

# Extending WSDL and UDDI with Quality Service Selection Criteria

Amna Eleyan , Liping Zhao

Birzeit University, University of Manchester

Palestine, United Kingdom

**Abstract**— Quality criteria play an important role in Web Services as they differentiate similar services by qualities. Quality-based web services enable service requesters to choose and bind to a suitable Web service at run time based on their preferred quality criteria.

There are many quality criteria that are important to Web services. This paper proposes a quality criteria classification that organizes web services qualities into four groups: performance, failure probability, trustworthiness and cost. The quality criteria classification is specified within the Web Service Description Language (WSDL). The paper demonstrates an approach that enables the Universal Description, Discovery and Integration (UDDI) to help business partners to discover services based on quality criteria by extending the current Web service architecture with a quality server. The quality server uses a mathematical method to facilitate and assist the requester to discover and select the best available Web services.

**Index Terms**—Web services, quality criteria, WSDL, UDDI, quality server, mathematical model

## I. INTRODUCTION

### A. Motivation

WEB services is a technology, which allows applications to communicate with each other in a platform and programming language- independent manner over the Internet. Web services achieve system interoperability by exchanging an application development and service interactions using the XML-based [1] standards such as Simple Object Access Protocol (SOAP) [2], Web Service Description Language (WSDL) [3] and Universal Description, Discovery and Integration (UDDI) [4].

With the growing popularity of Web services, a quality criteria support for Web services will play an important role for the success of this emerging technology. This paper proposes quality criteria classification that organizes quality criteria into four groups: performance, failure probability, trustworthiness and cost.

The current Web service core technologies (SOAP, WSDL, and UDDI) are immature and still under development by the W3C [5]. UDDI is just a registry database and allows service requesters to look for Web services based on their functionality but not quality information. WSDL is an XML format for describing Web services [6]. These technologies do not address issues related to the description of quality aspects of a service.

To overcome the WSDL and UDDI limitations, the

following approaches are introduced. We present an extension to the WSDL to include quality criteria classification and we extend the current Web service architecture [7], [8] with quality server to enable the UDDI to publish and discover services based on the proposed quality criteria classification by using the mathematical method.

### B. Relationship between WSDL and UDDI

Web Services Description Language (WSDL) is a mechanism used to define and describe the details regarding the communication with Web services. Universal Description Discovery and Integration (UDDI) provides a method for publishing and finding service descriptions. The UDDI data entities provide support for defining both business and service information. The service description information defined in WSDL is complementary to the information found in a UDDI registry. The WSDL service interface definition is published in a UDDI registry as a tModel. Some of the tModel elements (such as *name* and *overviewURL*) are constructed using the information that is copied from the WSDL service interface definition. The WSDL service implementation definition is published in UDDI registry as a businessService with all relevant information copied into the businessService [9], [10]. Figure 1 illustrates the relationship between the WSDL and UDDI.

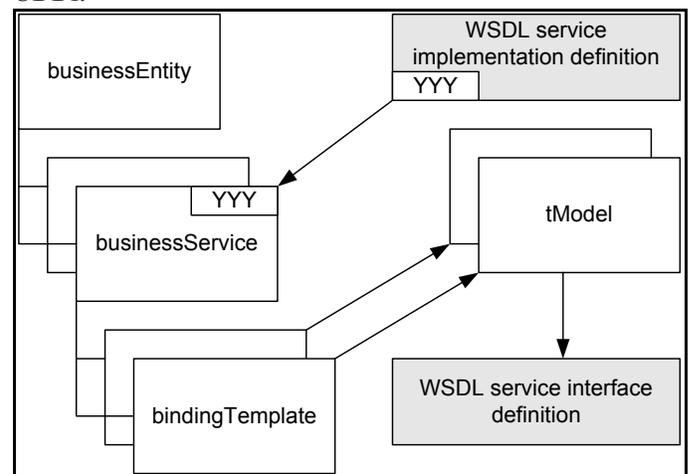


Figure 1. WSDL and UDDI Relationship

### C. Related Work and Our Contribution

Several research efforts have been made in the area of quality-based Web Services. Gouscos et al.[11] present a simple approach to model Web service QoS attributes and

provision price, and discuss how this information can be accommodated within basic specification standards such as WSDL and exploited within the Web service deployment and application life-cycle. Chen et al.[12] propose UX (UDDI eXtension), a system that is QoS-aware and facilitates the federated discovery for Web services. The QoS feedback from service requesters are used to predict the service's performance. UX server supports wide area discovery across domains. The UX server's inquiry interface conforms to the UDDI specification. A discovery export policy is proposed that controls how the registered information is exported to UX servers and requesters. Farkas et al.[13] propose a Web Service QoS Extension Language (WQEL) schema for defining the QoS parameters of the service and extended UDDI Inquiry API with a QoS Broker API. The QoS Broker is used to choose the best available web service component. Adams and Boeyen [14] present a framework for implementing security for Web services by extending UDDI and WSDL. The framework includes security of UDDI itself and security of Web services transactions. Extensions to the schema for both UDDI and WSDL are identified, as well as extensions to the security of the publication and discovery mechanism. Ali et al. [15] extend UDDI as "UDDIe" which supports the notion of "blue pages". UDDIe enable discovery of services based on QoS attributes by extending the businessService class in UDDI with propertyBag. . Ran [16] Extends UDDI data structure with qualityInformation data structure under the businessService data structure. The author organizes the QoS attributes into groups: QoS related to runtime, transaction support, configuration management and cost and security.

In this paper, we propose a quality criteria classification and specify it within the WSDL. An approach is presented to enable the current UDDI to publish and discover services based on the proposed quality criteria classification by extending the current Web service architecture with quality server. Also, the quality server uses the mathematical method to select the best service based on quality criteria.

