Objective and Subjective QoS Factors Supported Web Service Search Method Based on Extended WSDL

Ling Jiang[1], Tingna Liu[1]*, Desheng Liu[1,2]
School of Resources and Environment,
[1]University of Electronic Science and Technology of China, Chengdu, China
[2]Department of Geography, the Ohio State University, Columbus, Ohio, USA
*Corresponding author, e-mail:liutingna86@163.com

Abstract—**With the rapid development of Web Services applications and the increasing popularity, there are numerous functionally similar Web Services available, which made the selection of the Web Services based on functional properties is insufficient. QoS is becoming an important criterion in selection for the best Web Services. In this paper, an objective and subjective QoS factors supported Web Service search method was proposed. The Objective-QoS factors were added by extending the WSDL documents, which should be offered by the service provider. The Subjective-QoS factors comes from users' perspective, represent the actual experiments of the service users. A prototype system was developed to implement both of Objective-QoS and Subjective-QoS factors.**

Keywords-Web Service Search; Objective QoS; Subjective QoS; User's Feedback; Lucene

I. INTRODUCTION

Web Services are self-contained, self-describing, modular applications that can be published, located and invoked across the Internet. Since most Web Services are based on the standard criterions such as HTTP or XML (Extensible Markup Language), they are inherently capable of application of cross-platform over the Internet. At the same time, the Web Services have a set of open specifications, such as XML, SOAP (Simple Object Access Protocol), UDDI (Universal Description Discovery and Integration)[1], and WSDL (Web Service Definition Language)[2], which made them can be released, search, composition, and invocation through the network. Web Services are so important that the academia and technology vendors such as IBM and Microsoft have focused their eyes on.

The first thing to do before applying the Web Services to Web applications is to find them. However, due to rapid development of the Web Services, there are numerous functionally similar Web Services available, which made the selection of the Web Services based on functional properties is insufficient. QoS (Quality of Service) is becoming an important criterion in selection for the best Web Services. Unfortunately, the WSDL only aims at the functional aspects of a Web Service, without containing any description of non-functional or QoS characteristics. UDDI registries also do not have the ability to publish the QoS information, and the authenticity of the advertised QoS information available elsewhere may be questionable. At present, the research on this problem focuses on two topics: (1) based on the extension of UDDI [3] or WSDL [4], and (2) based on the participation of the third party, such as a broker [5]. However, with the successfully shutdown of the main commercial UDDI service, which supported by Microsoft, IBM, SAP, etc., the approach by extending UDDI do not seems an optimistic future. Another approach based on third party brokers proposed a new design concept. However, introducing brokers may bring some commercial drawbacks, such as injustice and malicious ratings.

In this paper, we present an Objective and Subjective QoS factors supported Web Service search method based on extended WSDL. This discovery method is a full text search engine, which take advantage of the structured information of WSDL extracted by wsdl4j [6]. Furthermore, in order to improve the performance of this discovery method, we adopt a segmentation processing and WordNet synonyms libraries proposed by [7]. Lucene [8] was used to implement the full text search engine.

The remainder of the paper is organized as follows. Section 2 presents the extension method of the Objective-QoS in WSDL. Section 3 introduces the Subjective-QoS structure. Section 4 describes the implementations of the proposed method and shows the prototype system of Web Service search. Section 5 concludes the paper.

II. Extension of Objective-QoS in WSDL

The traditional WSDL documents does not describe QoS information, which only meet the functional requirements of Web Service. Therefore, the insufficient quality of service information cannot meet the service requestors' needs, which make it difficult to select higher quality service among a large number of functionally similarity Web services.

