Selection of Paramount Web Service using Sentient QOS Parameters

J. Ghayathri
Associate Professor,
Department of Computer Science (PG),
Kongu Arts and Science College,
Erode, Tamil Nadu, India

S. Pannirselvam, Ph.D
Head & Associate Professor
Department of Computer Science,
Erode Arts and Science College,
Erode, Tamil Nadu, India

ABSTRACT
Web services are reusable software components available across the web. These web services can be utilized by the consumers to fulfill their requirements. This paper presents the research about selection of best favorable web service among the number of available services. The relative web services are selected using service name and then they are refined through some of the QoS parameters like response time, performance etc. Finally the paramount service is selected by considering the user preferences.

Keywords
QoS, Refining, WSDL, SOAP

1. INTRODUCTION
Today web services being deployed are distributed process that process XML (eXtensible Markup Language) encoded SOAP (Simple Object Access Protocol) messages sent over HTTP(Hypertext Transfer Protocol) and described using WSDL(Web Services Description Language).

As the web services are loosely coupled software components, they are published, located, and invoked across the web. A web service comprises of several operations like web service creation, publishing, discovering, locating, and passing messages. Each operation in the architecture of web services takes a SOAP package containing a list of input parameters, fulfills a certain task, and returns the result in an output SOAP package. Large enterprises are increasingly relying on web services as methodology for large-scale software development and sharing of services within and outside the organization. Today many applications are being built by piecing together web services published by third-party producers. The growing number of web services available within an organization and on the web raises anew and challenging search problem that locating desired web services. The rapid growth of web services in all areas makes the user difficult to select the right required service.

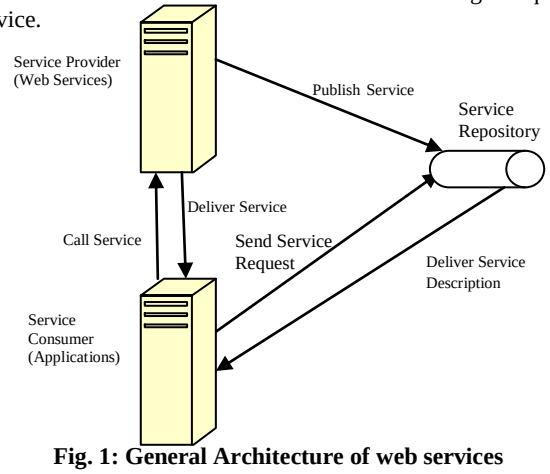

Fig. 1: General Architecture of web services

2. RELATED WORK
eFlow discusses the possibility of performing dynamic web service selection based on user requirements. In eFlow each service node contains a search recipe which defines the service selection rule to select a specific service for this node. No QoS (Quality of Service) model is defined in eFlow.[1] WebQ every task node has a separate set of QoS management rules for service selection. Both follows the selection rule based on local criteria [2]. Usually linear programming technique which is too complex for runtime is used to solve the service selection problem. QoS guarantee for web services is one of the main concerns of SLA framework. Much work has been done in the domain of web service discovery, which mainly focuses on functional properties of web services. However, in view of large number of services with comparable functionalities, web service discovery alone is inadequate for selecting optimal service that would satisfy users' expectations [9]. In web service selection model for selecting best web service based on QoS constraints the QoS attributes of web services can be stored in a database [5]. Normalization process could be used to compute QoS parameter values [7]. Combination of multiple parameters QoS could be used in preference selection model [6].

3. METHODOLOGY
Web service selection refers to the process by which a service implementation is chosen from numerous services discovered in response to requester's functional requirement. Services are selected to execute a business process based on the maximization of utilities which satisfies the users' required quality of services. Several QoS parameters such as latency throughput, reliability, availability, cost, accessibility, integrity, performance, regularity, accuracy, capacity, execution time, reputation, transmission cost etc. have been defined. This proposed work selects the most favorable service through the following processes.

a) *Service tracking*: Discovers the related services

b) *Service Refining*: Required services are refined using QoS parameters set by the users.

c) *Service Scoring*: Grading the services based on the QoS values.