## II. QUALITY CRITERIA IN WEB SERVICES

### A. Quality Definition

Quality criteria may have different definitions in different domains. However, in the Web services context, Quality criteria can be defined as a set of non-functional criteria [17] such as availability, performance and reliability that impact the performance of Web services.

Quality is the measure of how well does a particular service perform relative to expectations, as presented to the requester. It determines whether the requester will be satisfied with the service delivered, that is, the quality is meeting requirements.

### B. Quality Criteria Classification

The quality criteria classification in this paper is similar to the quality classification in [18], [16] and [19] in that they classify the quality criteria into groups with different

perspectives. The quality classification in [18] includes three groups: performance, safety and cost. Performance contains response time and throughput, safety contains availability and reliability and cost contains the service cost. The quality classification in [16] organizes the most important quality-of-service to Web services into four groups: QoS related to runtime, transaction support, configuration management and cost and security. The quality classification in [19] classifies the QoS parameters into the following groups: general, Internet service specific and task specific. General QoS parameters contain performance (throughput), performance (latency), reliability and cost. Internet service specific QoS parameters contain availability, security, accessibility and regulatory. Task specific QoS parameters contain task specific parameter.

This section represents a quality criteria classification that organized into four groups: performance, failure probability, trustworthiness, and cost as shown in Figure 2. These groups are organized regarding its characteristics and include generic criteria. The generic criteria are applicable to all Web services, reusable across domains (e.g., business specific- criteria domain) and can benefit all service requesters.

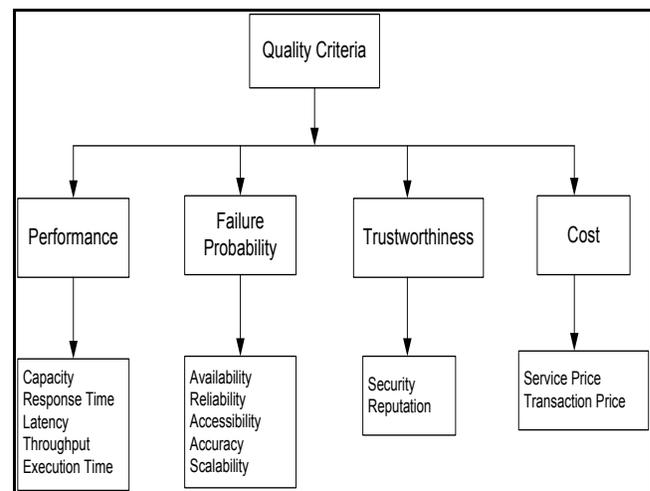


Figure 2. Quality Criteria Classification

### Performance

The performance of a Web services measure the speed in completing a service request. It can be measured by:

*Capacity*-The limit of concurrent requests that the service support for guaranteed performance.

*Response time*- The maximum time that elapses from the moment that a web service receives a SOAP request until it produces the corresponding SOAP response [11]. Response time is positively related to capacity [16].

*Latency*-The round-trip time between the service request arrives and the request is being serviced [20].

*Throughput*- The number of Web service request completed at a given time period [21]. It is the rate at which a service can process requests. Throughput is related negatively to latency and positively to capacity.

*Execution (processing) time-* The time taken by a Web service to process its sequence of activities [21].

In general, high performance Web services should provide higher throughput, higher capacity, faster response time, lower latency, and lower execution duration.

#### **Failure Probability**

The failure probability is the probability of a Web service being incapable to complete a service SOAP request within the maximum response time corresponding to this request [11]. The failure probability is composed of:

*Availability--* The probability that a service is operating when it is invoked. Associated with the availability is the time-to-repair (TTR) property, addressing the time taken to repair a service [20]. Availability is related to accessibility and reliability. Availability can be measured by the following formula:

$P_{\text{availability}} = C(X)/N$ , where  $C(X)$  is the number of successful executions;  $N$  is the total number of invocations.

Time-to-repair (TTR) can be measured by the following formula:

$TTR = t_{\text{restart}}(X) - t_{\text{failed}}(X)$ , where  $t_{\text{failed}}$  is the timestamp when the service  $X$  failed,  $t_{\text{restart}}$  is timestamp when service was restarted [19].

*Reliability-* It is the probability of a service to perform its required functions under stated conditions within a maximum expected time interval [16]. It refers to the assured and ordered delivery for messages being sent and received by service requesters and service providers [20]. Reliability can be measured by the following formula:

$R = 1 - P(\text{success})$ , where  $P(\text{success})$  is the number of successful execution/ $N$ ,  $N$  is the total number of invocations [19]. Reliability may also be measured by: Mean time between failure (MTBF), Mean Time to Failure (MTF), and To Transition (MTTT) [16]. Reliability is closely related to availability.

*Accessibility-* It is the capability of serving the Web Service request. The Web service might be available but not accessible because of a high volume of requests [20]. Accessibility can be represented by the following formula:  $P_{\text{accessibility}} = P_{\text{availability}}$  at Time  $T = t$  [19].

*Accuracy-* The amount of errors produced by the service during completing of the work [16].

*Scalability-* The capacity of increasing the computing capacity of service provider's computer system and system's ability to process more operations or transactions in a given period of time. It is closely related to performance and throughput [16].

#### **Trustworthiness**

Trust in general is a rational concept involving the trusted and the trusting parties. For example, on the eBay Web site, eBay is a trusted authority who authenticates the sellers in its auctions and maintains their ratings. However, eBay would be unable to authenticate parties who weren't subject to its legal contracts covering bidding and selling at its auctions [22]. Web services trustworthiness can be achieved when the

selected Web services components fulfill its requester needs or requirements ( i.e., functional and non-functional ) [23].