A. The Extension of Objective-QoS

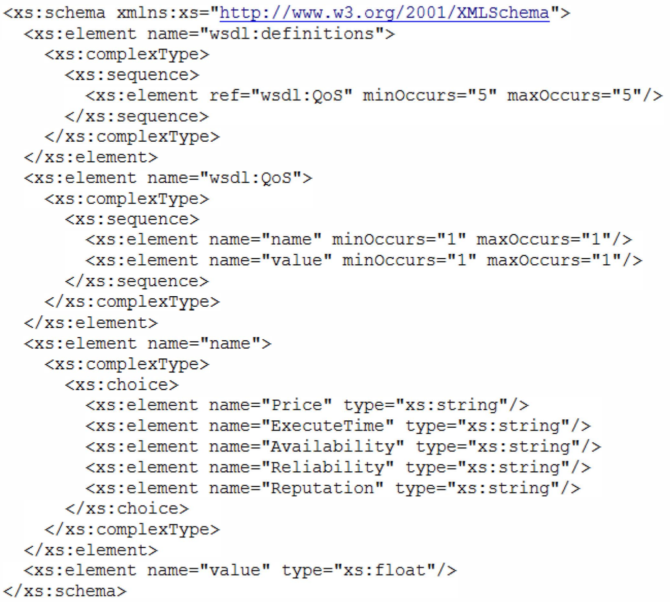
There are two methods of how to add QoS information in the Web Service description: the one is to extend the existing WSDL protocols, and the other is based on a new protocol could be support the QoS attributes. Considering the current WSDL is widely used in Web Service description, and new protocols do

Supported by the National Natural Science Foundation of China (Grant No 41201388), the Open Research Fund of State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing.

not achieve a uniform standard, we adopt the first method in this paper by extending WSDL to describe QoS information.

We consider some generic Objective-QoS criteria for Web Service, such as Web Service price, execution time, availability, reliability, and reputation, etc. (1) Price is the fee that a service requester has to pay for invoking the Web Service. Apparently, service requester tend to apply a better price/performance ratio services. (2) Execution time measures the expected delay in seconds between the moment when a request is sent and the moment when the results are received. This QoS attribute may be related to some unpredicted factors, like the network condition and the amount of service requesters. We generally use an average value of this QoS factor. (3) Availability reacts the probability that the service is accessible. (4) Reliability represents the ability to successfully handle the request and deliver a correct response of a Web Service. (5) Reputation refer to the trustworthiness of a Web Service.

Since WSDL is a XML based language, we used XML Schema to define the type of extended WSDL document (see Fig. 1).



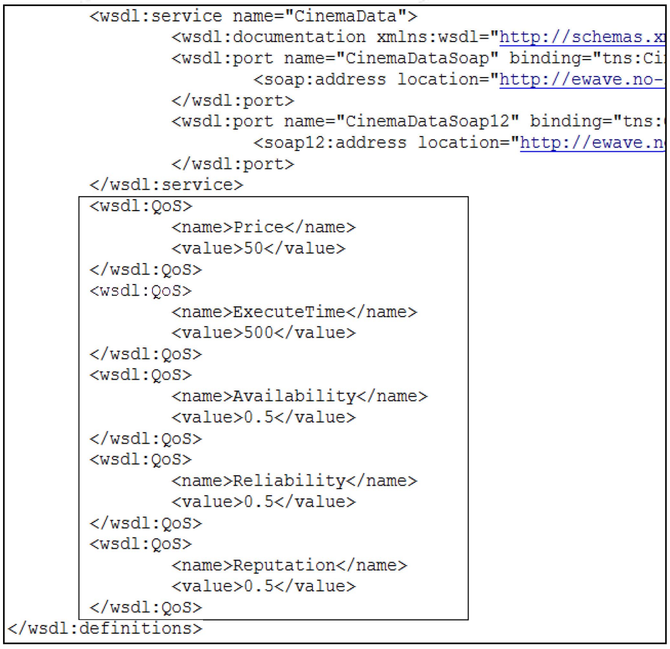
```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="wsdl:definitions">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="wsdl:QoS" minOccurs="5" maxOccurs="5"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="wsdl:QoS">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="name" minOccurs="1" maxOccurs="1"/>
        <xs:element name="value" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="name">
    <xs:complexType>
      <xs:choice>
        <xs:element name="Price" type="xs:string"/>
        <xs:element name="ExecuteTime" type="xs:string"/>
        <xs:element name="Availability" type="xs:string"/>
        <xs:element name="Reliability" type="xs:string"/>
        <xs:element name="Reputation" type="xs:string"/>
      </xs:choice>
    </xs:complexType>
  </xs:element>
  <xs:element name="value" type="xs:float"/>
</xs:schema>
```

Figure 1. WSDL definition of Objective-QoS

The Schema provide a list of Objective-QoS elements and attributes associated with particular data types. We use this Schema to constrain where and how many elements and attributes can appear, and what can appear inside those elements. An example of extended WSDL with Objective-QoS as seen in Fig. 2.

