

# A Context Service Graph for Web Service Discovery with QoS

Ravi Shankar Pandey

Department of Computer Science, Birla Institute of Technology, Patna

Email: ravishankarbit@yahoo.com

## ABSTRACT

Service discovery infrastructure provides automatic integration of web services to meet new business objectives. It has three actors service provider, service broker and service requestor. The broker provides the requested service from the registry. The searching of services is very difficult process. The services should be compatible in terms of functional aspect as well as non functional features. Several research efforts have been made to discover the correct services which are going to integrate. Service requestor may use same set of services many times or different requestor may need same set of service.

In this paper, I have proposed a model to support service discovery using the current history of set of services involve in execution of a business objective. The context service graph stores the successful execution of set of web services and stores the QoS at start node for comparing the QoS requirement with the new service requestor QoS. The context service graph is present at the broker side along with task details.

Keywords - Web Service, Service Discovery, Quality of Service, Context Service Graph

Date of Submission: Aug 19, 2021

Date of Acceptance: Sep 21, 2021

## I. INTRODUCTION

Service oriented architecture is used to automatic integration of software components for execution of business objectives. This architecture has three actors' service requestor, service broker and service provider. The service requestor makes a request to the service broker. The service broker finds the service from the service registry and gives the url of service provider. The service is executed for the service requestor as given in figure-1.

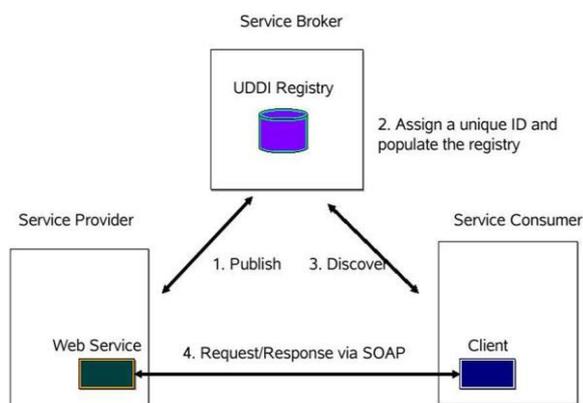


Figure-1: Service Oriented Architecture

In this service execution more than one service may be used. The success of service integration depends on the two kinds of information functional and non-functional feature of the service. The functional features are like input and output parameters type and number. The nonfunctional features are like reliability, availability, latency and space affinity. In some case same set of service may require several times and each time finding the correct set of service again and again is time taking.

In this paper, I have presented a formal model context service graph to store the most recent used set of services to reuse. The most recent services are stored including their invocability sequence with individual web service quality of service parameters and total qos of the services involved. The service requestor sends plane in the form of service requirement. The broker first finds the set of service from the context service graph. If service set is available then sends the urls to the service requestor. This provides more guarantee for execution of services and save time. If service is not available finds the service from the registry as in usual case.

Section -1 describes the introduction. Section -2 explains the related works. Section-3 illustrates my proposal. Section-4 is used for demonstration of my proposal with an example. Last section is for concluding the my proposal

## II. RELATED WORK

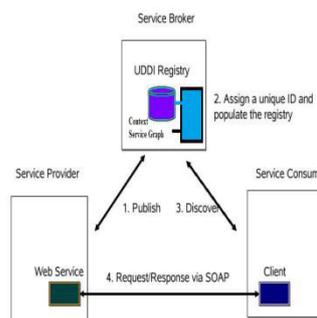
G. Denaro et. al.[1] have proposed Enhanced Service-Oriented Architecture which facilitates the broker. They have developed a interaction protocol service extension (IPSE) which records the web service operation invocation order. This IPSE is to find the correct operation order of a web service which is absent in the WSDL of a web service. This IPSE increases the guarantee of service execution. Initially the IPSE has same operation order as given in the WSDL, it refines over a period of time. San-Yih Hwang et. al.[2] have proposed web service selection based on the reliable web service choreography. The web service finite state machines (FSM) are defined for web service, role and target web service. The one role may be implemented by many web services. The web service choreography dictates the web service invocation and proposed web service invocability. They have categorized web service finite state machine in two categories FSMup and FSMdown. The FSMup is for service consumer while