Web services trustworthiness can be measured by:

*Security* - It represents the measure of trustworthiness and can be provided by:

*Authentication:* Determining the identity of the sender [24]. Service requesters need to be authenticated by the service provider before sending information.

*Authorization:* Determining if the sender is authorized to perform the operation requested by the message [24]. That is, what the requester are permitted to access?

*Integrity:* message integrity is protecting the message content from being illegally modified or corrupted [25].

*Confidentiality:* confidential information is to ensure that information is protected against the access of unauthorized principals (users or other services) [26].

*Non-Repudiation:* to prove the identity of the originator of the SOAP message, and to prove the fact that they sent the message.

*Reputation-* It is the measure of trustworthiness of a service, based on the end user's experiences of using the service. Different end users may have different opinions on the same service. The reputation can be defined as the average ranking given to the service by the end users. The value of the

reputation is computed using the expression  $q_{rep} = \frac{\sum_{i=1}^n R_i}{n}$ ,

where  $R_i$  is the end user's ranking on a service's reputation,  $n$  is the number of times the service has been graded. Usually, the end users are given a range to rank Web services, for example, in Amazon.com, the range is [0,5] [27].

#### **Cost**

It is the cost charged by the service provider entity to the service client entity fro a request that is successfully responded [11]. Web service providers either directly advertise the service and its execution price, or they provide means to enquire about it [27] The cost value can be measured by:

*Service Cost-* It is the amount of money which a service requester has to pay to the service provider to use a Web service such as checking a credit, or the amount of money the service requester has to pay to the service provider to get a commodity like a monthly phone service [28]. It is the price of the actual service or products.

*Network transportation /Transaction Cost-* It is the cost involving in each requesting, invoking, and executing the service. This cost associated with the hardware and software needed to set up and run the service as well as to maintain and update the service and its interface [29].

The value of total cost per advertised service can be calculated by:

Total Cost = Service execution Cost+ (Network transportation/Transaction) Cost.

### C. The XML Schema for Quality Criteria Classification

The above quality criteria classification is specified within WSDL. Because WSDL is an XML based language, the proposed quality classification is implemented using XML Spy editor as shown in Figure 3.

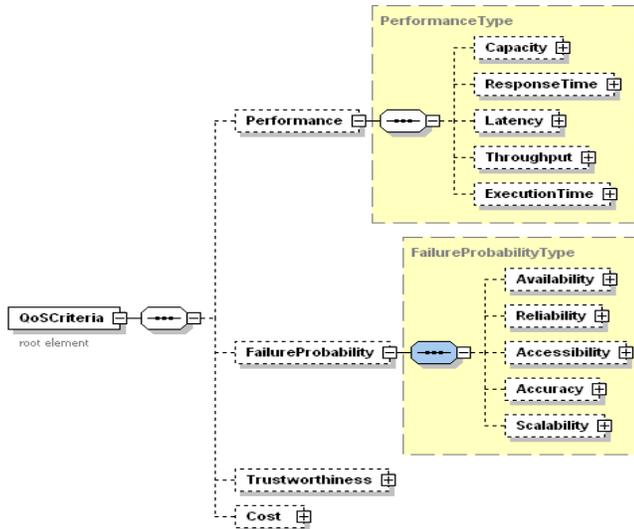


Figure 3. Structure of Quality Criteria Classification

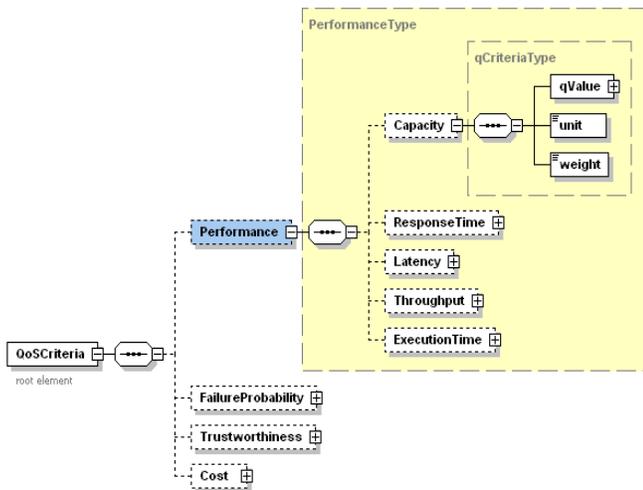


Figure 4. Properties of each sub-criterion element

Figure 4 shows the properties or child elements (qValue, unit, weight) for each sub-criterion. qValue has the value of sub-criteria, unit has enumerator values (Msec, Percentage, Request/sec, Pound and None), weight has value range between [0,1] and the default value is 1. qvalue includes further child elements (Min, Max, Preferred) and attribute called qllevel. Min, Max, and Preferred has the minimum, maximum and preferred values from the requester point of view. qllevel has enumerator values (High, Medium, and Low) which is the level of importance associated with every quality sub-criteria. For example, High value regarding the sub-criteria Availability is between [90, 99], whereas for Reputation is between [4, 5].

### III. EXTENDING WSDL WITH QUALITY CRITERIA

The Web Services Description Language (WSDL) is the current standard for specification of Web services. WSDL documents can be used to register services with the UDDI registry. There are two kinds of documents that are used while registering a service [10]. The first is known as the Service Interface Document that provides an abstract definition of a Web service and omits implementation details such as port address, communication protocol, etc. The other document is the Service Implementation Document that contains a description of a service that implements a service interface.