B. *The Extraction of Objective-QoS*

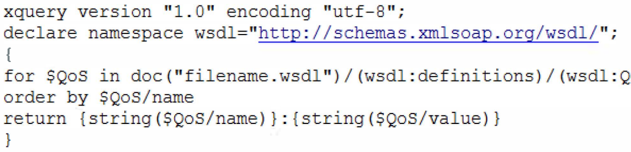
The extendibility of WSDL enables the Web Service provider could add additional QoS attributes whenever they need, with considering that QoS delivered to users may be affected by many factors——the performance of the Web Service itself, the hosting platform, and the network condition. These Subjective-QoS attributes needed to be extracted from the extended WSDL document and added to the index, to enable the Web Service search method could be supported with the Subjective-QoS attributes.



```
        <wsdl:service name="CinemaData">
                <wsdl:documentation xmlns:wsdl="http://schemas.x
                <wsdl:port name="CinemaDataSoap" binding="tns:Ci
                        <soap:address location="http://ewave.no-
                </wsdl:port>
                <wsdl:port name="CinemaDataSoap12" binding="tns:
                        <soap12:address location="http://ewave.n
                </wsdl:port>
        </wsdl:service>
        <wsdl:QoS>
                <name>Price</name>
                <value>50</value>
        </wsdl:QoS>
        <wsdl:QoS>
                <name>ExecuteTime</name>
                <value>500</value>
        </wsdl:QoS>
        <wsdl:QoS>
                <name>Availability</name>
                <value>0.5</value>
        </wsdl:QoS>
        <wsdl:QoS>
                <name>Reliability</name>
                <value>0.5</value>
        </wsdl:QoS>
        <wsdl:QoS>
                <name>Reputation</name>
                <value>0.5</value>
        </wsdl:QoS>
</wsdl:definitions>
```

Figure 2. Example of an extended WSDL document with Objective-QoS

We use XQuery language to extract QoS attributes in the extended WSDL. The XQuery language is as follows:



```
xquery version "1.0" encoding "utf-8";
declare namespace wsdl="http://schemas.xmlsoap.org/wsdl/";
{
for $QoS in doc("filename.wsdl")/(wsdl:definitions)/(wsdl:QoS)
order by $QoS/name
return {string($QoS/name)}:{string($QoS/value)}
}
```

Figure 3. XQuery expression for extraction of Objective-QoS

Besides considering different users may have different QoS requirement, our method support multiple QoS composition request as well as single QoS request in service discovery. For example, some service requestors need restriction of the service price, while others need both the price and execution time restriction. A matching method based on the multiple criteria QoS requirements as well as functional capability has been proposed. We use a filter mechanism to select the wanted Subjective-QoS attributes.

III. IMPLEMENT OF SUBJECTIVE-QOS

A. *The Structure of Subjective-QoS*

The most of current research emphasizes the Objective-QoS factors, which are offered by service provider. Objective-QoS information published by the service providers may not always be accurate and up-to-date. Furthermore, some Web Services providers do not guarantee or assure the level of QoS offered by their Web Services, which result in bad experience in selection of the best of Web Service based only on the Objective-QoS. Meanwhile, these QoS factors cannot respond the feelings of users who actually use the service.

To solve above problems, a mechanism and a model were proposed by introducing QoS factors from users' perspective. We call the QoS that comes from users' perspective as Subjective-QoS in this paper. The so-called Subjective-QoS factors are comparative QoS of the Objective-QoS factors, which refer to real experiences of the users who invoked the services. Firstly, users can rate Web Services they applied with various Subjective-QoS factors. Multiple evaluation indicators were selected as Subjective-QoS attributes to response the feelings of actually service users, such as usability, stability, and recommendation level. These ratings can be provided to new users to help them to rank Web Services in selection. Secondly, in order to prevent users' malicious ratings and made the ratings visualized to other user as well, a message board system has been utilized, in which users can express their opinions and suggestions. Fig. 4 illustrates a general framework of the implementation of the Subjective-QoS factors.