FSMdown for service providers. The aggregated web service choreography reliability is also calculated. Joyce EI Haddad et. al. [3] have proposed automatic web service selection algorithm for web service composition which also includes the Quality of service (QoS) parameters. They have categorized the web service on the behavior into three categories pivot, compensatable and retrievable. The service selection patterns are classified into two patterns sequential (AND) and parallel (XOR). Lingzhao Zeng et. al.[4] have proposed a method for selecting component web service at the time of execution rather than design time based on QoS. They have used linear programming method based on optimization. The direct acyclic graph is used to represent the execution path of the composite service. Yutu Liu et. al.[5] have proposed a QoS computation model of dynamic web service selection. The QoS registry has proposed which ranks services in registry using the QoS. Their QoS model is applicable for service consumer as well as service providers. The service domain is also used in organizing the services in registry. Freddy Lecue[6] has proposed a method for ranking and optimization of web service composition. They have proposed a formal model of semantic similarities in terms of input/output of a web services. Achraf Karray, et. al. [11]have proposed a service selection based heuristic approach. The several meta models [7, 8, 9] have proposed for description of web services which include the quality of service parameters either based on time and space. Wala Ben Messaouda et. al. [12] have done comparative studies for web service discovery approaches. Randa Hammami et. al. [13] have proposed an algorithm for service discovery.

**III. PROPOSED MODEL**  
**Proposed Architecture**

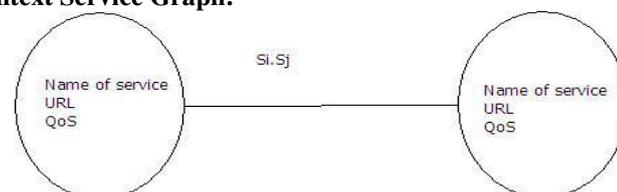
The context service graph module facilitates the service broker to find the correct set of services which are recently used to meet similar business objectives. This works between broker and service registry. This module takes the successful integration of services details like service name, url and next service details link. This information is stored in the acyclic graph data structure. The acyclic graph is used due to service composition  $si.sj$  is not equal to  $sj.si$ . The service requestor makes a request to service broker. The service broker finds the services used in business objective. These services are composed automatically and execute the business objective. The context service graph keeps the successful composition details. It stores the service name, url and QoS of the service at the node of the graph and directed edge is added between previous node and next node. The total QoS is computed and is stored at the first node of the graph and is marked as first node. The one graph may represent one or more service composition only QoS parameters are different for different service composition. The range is stored for the QoS at each node of the graph including total QoS which is stored at the first node. The range is defined using min/max value of QoS as proposed in [10].

This min/max value helps to meet the service requirement provided by the service consumers.



**Figure-2: Service Oriented Architecture with Context Service Graph**

**Context Service Graph:**



**Figure-3 A model of context service graph**

The formal model of the graph is  $\langle S,U,Q,\tau,\sigma \rangle$ . The S is set of services, U is set of urls, Q is quality of service parameters,  $\tau$  is transition function which gives the urls of services  $s1$  and  $s2$  and  $\sigma$  is state function which stores current context of the graph. The  $\sigma$  is composition of  $\sigma_{sn} * \sigma_{ur} * \sigma_q$  where  $\sigma_{sn}$  keeps service name,  $\sigma_{ur}$  stores current url of service. The same service name may have more than one urls and  $\sigma_q$  has the value of current quality of service parameters. The formal model is given below.

$$\langle S,U,Q,\tau,\sigma \rangle$$

$$\sigma = \sigma_{sn} * \sigma_{ur} * \sigma_q$$

$$\sigma_{sn} = N * SN \rightarrow SN \text{ where } N \text{ is name and } SN \text{ is service name.}$$

$$\sigma_{ur} = N * U \rightarrow U \text{ where } N \text{ is name and } U \text{ is set of rules.}$$

$$\sigma_q = N * Q \rightarrow Q$$

$$\tau: S * S \rightarrow U$$

The formal model of context free graph is  $\langle G,R,Q,\Omega \rangle$  where G is graph  $R = [t1..tn]$ , Q is user requirements where  $t1..tn$  are small tasks, Q is quality of service parameters which includes reliability and availability other quality of service parameters are not considered for making it simple and  $\Omega$  is transition function. The transition function takes the user requirements, set of stored context free graph and QoS and returns the matched context free graph as per user's requirements. The table -1 contains the sample of context service graph G1 along with Min1, Max21 QoS and G2 is another graph address and respective min/max QoS.