d) *Service Selecting*: Selecting the most favorable service

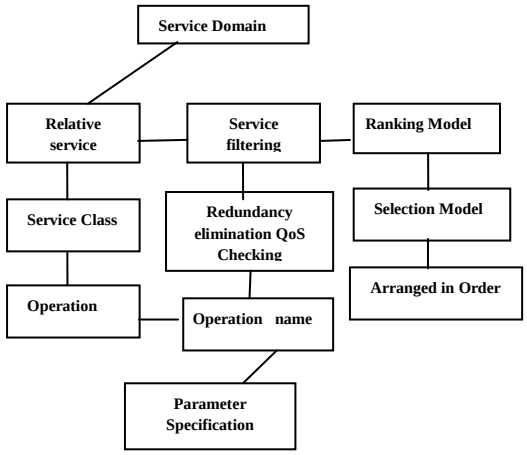

Fig.2: Frame Work of the proposed System

3.1 Service Tracking

Service selection process requires service discovery as its preliminary work; however, service selection is a core issue that must be addressed in order to retrieve appropriate service for a requester. Functional and Non-Functional properties especially QoS are the two main classes of requirements that are considered in selecting optimal service for a requester. As the web service discovery alone is inadequate for selecting favorable service that would satisfy users' expectations, the efficient methodologies and procedures are required for appropriate web service selection, which is the main concern in the domain of service oriented computing.

A set of related services have been selected from the service repository. The service repository holds the WSDL files of the service providers. The matching service could be processed through service name and operation of WSDL where service name specifies the particular service and the operation specifies the method that can be utilized by the consumer. Discovering the service only might end up in irrelevant service selection. So to make the selection process much efficient the operations are also matched with the requirement and based on that the set of relative services are selected. For instance a service – Travel Service is first searched through the service name and operation. The structure of a WSDL file for a Travel Service is given below.

```
<wsdl:operation name="GetAllCountries">
    <soap12:operation soapAction="http://www.kompletnet.hr/GetAllCountries"
style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
<wsdl:service name="TravelService">
  <wsdl:port name="TravelServiceSoap" binding="tns:TravelServiceSoap">
    <soap:address
location="http://puturist.com/Admin/Travel/WebServices/Travel/TravelService.as
mx" />
  </wsdl:port>

<wsdl:port name="TravelServiceSoap12" binding="tns:TravelServiceSoap12">
    <soap12:address
location="http://puturist.com/Admin/Travel/WebServices/Travel/TravelService.as
mx" />
  </wsdl:port>
 </wsdl:service>
```

The Algorithm given in table 3.1 is used to find the set of relative services R.

Table 3.1 Algorithm for Service Tracking

```
Procedure ServiceTracking(Repository,R)
Step 1: S=sevice name ; O= operation; i=0;    // web
                    service name and method//
Step 2: Open WSDL file(W, Repository);  //open
                                repository //
Step 3:
      While service_repository != eor do // till the
                                end of repository//
       {
            i=i+1;
           If service name(W_i )= S
           If operation(W_i) =O
           R_i ←W_i;
       }
       return
```

3.2 Service Refining

This process refines the selected set of related services by applying QoS parameters of web services. The QoS parameters considered in this system are response time, availability, performance and reliability. Response Time is the performance factor of a web service which explains how fast a service request can be completed. It is measured in millisecond. Reliability is the quality aspect of a web service which represents the degree of being capable of maintaining the service and service quality. It is measured in percentage. Availability is the absence of service down-time. It signifies the probability that the service is up and ready for immediate consumption. It is measured in percentage. Performance is the capability of the web service to serve the client's requests. It is measured in percentage [9]. For doing the refinement some decisive factors may be fixed for QoS parameters by the users. The services which pass through the refinement constraints form a set of refined services. The algorithm given in table 3.2 is used to refine the services from the set of relative services. Using the QoS constraints required by the consumer and the threshold values for the unknown values are considered as the basic parameters for service refining.

3.3 Service Scoring

Query submitted by the user returns the selected list of URL/URI of the services from the WSDL located in the service repository. These services are refined through the QoS constraints which in turn return a set of refined services. For the set of refined services the scoring is calculated by summing up the values of the QoS.

3.4 Service Selection

The most favorable service is selected based on the total score of each service. The grading for the web services is done using the normalization of QoS values of the services and the consumers' requirement. The consumers' requirement is considered in the form of min-max constraints of the non functional parameter. Because among the QoS parameters to choose the best service the response time must be minimum.