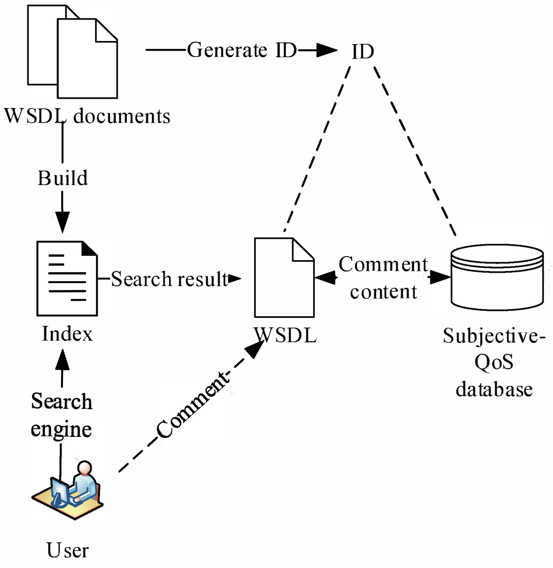


Figure 4. Framework of the implementation of Subjective-QoS

B. *The Subjective-QoS Evaluation Model*

Subjective-QoS evaluation is a multi-attribute comprehensive evaluation method. Multi-attribute comprehensive evaluation method is a process by synthesis of multiple evaluation values to a holistic comprehensive evaluation values with a sophisticated mathematical model. The Subjective-QoS evaluation process as shown in Fig. 5.

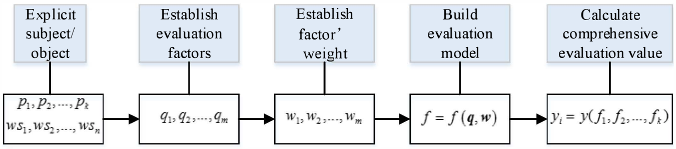

Explicit subject/ object
Establish evaluation factors
Establish factor' weight
Build evaluation model
Calculate comprehensive evaluation value
$p_1, p_2, \ldots, p_k$
$ws_1, ws_2, \ldots, ws_n$
$q_1, q_2, \ldots, q_m$
$w_1, w_2, \ldots, w_m$
$f = f(\boldsymbol{q}, \boldsymbol{w})$
$y_i = y(f_1, f_2, \ldots, f_k)$

Figure 5. Subjective-QoS evaluation process

The actual Web Service users are the subject in the evaluation process that can be written as $\boldsymbol{p} = (p_1, p_2, \ldots, p_k)$, which represented there are k users in the evaluation of a Web Service. While the Web Services are the object that be written as $\boldsymbol{ws} = (ws_1, ws_2, \ldots, ws_n)$, which represented there are n Web Services could be evaluated.

The vector $\boldsymbol{q} = (q_1, q_2, \ldots, q_m)$ shows the evaluation factors, and the vector $\boldsymbol{w} = (w_1, w_2, \ldots, w_m)$ indicate the weight of every evaluation factor. We use three evaluation factors in our experimental that is the overall satisfaction, service reliability and service availability, as in (1). We adopt a simple linear weighting method as the evaluation model (as in formula (3)) with a simple weight-distributed (as in formula (2)) to the weight vector.

$$\boldsymbol{q} = (q_1, q_2, \ldots, q_m) = (q_{total}, q_{rel}, q_{ava}) \qquad (1)$$

$$\boldsymbol{w} = (w_1, w_2, \ldots, w_m) = (0.4, 0.3, 0.3) \qquad (2)$$

$$f(\boldsymbol{q}, \boldsymbol{w}) = \boldsymbol{q}\boldsymbol{w}^T = (q\ q\ \ ,\ .q\ .\ w\ )(\ \ ,w\ ^T, \qquad (3)$$

We assume users are equally important in the process of calculate the comprehensive evaluation value, so we take an average score as the final value seen as in formula (4).

$$y_i = y(\ f,\ f,\ \ldots\ _kf = \frac{1}{k}(\ \ _1f+\ \ _2f+\ \ +.) \qquad (4)$$

where y_i is the value of the Subjective-QoS evaluation of ws_i, which could be stored into the Subjective-QoS database. The Subjective-QoS values can be used as reference for the later service requestors.