$$\Omega: G * R * Q \rightarrow G$$

Graph_Address	QoS
G1	Rel (Min1, max1) Avl (Min1, max1)
G2	Rel (Min1, max1) Avl (Min1, max1)

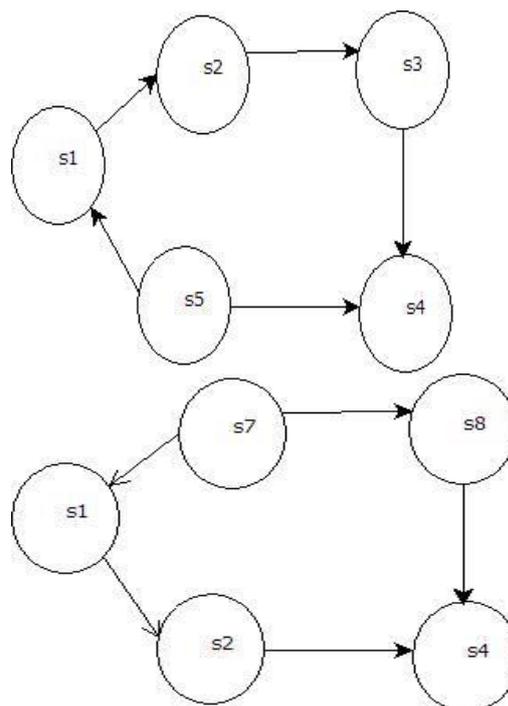
**Table-1: Context free graph**

**Algorithm: Create a context service graph**  
**Input: WSDL from registry**  
**Output: Graph which stores service details along with QoS and urls**

1. Create a node for new web service description gets from service discovery
  - 1.1 Extract the service name ,url and QoS from WSDL
  - 1.2 Stores the address of next node which keeps the information of other WSDL which is invoked by previous web service.
  - 1.3 If service name is already present in the node only change QoS value of new WSDL  
 If new QoS is less than min new is assigned as min and if QoS is greater than the max new is assigned as new max
2. Repeat step 1 till discovery of web service continues.

**IV. EXAMPLE**

In this example, I have considered the services customer (s1,<.30,.80>,<.40,.70>), travelagent(s2,<.50,.90>,<.60,.90>), airline(s3,<.40,.70>,<.60,.90>), train(s4,<.60,.90>,<.70,.90>), visa(s5,<.50,.90>,<.60,.90>), limbo(s6,<.30,.90>,<.40,.80>), bus(s7,<.60,.90>,<.40,.80>) and master(s8,<.70,.90>,<.50,.90>). The two service compositions are demonstrated in figure-4. The service composition is g1(s1,<.30,.90>,<.40,.90>,s2,s3,s4,s5) to execute a travel plane. The second composition is g2(s1,<.30,.90>,<.40,.90>,s7,s8,s4,s2) . The broker finds first the user request from the context service graph module. The result is positive gives the service stored in the graph. Otherwise broker finds result from the registry.



**Figure-4 Examples of Context Service Graph**

Graph Address	QoS
g 1(s1,s2,s3,s4,s5)	<.30,.90>,<.40,.90>
g2(s1s7,s8,s4,s2)	<.30,.90>,<.40,.90>

The Customer WSDL sample file is used to extract the information for node of the context graph. The module takes input WSDL and finds the service name form service definition, url from binding and QoS from the documentation section and stores in the node. The service name is Customer, url is http://example.com/Customer and qos parameters are Service\_reliability =.40, .90, Service\_Availaibility =.30, .80.

```
<?xml version="1.0"?>
<definitions name="Customer"
targetNamespace="http://example.com/ Customer /service"
xmlns:tns="http://example.com/ Customer /service"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:defs="http://example.com/ Customer /definitions"
xmlns="http://schemas.xmlsoap.org/wsdl/">
<import namespace="http://example.com/ Customer /definitions"
location="http://example.com/ Customer / Customer.wsdl"/>
<binding name=" CustomerSoapBinding"
type="defs:Customer PortType">
```

```
<soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name=" queryFlights ">
    <soap:operation
soapAction="http://example.com/GetLastFlights"/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
  <service name=" CustomerService">
    <documentation> <wsdl:documentation>
< wsdl : documentation >
< wsdl : qos >
< wsdl : Service_reliability=<.40,90>>
< wsdl : Service_Availaibty=<.30,.80> >
< wsdl : qos >
<wsdl : documentation >
  </documentation>
    <port name=" CustomerPort" binding="tns:
CustomerBinding">
      <soap:address location="http://example.com/
Customer"/>
    </port>
  </service>
</definitions>
```