But, even WSDL is an XML format for describing Web services, it does not address issues related to the description of quality aspects of a service [30]. In this paper, WSDL is extended to accommodate quality criteria of the proposed quality criteria classification that described in Section II.B. The quality criteria extension is made in the *Service Implementation Document* part as extended in [11],[31]. Figure 5 shows an example of quality requirements by extending Amazon Web service WSDL with quality criteria classification. Amazon Web service WSDL document can be retrieved from the URL: <http://webservices.amazon.com/AWSECommerceService/AWSECommerceService.wsdl>. Amazon Web Service or Amazon E-Commerce Service (ECS) [32] provides many request operations to look up Amazon products. Two request operations are selected : *ItemSearch* and *ItemLookup*. WSDL is extended by augmenting Quality Criteria XML Schema that described in Section II.C in the <service> element that is in the *service implementation definition* part.

The service requester as shown in Figure 5 selects availability in failure probability group, reputation in trustworthiness group and service price in cost group. He/She selects availability with properties: qllevel=High, preferred value 95 and weight=0.5, reputation with properties: qllevel=High, preferred value=4.5 and weight=0.3 and service price with properties: qllevel=Medium, preferred value= 40 and weight=0.2.

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://webservices.amazon.com/AWSECommerceService/2006-02-15" targetNamespace="http://webservices.amazon.com/AWSECommerceService/2006-02-15">
  <message name="ItemSearchRequestMsg">
  </message>
  <message name="ItemLookupRequestMsg">
  </message>
  <portType name="AWSECommerceServicePortType">
    <operation name="ItemSearch">
      <input message="tns:ItemSearchRequestMsg"/>
    </operation>
    <operation name="ItemLookup">
      <input message="tns:ItemLookupRequestMsg"/>
    </operation>
  </portType>
  .....
  <service name="AWSECommerceService">
    <port name="AWSECommerceServicePort" binding="tns:AWSECommerceServiceBinding">
      <soap:address location="http://soap.amazon.com/onca/soap?Service=AWSECommerceService"/>
    </port>
    <QoSCriteria>
      <FailureProbability>
      <Availability>
        <qValue qlevel="High">
          <Min>90 </Min>
          <Max>99 </Max>
          <Preferred>95 </Preferred>
        </qValue>
        <unit>Percentage </unit>
        <Weight>0.5 </Weight>
      </Availability>
      </FailureProbability>
      <Trustworthiness>
      <Reputation>
        <qValue qlevel="High">
          <Min>4 </Min>
          <Max>5 </Max>
          <Preferred>4.5 </Preferred>
        </qValue>
        <unit>None </unit>
        <Weight>0.3 </Weight>
      </Reputation>
      </Trustworthiness>
      <Cost>
      <ServicePrice>
        <qValue qlevel="Medium">
          <Min>30 </Min>
          <Max>60 </Max>
          <Preferred>40 </Preferred>
        </qValue>
        <unit>Pound </unit>
        <Weight>0.2 </Weight>
      </ServicePrice>
      </Cost>
    </QoSCriteria>
  </service>
</definitions>

```

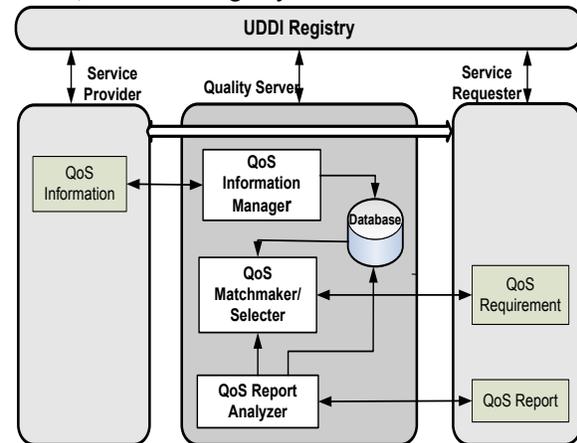
**Figure 5.** an Example of Quality Requirement in Amazon Web Service' WSDL extended with Quality Criteria Classification

#### IV. AN APPROACH FOR ENABLING UDDI WITH QUALITY CRITERIA

The Universal Description Discovery and integration (UDDI) provides a registry of businesses and Web services. UDDI describes business by their physical attributes such as name and address and the services that they provide. Business services are associated with tModels which can be associated with description standards such as WSDL. The current UDDI allow search to be carried out on limited attributes of a services such as on service name, key reference (which must be unique for a service), or based on a categoryBag (which list all the business categories within which a service is listed). Because UDDI does not represent service quality capabilities, it can't search for services on the basis of quality criteria [33].

This paper enables the current UDDI in the proposed Quality-based Web Service Architecture (QWS) to publish and discover Web services based on the proposed quality criteria classification by extending the current Web services architecture[7], [8] with quality server as shown in Figure 6.

The proposed quality-based Web Service architecture has four components: service requester, service provider, quality server, and UDDI registry.



**Figure 6.** Quality-based Web Service (QWS) Architecture

These components and their responsibilities are described below.

##### *Service Provider*

Service providers describe their services based on their functionality and quality specification, and publish the Web services based on their functionality (such as the service name, service access point, UDDI classification of the service, etc.) in the current UDDI registry. Whereas, the service providers send the quality specification of their services to the quality server in order to store them in its database. Service providers separate the service's functionality from quality specification because the current UDDI registries are not designed to accept quality specification and do not allow the requester to look for Web services based on their quality issues.

##### *Service Requester*

Service requester sends his request including both the functional requirements as well as the quality requirements to quality server and let the server to select the most suitable Web service on behalf of him. If the result is not satisfying the requester, then he/she can reduce their quality of service constraints or consider trade-offs between the desired qualities of service. After invoking the service, requester submits a quality report regarding his feeling about the service. The quality report is sent to the quality Report Analyzer for processing.