Table 3.2: Algorithm for Service Refining

```
Procedure Service Refining(R,S)
 //Input//
        // this is output from  Table 3.1
        A set of n candidates s(f) = (s1,s2,s3,…,sn)
that
        each fulfills requester's functional
        requirements;
Step 1: A set of thresholds (default constraints)
for
        desired services having n elements
        t = <t1, t2, ..tm, >;
          // where t1, t2, .. tm = response time,
            availability, performance ,reliability
            etc.   values.
Step 2: A set of constraints for desired services
        having n elements c = <c1, c2, .. cm>;
        //where   c1, c2, .., cm = response time,
         reliability, availability, performance etc.
         values
 //output //
          //Select a best service sp ∈ sf that fulfills
           requester's functional and nonfunctional
           requirements
Step 3 : Initialization:  Set threshold data
          // Read QoS constraints
Step 4: Consumer enters constraints requirements
Step 5: Refining: Compare each service's QoS
          value with consumer's constraints
Step 6: Calculate total number of services (n)
Step 7: While i < = n do
              for j = 1 to m
                if qij(si) < cj then refines out service
si                                  // (refine current web
service )
                 endif
               endfor
             endwhile
```

Whereas the other three parameters should have the maximum value. This grading improves the efficiency of further reference of services. The algorithm Grading given in table 3.3 is used to select the best service among the refined services.

Table 3.3: Algorithm for Service Grading

```
Procedure Grading(S)
Step 1: Compute normalized QoS data for each
          filtered  service
Step 2: Compute weighted values for the
           constraints  based on  requester's
           requirement
Step 3: Compute product of weight and
           normalized QoS values for each service and
           get total scores for each service
Step 4: Select service with maximum score as best
           service
```

The Service Selection algorithm depicted in the table 3.4 is used to select the best service from the available service repository.

3.4.1 Normalization of QoS Parameters

In the normalization process, Equation 3.4.1.1 is used for reliability, availability and performance parameters that require maximization whereas Equation 3.4.1.2 is used for response time that requires minimization [9].

$$qp = (q - qmin)/(qmax - qmin)$$

$$qn = (qmax - q)/(qmax - qmin)$$

qp=(q-qmin)/(qmax-qmin) ----- (Eqn. 3.4.1.1)

qn =(qmax-q)/(qmax-qmin) ----- (Eqn. 3.4.1.2)

qp, qn represent normalized value for positively and negatively inclined QoS parameter respectively, qmax and qmin represent the maximum and minimum QoS values for a set of QoS parameters and q is the QoS value of the parameter being considered.

The QoS values for the constraints are normalized using the following formula:

$q = (qc - y)/(qmax - y)$) -- (Eqn.3.4.1.1)

q' = (qmax − qc)/(qmax − y)-- Eqn.3.4.1.2)

$qthreshold != y, 1\ if\ qthreshold = y$

$q = (qc - y)/(qmax - y)$ -- (Eqn.3.4.1.3)

$q' = (qmax - qc)/(qmax - y)$ -- (Eqn.3.4.1.4)

qthreshold != y, 1 if qthreshold = y

where qc represents the parameter to be considered as QoS value, y is the default threshold value for the QoS parameter. q and q' gives normalized value for gradient positive and negative parameter respectively.

Table 3.4 Algorithm for Service Selection

```
Service_Selection()

Step 1: Call  ServiceTracking(Repository,R)
          // Find relative services
Step 2: Consider constraint values of QoS
           parameters of Consumers
Step 3: Call ServiceRefining(R,S)
          // Get the refined services of using the
           given constraints
Step 4: Calculate normalized values  of the QoS
           parameters of refined services of each
            customers using
                Eqn. 3.4.1.1 and Eqn. 3.4.1.2
Step 5: Calculate normalized values of constraint
           values of QoS parameters by using
            Eqn. 3.4.1.3 and  Eqn. 3.4.1.4
Step 6: For each consumer the weighted
          normalized values are calculated by using
             wq = qp-q
             wq' = qn-q'
Step 7:Call ServiceGrading(S)
```

4. EXPERIMENTSAND RESULTS

There are three consumers and ten services considered for the experiment. The constraints of the consumers are assigned and the algorithm and formulae is used to calculate maximum and minimum values of the Qos Parameters.

