the maximum normalized value in its corresponding set.

The approach proposed in the paper has some advantages compared with those described above. Firstly, it considers measurable as well unmeasurable QoS properties. This is due to the fact that the values of QoS properties are presented as events with a probability to happen. Further on, some approaches are based on the ratings that are specified by clients after web service consumption. Other approaches use values of QoS properties that are claimed by the web service providers. In contrast, the approach proposed in the paper relies on values of QoS properties obtained during web service execution and stored in a history log. Therefore, this approach appears to be more reliable. Finally, the proposed approach allows clients to specify the priorities of the QoS properties, and this leads to the possibility of distinguishing the services with the same probability to satisfy clients’ requirements.
6. CONCLUSION

In this paper, a QoS-aware web service selection approach is presented. The QoS data collected during web service executions is used for prediction of future values of QoS properties based on probability evaluation. Thus, the client will be able to select a web service with the largest probability of having QoS properties that are as close as possible to the preliminary defined requirements.

Future work includes further validation of the proposed approach with different QoS data. The big challenge is to deal with composite web services. Therefore, we plan to design an approach predicting the overall quality of a web service composition, based on the expected quality of the web services participating in that composition.

ACKNOWLEDGEMENT

The work presented in this paper was supported by the Human Resource Development Operational Programme 2007-2013 of the European Social Fund under Grant No. BG 051PO001-3.3.04/13 and the National Scientific Fund of Bulgarian Ministry of Education and Science under Grant No. MU-01-143.

REFERENCES