##### *UDDI Registry*

UDDI is a registry that allows the service providers to publish their services and the service requesters to look for Web services based on their functionality but not quality issues.

### Quality Server

The quality server consists of four main components; quality information manager (QIM), quality matchmaker, quality report analyzer and quality database. The quality server provides the following tasks.

- Quality server collects quality specifications about Web services provided by the service providers. By doing so, it enables the service providers to register their quality specifications.
- Quality server submits a query to UDDI registry on behalf of the requester for services' functional information such as service name, service URL, service category, etc.
- Quality server holds up-to-date information on quality specifications currently available for services.
- Matches the quality specifications against the quality requirement.
- Makes service selection decisions for requester. By doing so, quality server assists the requester to choose the best available service based on quality criteria.

The quality server components and their functions are described below.

#### Quality Information Manager (QIM)

When the service providers publish their Web services with functional description to UDDI registries, the quality information manager (QIM) collects quality specifications of the corresponding published services in the UDDI from the service providers and places it in the quality server's database. The quality specifications are required for quality matchmaking and selection.. QIM updates regularly the quality server's database whenever significant changes happen, to keep the server's information consistent and up to date with UDDI registries. QIM regularly checks the available services for new quality specification. Once an offer expires, it is deleted from the quality server database.

#### Quality Matchmaker

The quality matchmaker is the core of a quality server. Before a requester binds to Web services and begins to execute its tasks, the quality matchmaker must first determine whether the service quality desired by the user can be achieved. It discovers and selects the best available Web service on behalf of the requester. When the requester sends the service request including both the functional and quality requirements to the quality server, a quality matchmaker matches the functional requirements with the functional specification in the UDDI registry and the quality requirements with the quality specifications in the quality database. The quality matchmaking process between the quality requirements and quality specifications is out of the scope of this paper.

#### Quality Report Analyzer

After the Web service is consumed, the requester sends a quality report based on his judgments on the services to quality report analyzer, which can be subjective. The quality report includes information such as service location, invocation date, service execution duration, quality criteria offered, service rank, and comments. An example of a quality report is shown

in Table 1.

**Table 1.** Example of Quality Report

Example of Quality Report	
<b>QoS Report</b>	
Service URL	http://architag.com/WeatherInfo
Invocation Date	1/9/2000
Service Execution Duration	40msec
QoS Attributes Offered	Processing Time, Throughput, Availability
Service Rank	4
Comments	...

The quality report analyzer produces statistical information about the service and store them in the quality server's database as the historical quality information. The quality matchmaker uses this quality information for future service matching and selection

#### Quality Database

The quality Database stores the information retrieved by the quality Information Manager and quality Report Analyzer. The information stored in quality Database includes: Service functional specifications retrieved from the UDDI registry (i.e. service endpoint, URI, function name), quality specifications retrieved from the service providers (i.e. availability, service price) and statistical information of each service which produced by quality report analyzer (i.e. reputation).

The quality information stored in quality Database will be used by quality matchmaker for selecting the best candidates Web service.

## V. SELECTING THE BEST WEB SERVICE

The quality service selection in this paper is based on a mathematical model. The proposed mathematical model uses two methods in order to select the best Web service. Analytical Hierarchy Process (AHP) method is used to calculate the quality criteria weights based on the service requester's quality preferences. Euclidean distance method is used as in [34], to measure the distance between the quality requirements specified by the service requester and the quality specifications specified by the service provider. The Web service with the minimum Euclidean distance is the best service to select. The mathematical model is described in the following steps using an example.

#### Step-1: Construct pair-wise comparison matrix

The pair-wise comparison matrix  $A$ , equation (1), is constructed with respect to the service requester's quality preferences and compares them in a pair wise way. The pair-wise comparison matrix  $A$  is a reciprocal matrix representing the service requester judgements of selecting the relative importance of his preference of quality criterion  $C_i$  over  $C_j$  from Table 2. The main diagonal of the matrix is always 1. The requester specifies  $m(m-1)/2$  preferences, where  $m$  is the number of quality criteria.

$$A = \begin{bmatrix} 1 & a_{12} & \cdots & a_{1m} \\ a_{21} & 1 & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & 1 \end{bmatrix} \quad (1)$$

**Table 2** Relative Importance Measurement Scale [35]

Relative Importance Measurement Scale	
Importance Intensity	Definition