Table 4.1: Constraint Table of Consumers A, B, C

Consumer	RT	Avail.	Per.	Relia.
A	500	90	75	60
B	250	60	80	0
C	800	80	70	65

Table 4.2: Values of QoS Parameters of relative Services

Service	RT	Avail.	Per.	Relia.
S1	105.00	80	55	62
S2	320.50	95	78	60
S3	780.81	93	80	88
S4	520.11	87	68	75
S5	536.50	72	79	66
S6	247.00	99	100	72
S7	73.00	70	96	82
S8	525.12	67	60	78
S9	709.40	87	75	73
S10	147.44	94	97	60

Table 4.3: Weighted Sum of Normalized Refined Services for Consumer A

Service	RT	Avail.	Per.	Relia.	Score
S2	0.0000	0.0810	0.0358	0.0000	0.1168
S6	0.0000	0.0964	0.1202	0.0327	0.2493
S10	0.2643	0.0772	0.1087	0.0000	**0.4502**

Table 4.4: Weighted Sum of Normalized Refined Services for Consumer B

Service	RT	Avail.	Per.	Relia.	Score
S6	0.0000	0.1952	0.1739	0.0642	0.4333
S7	0.1971	0.0401	0.1525	0.1176	**0.5075**
S10	0.0000	0.1685	0.1579	0.0000	0.3263

Table 4.5: Weighted Sum of Normalized Refined Services for Consumer C

Service	RT	Avail.	Per.	Relia.	Score
S3	0.0000	0.0000	0.0000	0.0026	**0.0026**
S6	**0.7531**	0.0000	0.0128	0.0000	0.7659
S9	0.1007	0.0000	0.0000	0.0000	0.1007

Table 4.6: List of Paramount Services of Customer A, B, C

Consumer	Service
A	S10
B	S7
C	S3

As per the weighted normalized QoS the table 4.6 gives the services S10, S7 and S3 are paramount services of the customers A, B and C respectively.

5. CONCLUSION

In this paper paramount web service is selected using refining algorithm which uses sentient QoS parameters. This research work considers both functional and non-functional parameters for selection of services which improves the efficiency and the consumer's percentage of satisfaction. In future the selection algorithm may be extended using semantic matchmaking.

6. REFERENCES

[1] Web Services Selection Algorithm Based on the Quality of Service ,Ekkachai Naenudorn and Suphakit Niwattanakul, Software Technology Laboratory HP Laboratories Palo AltoHPL-2000-39, March, 2000

[2] A QoS Oriented Framework for Adaptive Management of Web Service based Workflows (2003) Chintan Patel, Kaustubh Supekar, Yugyung Lee, In Proceeding of Database and Expert Systems 2003 Conference

[3] Similarity Search for Web Services, Xin Dong Alon Halevy Jayant Madhavan Ema Nemes Jun Zhangflunadong, alon, jayant, enemes, University of Washington, Seattle, Proceedings of the 30th VLDB Conference, Toronto, Canada, 2004

[4] W3C Working Group, QoS for Web Services:Requirements and Possible Approaches, http://www.w3c.or.kr/kr-office/TR/2003/ws-qos/.

[5] Web service selection based on QoS Constraints, Raj, R.J. and T. Sasipraba in Trendz in Information Sciences & Computing (TISC). 2010. IEEE.

[6] WSRank: A Collaborative Ranking Approach for Web Service Selection, Meng, L., J. Li, and H. Sun in Computer and Information Technology (CIT), 2011 IEEE 11th International Conference. 2011. IEEE.

[7] QoS Based Efficient Web Service Selection, Maheswari, S. and G.R. Karpagam, European Journal of Scientific Research, 2011. **66**(3): p. 428-440.

[8] Quality driven web services selection. Hu, J., C. Guo, H. Wang, and P. Zou in e-Business Engineering, 2005. ICEBE 2005. IEEE International Conference. 2005. IEEE.

[9] QoS-based Web Service Selection Using Filtering, Ranking and Selection Algorithm, Tajudeen Adeyemi Ajao, Safaai Deris, Isiaka Adekunle Obasa in International Journal of Scientific & Engineering Research, Volume 4, Issue 7, July-2013 184 ISSN 2229-5518.