## V. CONCLUSION

In this paper, I have presented a formal model for faster discovery of web services using the recent web service combinations to execute business objectives. In this proposal, I have used the directed graph data structure to keep the set of compatible web services information like service name, url and QoS. The context graph table stores the graph start node address and qos of the total web services involved in the execution. The same services involve in execution plane only change the qos in the context graph table. The service requestor makes a request and first checks in the context graph table and compares the qos and services details. If available then get the set of service from the context graph otherwise finds in the registry.

## REFERENCES

- [1] G. Denaro, M. Pezze, D.Tosi, "Towards self-adaptive service-oriented architectures" In: International Symposium on Software Testing and Analysis (ISSTA '06), ACM July 2006, pp 10-16.
- [2] San-Yih Hwang, Wen-Po Liao, Chien-Hsiang Lee, "Web Services Selection in Support of Reliable Web Service Choreography", In Proc. of 2010 IEEE International Conference on Web Services, p.p. 115-122
- [3] Joyce El Haddad, Maude Manouvrier, and Marta Rukoz, "TQoS: Transactional and QoS-Aware Selection Algorithm for Automatic Web Service Composition", IEEE Trans. on Service Computing, Vol. 3, No. 1, January-March 2010, pp.73-85

[4] Liangzhao Zeng, Boualem Benatallah, Marlon Dumas, Jayant Kalagnanam and Quan Z. Sheng, "Quality Driven Web Service Composition" , In Proc. of the 12th international conference on World Wide Web ,Budapest Hungary, ACM-2003, p.p.411-421

[5] Freddy Lecue Optimizing, "QoS-Aware Web Service Composition" , In Proc. of the 8th International Semantic Web Conference(ISWC '09),Chantilly, VA, Springer-Verlag, pp.375-391

[6] Yutu Liu, Anne H.H. Ngu and Liangzhao Zeng," QoS Computation and policy in Dynamic Web Service Selection", In Proc. of 2010 ICACT, New York USA ACM pp 66-73

[7]The paper entitled "A Meta-Model Based Proposal for QOS of WSCDL Choreography", authored by Shri R.S.Pandey and Prof. B.D.Chaudhary, published in proceeding of "International Multi Conference of Engineers and Computer Scientists 2010", Hong Kong, held during 17th to19th March 2010

[8]The paper entitled "A Space Based QoS Meta Model of Web Service ", authored by Shri R.S.Pandey, published in proceeding of "International Conference on Knowledge Discovery and Computational intelligence- ICKD CI' 17 (March 9th &10th, 2017)", March 2017

[9] A. DAmbrogio," A model-driven wsdl extension for describing the qos of web services", In Proc. of IEEE Int. Conf. on Web Services(ICWS 06), Washington, DC, USA,2006, p.p. 789-796

[10]Ravi Shankar Pandey and B. D. Chaudhary, " An Estimation of Min-Maxof QoS Attributes of a Choreography" in Proc. Advances in Engineering,Science and Management (ICAESM12) pp.872-878 , 30-31 March 2012, Nagapattinam, Tamil Nadu ,ISBN: 978-1-4673-0213-5

[11]Achraf Karray, Rym Teyeb and Maher Ben Jemaa, "A Heuristic Approach for Web-Service Discovery and Selection," In: International Journal of Computer Science & Information Technology (IJCSIT'13), April 2013, vol 5, p 107-119.

[12] Wala Ben Messaouda\*, Khaled Ghediraa , Youssef Ben Halima "Towards behavioral web service discovery approach: State of the art" In: 20th International Conference on Knowledge Based and Intelligent Information and Engineering Systems ScienceDirect Procedia Computer Science 96 ( 2016 ) 1049 – 1058

[13] Randa Hammami , Hatem Bellaaj and Ahmed Hadj Kacem "A Novel Approach for Semantic Web Service Discovery" 25th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises 978-1-5090-1663-1/16 \$31.00 © 2016 IEEE DOI 10.1109/WETICE.2016.62