9	Extremely Preferred
8	Very strongly to extremely
7	Very strongly preferred
6	Strongly to very strongly
5	Strongly preferred
4	Moderately to strongly
3	Moderately preferred
2	Equally to moderately
1	Equally preferred

### Example:

The service requester's quality preferences are:

- Availability (AV) is assigned by the service requester as two times more important than the Reputation (REP).
- Availability (AV) is assigned by the service requester as four times more important than the Price (P).
- Reputation is the same as important as Price.

The number of quality criteria,  $m=3$ . The requester specifies 3 preferences or judgments. Thus, a comparison matrix A from the equation [1] is formed:

$$A = \begin{matrix} & \begin{matrix} AV & REP & P \end{matrix} \\ \begin{matrix} AV \\ REP \\ P \end{matrix} & \begin{bmatrix} 1 & 2 & 4 \\ 1/2 & 1 & 1 \\ 1/4 & 1 & 1 \end{bmatrix} \end{matrix}$$

### Step-2: Calculate the weight vector of quality criteria

The weights of quality criteria can be calculated from the matrix A by using equation (2).

$$w_i = \frac{1}{\text{number of criteria}} \left[ \frac{\text{preference of criterion } i \text{ in column 1}}{\text{sum of the entries in column of criteria 1}} + \frac{\text{preference of criterion } i \text{ in column 2}}{\text{sum of the entries in column of criteria 2}} + \cdots + \frac{\text{preference of criterion } i \text{ in column } m}{\text{sum of the entries in column of criteria } m} \right] \quad (2)$$

### Example:

$$W(AV) = \frac{1}{3} \left( \frac{1}{1.75} + \frac{2}{4} + \frac{4}{6} \right) = 0.579$$

$$W(REP) = \frac{1}{3} \left( \frac{0.5}{1.75} + \frac{1}{4} + \frac{1}{6} \right) = 0.234$$

$$W(P) = \frac{1}{3} \left( \frac{0.25}{1.75} + \frac{1}{4} + \frac{1}{6} \right) = 0.187$$

The weight vector is:

$$W = [0.579 \quad 0.234 \quad 0.187]$$

### Step-3: Calculate the Consistency Ratio (CR)

The Consistency Ratio (CR) measures the degree of consistency among the pair-wise judgements [36]. It can be calculated from equation (3) [37]. The Consistency Ratio (CR) of value 0.10 or less is considered acceptable and the requester judgement is consistent [35]. An acceptable consistency property helps to ensure decision-maker reliability in determining the priorities of a set of quality criteria.

$$CR = \frac{CI}{RI} \quad (3)$$

Where  $CI$  is the Consistency Index and  $RI$  is the Random Index. The  $RI$  value is selected from Table 3.

**Table 3** Average Random Index (RI) [35]

Average random index (RI)										
Size of matrix	1	2	3	4	5	6	7	8	9	10
Random index	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49

The Consistency Index ( $CI$ ) is defined as [38], [39]:

$$CI = \frac{(\lambda - m)}{(m-1)} \quad (4)$$

Where  $\lambda$  is the average of the row totals of the normalized matrix A divided by the weight vector

### Example:

The Consistency Ratio (CR) is calculated from equations (3) and (4) as in the following.

1. Random Index  $RI$  for matrix A of size 3 is equal to 0.58, as given in Table 3.

2. Calculate  $\lambda$  from the following:

- Calculate the weighted sum matrix by the following:

$$0.579 \begin{bmatrix} 1 \\ 0.5 \\ 0.25 \end{bmatrix} + 0.234 \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix} + 0.187 \begin{bmatrix} 4 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.795 \\ 0.711 \\ 0.566 \end{bmatrix}$$

- Divide all the elements of the weighted sum matrices by their respective priority vector element to obtain:

$$\frac{1.795}{0.579} = 3.1, \quad \frac{0.711}{0.234} = 3.04, \quad \frac{0.566}{0.187} = 3.02$$

- $\lambda$  can be obtained from the average of the above values:

$$\lambda = \frac{(3.1 + 3.04 + 3.02)}{3} = 3.053$$

3. Calculate the Consistency Index CI from equation (4)

$$CI = \frac{(\lambda - m)}{(m - 1)} = \frac{3.053 - 3}{3 - 1} = 0.0265$$

4. Calculate the Consistency Ratio (CR) from equation (3)

$$CR = \frac{CI}{RI} = \frac{0.0265}{0.58} = 0.046$$

The Consistency Ratio (CR) is equal to 0.046 which is less than 0.1, so the pair-wise requester's judgement is consistent and therefore the procedures will continue in order to select the best Web service.

#### Step-4: Normalize the proposed performance matrix

It is assumed that the performance matrix  $P$ , equation (5) is published by the service providers. The service providers publish their Web services with the same functional information but differ with their quality criteria values.

$$P = \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1n} \\ p_{21} & p_{22} & \dots & p_{2n} \\ \dots & \dots & \dots & \dots \\ p_{m1} & p_{m2} & \dots & p_{mn} \end{bmatrix} \quad (5)$$

Since the criteria are measured in different measurement units, the performance matrix  $P$ , equation (5), should be converted into a non-dimensional one. This could be done as each element of  $P$  is normalized by the following calculation:

$$q_{ij} = \frac{p_{ij}}{\sqrt{\sum_{k=1}^n p_{ik}^2}} \quad (6)$$

This step produces a normalized performance matrix  $Q = \{q_{ij}\}$ .

The equation (6), considers only the increasing quality criteria that is the more the value the more benefit the service requester such as Availability and Reputation and it does not consider the decreasing quality criteria that is the more the value the less benefit the requester such as Price criterion. Further investigation required to consider the decreasing quality criteria as well the increasing criteria in the mathematical model.

#### Example:

Suppose that there are three Web services ( $n=3$ ) have the same functional properties and published by different service providers, characterized by three quality criteria ( $m=3$ ):  $C_1$ =Availability,  $C_2$ =Reputation and  $C_3$ =Price. The values of the quality criteria are represented in a performance matrix  $P$  from the equation (5):

$$P = \begin{matrix} AV \\ REP \\ P \end{matrix} \begin{bmatrix} 95 & 99 & 95 \\ 4 & 3.5 & 3.5 \\ 38.37 & 30.27 & 38.38 \end{bmatrix}$$