IV. Architecture of Prototype System

The prototype system was developed in a lab environment. We choice the Java as development language, and use Tomcat as a Web application server. Experimental data comes from Dr. Yilei Zhang's research [9], the research data provide 3738 WSDL file [10]. The architecture of the prototype system and experiment results as seen in Fig. 6.

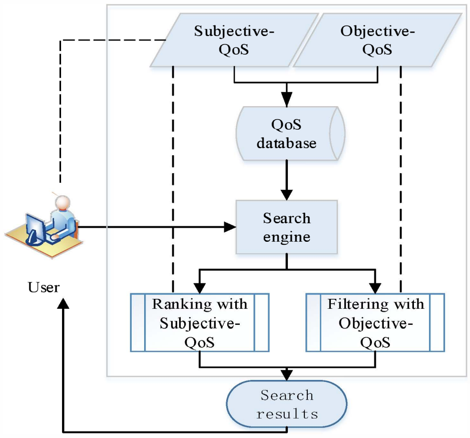


Figure 6. Architecture of prototype system

A series of experiments has been done to verify the effectiveness of the filtering ability with Objective-QoS. We adopt five Objective-QoS factors as seen in table 1. The Price, Time, Ava, Rel, and Rep in the table header represent service price, execution time, service availability, reliability, and reputation, respectively. We choice a range of percentage to filter the unqualified services. The symbols of × and √ represent if use the particularly Objective-QoS factor or not, so there should be six groups of experiments.

TABLE 1 OBJECTIVE-QOS FACTORS RESTRICTION

Exp.X	Price<80%	Time<80%	Ava>30%	Rel>30%	Rep>30%
Exp.1	×	×	×	×	×
Exp.2	√	×	×	×	×
Exp.3	√	√	×	×	×
Exp.4	√	√	√	×	×
Exp.5	√	√	√	√	×
Exp.6	√	√	√	√	√

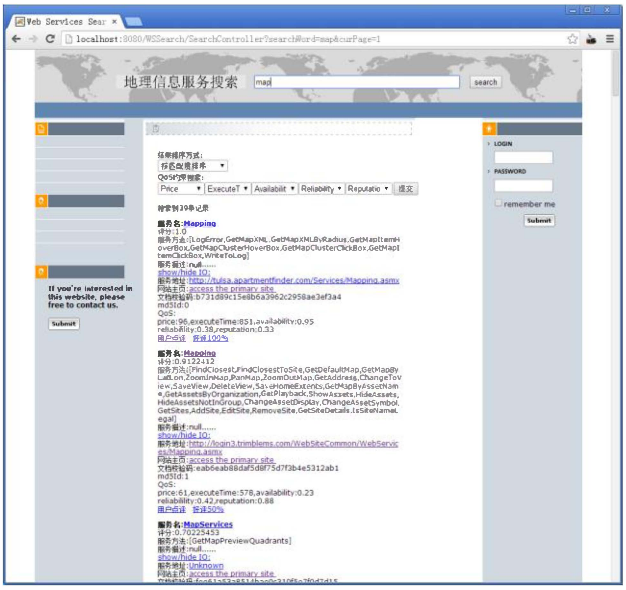
Four keywords (state, region, city location, zip) have been selected to search Web Services in our prototype system for illustrating the filter effectiveness with Objective-QoS restriction. Generally, the lower the service price and execution time the better, and the higher of the service availability, reliability, and the reputation the better. The numbers of results will be down with the more restrictions were adopted, which can be seen apparently in table 2, and the user interface was show in Fig. 7.

TABLE 2 NUMBERS OF RESULTS WITH OBJECTIVE-QOS FACTORS RESTRICTION

	state	region	city location	zip
Exp.1	189	70	73	47
Exp.2	156	57	61	36
Exp.3	124	52	47	28
Exp.4	85	37	35	18
Exp.5	58	24	26	15
Exp.6	40	17	17	6

V. CONCLUSION

In this paper, we present a Web Service search method by introducing the Objective-QoS and Subjective-QoS. The Objective-QoS requirements are used to meet the Web Service user's need by filtering the unqualified service, while the Subjective-QoS can be used as references for the later users. Experimental results show that our method is not only efficient, but also works well for complicated scenarios. The QoS supported Web Service discovery method possesses high discovery accuracy as well as an acceptable response time. Meanwhile, the system also has some problems, such as it does not guarantee the validity of the search results with lots of dead links of the Web Services. We will pay attention to the validity of Web services and minimize the dead Web Services.