The normalized performance matrix can be obtained from equation [8] as shown below:

$$Q = \begin{bmatrix} 0.569 & 0.593 & 0.569 \\ 0.628 & 0.550 & 0.550 \\ 0.617 & 0.487 & 0.618 \end{bmatrix}$$

#### Step-5: Construct a weighted normalized performance matrix

The normalized values are then assigned weights with respect to their importance to the requester, given by the vector  $w = \{w_1, w_2, \dots, w_m\}$ . When these weights are used in conjunction with the matrix of normalized values  $Q = \{q_{ij}\}$ , this produces the weighted normalized matrix  $V = \{v_{ij}\}$ , defined as  $V = \{w_i q_{ij}\}$ , or

$$V = \begin{bmatrix} w_1 q_{11} & w_1 q_{12} & \dots & w_1 q_{1n} \\ w_2 q_{21} & w_2 q_{22} & \dots & w_2 q_{2n} \\ \dots & \dots & \dots & \dots \\ w_m q_{m1} & w_m q_{m2} & \dots & w_m q_{mn} \end{bmatrix} \quad (7)$$

#### Example:

The weighted normalized performance matrix can be obtained from equation (7);  $V = \{w_i q_{ij}\}$ , where  $w_i$  is obtained from step-2, as shown below:

$$V = \begin{bmatrix} 0.329 & 0.343 & 0.329 \\ 0.147 & 0.129 & 0.129 \\ 0.115 & 0.091 & 0.116 \end{bmatrix}$$

#### Step-6: Calculate the relative distances

In this step each of the services is measured according to its closeness to the requester quality requirements. The relative Euclidean distances are calculated as follows:

$$E_j = \sqrt{\sum_{i=1}^m (v_{ij} - w_i r_i / \sqrt{\sum_{i=1}^m p_{ij}^2})^2} \quad (8)$$

Where  $j=1, 2, \dots, n$  is the number of Web services.

#### Example:

Suppose that requester's quality requirements are  $r = (98, 3, 40)$  for the corresponding *Availability*, *Reputation* and *Price*. The values of the relative Euclidean distances, measuring the closeness between these requirements and the available services are obtained from equation (8):

$$E_1 = 0.268, E_2 = 0.239, E_3 = 0.258$$

### Step-7: Rank services in preference order

This is done by comparison of the values calculated in Step-6. Obviously, the Web service with smallest value  $E^* = \min\{E_1, E_2, \dots, E_n\}$  gives the closest match to the requester quality requirements and should be selected as the best one.

### Example:

It is seen from the result of step-6 that the second Web service is the best one, since its Euclidean distance is smallest (0.239), compared to the distances of other services. So, the requester will select the second Web service.

If the requester's preferences are changed so that the weight vector is:

$$W = [W(AV) \ W(REQ) \ W(P)] = [0.131 \ 0.677 \ 0.192]$$

Then the Euclidean distance will be:

$$E_1 = 0.399, \ E_2 = 0.398, \ E_3 = 0.35$$

It is seen that the third Web service is the best for having the smallest Euclidean distance.

This example illustrates that the relative weight given to the quality criteria affects the final ranking of the service and depends on the requester preferences and therefore make certain quality criteria weigh more than others.

In the proposed quality-based Web service architecture (QWSA), it is considered to select more than one best service to be a more efficient approach; if one selected service failed, the others can be used instead.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a quality criteria classification that organizes web services qualities into four groups: performance, failure probability, trustworthiness and cost. The quality criteria classification is specified within the Web Service Description Language (WSDL). The WSDL extension is illustrated by extending the Amazon Web Services' WSDL with an example of quality requirement based on quality criteria classification. We demonstrate an approach that enables the Universal Description, Discovery and Integration (UDDI) to publish and discover Web services based on quality criteria classification by extending the current Web service architecture with a quality server. Quality server registers quality specifications in its database by using quality Information Manager (QIM) and enables service discovery and selection based on quality criteria by using quality Matchmaker. The quality matchmaker implements the mathematical model to select the best service.

Further research is needed to define the quality matchmaking process (QMP), which implements four algorithms: Interface matchmaking (functional matchmaking),

quality criteria type matchmaking (non-functional matchmaking), quality criteria value constraint matchmaking and mathematical matchmaking

We need to implement the quality matchmaker and the service selection process, by developing a simulation system. Also, we need to demonstrate the feasibility of the quality service selection through a case scenario.

## VII. REFERENCES

- [1] T. Bray, J. Paoli, C. M. Sperberg-McQueen, and E. Maler, "Extensible Markup Language (XML) 1.0 (Third Edition)," 4 February 2004. Available at: <http://www.w3c.org/TR/REC-xml>.
- [2] M. Gudgin, M. Hadley, N. Mendelsohn, J.-J. Moreau, and H. F. Nielsen, "SOAP Version 1.2 Part 1: Messaging Framework," 24 June 2003. Available at: <http://www.w3c.org/TR/SOAP12-part1>.
- [3] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web Services Description Language (WSDL) 1.1," March 2001. Available at: <http://www.w3.org/TR/wsdl>.
- [4] A. Manes, "Web Services Standardization: UDDI," 19 September 2003. Available at: <http://www.uddi.org/news.html>.
- [5] W3C Working Group, "Web Services Architecture," Feb. 2004.
- [6] Roberto Chinnici, Jean-Jacques Moreau, Arthur Ryman, and Sanjiva Weerawarana, "Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language " Available at: <http://www.w3.org/TR/wsdl20/> 2007.
- [7] D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris, and D. Orchard, "Web Services Architecture.," 11 February 2004. Available at: <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/wsa.pdf>.
- [8] K. Gottshalk, S. Graham, H. Kreger, and J. Snell, "Introduction to Web Services Architecture," *IBM Systems Journal*, vol. 41, 2002.
- [9] P. Brittenham and D. Ehnebuske, "Understanding WSDL in a UDDI Registry, Part1," Available at: <http://www-106.ibm.com/developerworks/library/ws-wsdl/?n-ws-9201>. Sep 2002.
- [10] F. Curbera, D. Ehnebuske, and D. Rogers, "Using WSDL in a UDDI Registry," May 21, 2002. Available at: <http://www.uddi.org/pubs/wsdlbestpractices-V1.07-Open-20020521.pdf>.
- [11] D. Gouscos, M. Kalikakis, and P. Georgiadis, "An Approach to Modeling Web Service QoS and Provision Price," in *4th International Conference on Web Information Systems Engineering Workshops (WISEW'03)*, Roma, Italy, December 13, 2003, pp. 121-130.
- [12] Z. Chen, C. Liang-Tien, B. Silverajan, and L. Bu-Sung, "UX-An Architecture Providing QoS-Aware and Federated Support for UDDI," in *Proceeding of the first International Conference on Web Services (ICWS03)*, Las Vegas, Nevada, USA, 2003.
- [13] P. Farkas and H. Charaf, "Web Services Planning Concepts," *Journal of WSCG*, vol. 11, February 2003.
- [14] C. Adams and S. Boeyen, "UDDI and WSDL Extensions for Web Services: A Security Framework," in *ACM Workshop on XML Security*, Fairfax, VA, USA, 2002.

- [15] A. Ali, O. Rana, R. Al-Ali, and D. Walker, "UDDIe: An Extended Registry for Web Services," in *Proceedings of the service oriented computing: Models, Architectures and Applications, SAINT-2003 IEEE Computer Society Press.*, Orlando Florida, USA., January 2003.
- [16] S. Ran, "A Model for Web Services Discovery With QoS," *ACM SIGecom Exchanges*, vol. 4, pp. 1-10, 2003.
- [17] R. Sumra and A. D., "Quality of Service for Web Services- Demystification, Limitations, and Best Practice," Available at: <http://www.developer.com/services/article.php/2027911>.
- [18] H.-Y. J. Y.-J. Seo, and Y.-J. Song, "A Study on Web Services Selection Method Based on the Negotiation Through Quality Broker: A MAUT-based Approach " in *First International Conference on Embedded Software and Systems (ICESSE 2004)*, Hangzhou, China, 2004.
- [19] C. Patel, K. Supekar, and Y. Lee, "A QoS Oriented Framework for Adaptive Management of Web Service Based Workflows," in *Database and Expert Systems Applications*. vol. 2736 / 2003: Springer-Verlag Heidelberg, October 2003, pp. 826 - 835.
- [20] A. Mani and A. Nagarajan, "Understanding Quality of Service for Web Services," IBM DeveloperWorks Technical Paper, January 2002.
- [21] K. Lee, J. Jeon, W. Lee, S.-H. Jeong, and S.-W. Park, "QoS for Web Services: Requirements and Possible Approaches," 25 November 2003. Available at: <http://www.w3c.or.kr/kr-office/TR/2003/ws-qos/>.
- [22] M. P. Singh, "Trustworthy Service Composition: Challenges and Research Questions," *Lecture Notes on Artificial Intelligence*, vol. 2631, pp. 39-52, 2003.
- [23] J. Zhang, "Trustworthy Web Services: Actions for Now," *IP Professional*, vol. 7, pp. 32-36, 2005.
- [24] R. Salz, "Securing Web Services," Available at: <http://webservices.xml.com/pub/a/ws/2003/01/15/ends.html>, 2003.
- [25] B. Atkinson, G. Della-Libera, S. Hada, and et al., "Web Services Security (WS-Security)," Available at: <http://www-106.ibm.com/developerworks/webservices/library/ws-secure/>, 2002.
- [26] M. C. Mont, K. Harrison, and M. Sadler, "The HP time vault service: exploiting IBE for timed release of confidential information," in *the 12th international conference on World Wide Web*, Budapest, 2003, pp. 160 - 169.
- [27] L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, and Q. Sheng, "Quality Driven Web services Composition," in *Proceedings of the Twelfth International World Wide Web Conference (WWW'2003)*, Budapest, Hungary, May 2003.
- [28] Y. Liu, A. H. Ngu, and L. Z. Zeng, "QoS computation and policing in dynamic web service selection," in *International World Wide Web Conference*, New York, NY, USA, 2004.
- [29] S. G., J. A. Miller, A. P. Sheth, A. Maduko, and R. Jafri, "Modeling and Simulation of Quality of Service for Composite Web Services ", in *Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics (SCI'03)*, Orlando, Florida, (July 2003).
- [30] S. Andreozzi, D. Montesi, and R. Moretti, "Web Services Quality," in *Conference on Computer, Communication and Control Technologies (CCCT03)*, Orlando, 31 July - 2 August 2003.
- [31] C. Marchetti, B. Pernici, and P. Plebani, "A Quality Model for e-Service Based Multi-Channel Adaptive Information Systems," in *4th International Conference on Web Information Systems Engineering Workshops (WISEW'03)*, Roma, Italy, December 13, 2003, pp. 165-172.
- [32] "Amazon Web Services," Available at <http://amazon.com/webservices>.
- [33] M. Paolucci, T. Kawamura, T. R. Payne, and K. Sycara, "Importing the Semantic Web in UDDI," in *Proceedings of Web Services, E-Business and Semantic Web Workshop, CAiSE 2002.*, Toronto, Canada, 2002, pp. 225-236.
- [34] L. Taher, H. El Khatib, and R. Basha, "A Framework and QoS Matchmaking Algorithm for Dynamic Web Services Selection," in *Second International Conference on Innovations in Information Technology (IIT'05)* Dubai, UAE, 2005.
- [35] T. L. Saaty, "How to make a decision: The Analytic Hierarchy Process," *European Journal of Operational Research*, vol. 48, pp. 9-26, 1990.
- [36] H. Ye, B. Kerherve, and G. V. Bochmann, "QoS-based Distributed Query Processing," *Ingénierie des Systèmes d'Information (RSTI série ISI)*, vol. 9, 2004.
- [37] M. Hajeesh and A. Al-Othman, "Application of the analytical hierarchy process in the selection of desalination plants," *Desalination*, vol. 174, pp. 97-108, 2005.
- [38] L. Taher, R. Basha, and H. El Khatib, "Establishing Association between QoS Properties in Service Oriented Architecture," in *Proceedings of the IEEE International Conference on Next Generation Web Services Practices (NWeSP'05)*, 2005.
- [39] L. Taher, H. Khatib, and R. Basha, "A Framework and QoS Matchmaking Algorithm for Dynamic Web Services Selection," in *The Second International Conference on Innovations in Information Technology (IIT'05)*, 2005.