Web Services Sear ×
localhost:8080/WSSearch/SearchController?searchWord=map&curPage=1
地理信息服务搜索
map
search
LOGIN
PASSWORD
remember me
Submit
If you're interested in this website, please free to contact us.
Submit

结果排序方式:
按匹配度排序
QoS约束检索:
Price ▾ ExecuteT ▾ Availabilit ▾ Reliability ▾ Reputatio ▾ 提交

检索到39条记录

服务名:Mapping
评分:1.0
服务方法:[LogError,GetMapXML,GetMapXMLByRadius,GetMapItemHoverBox,GetMapClusterHoverBox,GetMapClusterClickBox,GetMapItemClickBox,WriteToLog]
服务描述:null......
show/hide IO:
服务地址:http://tulsa.apartmentfinder.com/Services/Mapping.asmx
网站主页:access the primary site
文档校验码:b731d89c15e8b6a3962c2958ae3ef3a4
md5Id:0
QoS:
price:96,executeTime:851,availability:0.95
reliability:0.38,reputation:0.33
用户点评 好评100%

服务名:Mapping
评分:0.9122412
服务方法:[FindClosest,FindClosestToSite,GetDefaultMap,GetMapByLatLon,ZoomInMap,PanMap,ZoomOutMap,GetAddress,ChangeToView,SaveView,DeleteView,SaveHomeExtents,GetMapByAssetName,GetAssetsByOrganization,GetPlayback,ShowAssets,HideAssets,HideAssetsNotInGroup,ChangeAssetDisplay,ChangeAssetSymbol,GetSites,AddSite,EditSite,RemoveSite,GetSiteDetails,IsSiteNameLegal]
服务描述:null......
show/hide IO:
服务地址:http://login3.trimblems.com/WebSiteCommon/WebServices/Mapping.asmx
网站主页:access the primary site
文档校验码:eab6eab88daf5d8f75d7f3b4e5312ab1
md5Id:1
QoS:
price:61,executeTime:578,availability:0.23
relialibility:0.42,reputation:0.88
用户点评 好评50%

服务名:MapServices
评分:0.70225453
服务方法:[GetMapPreviewQuadrants]
服务描述:null......
show/hide IO:
服务地址:Unknown
网站主页:access the primary site

Figure 7. User Interface

ACKNOWLEDGMENT

The work conducted by Ling Jiang was partially supported by the National Natural Science Foundation of China (Grant No. 41201388), the Open Research Fund of State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing.

REFERENCES

[1] "Online community for the Universal Description, Discovery, and Integration", http://uddi.xml.org/, 2015.

[2] E. Christensen, F. Curbera, "Web Services Description Language", W3CNote, http://www.w3.org/TR/wsdl, 2001.

[3] A. M. Karande, D. R. Kalbande, "Web service selection based on QoS using tModel working on feed forward network", Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference on. IEEE, 2014, pp. 29-33.

[4] X. Liu, W. Gui, H. Wang, Y. Liu, L. Jun, "A new nynamic QoS management model base on DQ-WSDL for Web Services", 2013.

[5] M. A. Serhani, R. Dssouli, A. Hafid, H. Sahraoui, "A QoS broker based architecture for efficient web services selection. Web Services", ICWS 2005. Proceedings of 2005 IEEE International Conference on. IEEE, 2005, pp. 113-120.

[6] WSDL4J. http://wsdl4j.sourceforge.net/, 2015.

[7] T. Liu, L. Jiang, "A Web Service discovery method based on data segmentation and WordNet", Proceedings of the 4th International Conference on Computer Engineering and Networks, Springer International Publishing, 2015, pp. 797-804.

[8] Lucene, http://lucene.apache.org/, 2015.

[9] Y. Zhang, Z. Zheng, M. R. Lyu, "Wsexpress: A qos-aware search engine for web services", Web Services (ICWS), 2010 IEEE International Conference on. IEEE, 2010, pp. 91-98.

[10] WS-DREAM, http://www.wsdream.net/dataset.html, 2015.