AN ENHANCED METHOD TO COMPOSE BUSINESS PROCESS WEB SERVICES USING BPEL AND OPTIMIZED QoS PARAMETERS

J. Ghayathri[1], S. Pannirselvam[2]

[1]Associate Professor in Computer Science (PG), Kongu Arts and Science College (Autonomous), Erode TamilNadu, India.
[2]Head & Associate Professor in Computer Science Erode Arts and Science College (Autonomous), Erode, Tamilnadu, India

Abstract
Web services are reusable software components available across the web. These web services fulfill the requirements of the consumers in the form of business process requirements, software and technical requirements and other personal requirements. In order to achieve BPEL (Business Process Execution Language) process's QoS (Quality of Service) global optimization on dynamic Web service composition, the traditional dynamic Web service invocation model of BPEL process is enhanced and the service selection, refining and fault tolerance is added. Meanwhile, by improving the grading method, the paper designs and realizes the algorithm, whose result is the global optimization on QoS under the constraints. Through the comparison, the conclusion is that the optimal solution can be found more quickly by the improved algorithms and the business process's QoS global optimization under the constraints can be achieved by the modified model. This paper aims to determine a subset of web services from the candidate services to be invoked at run-time so as to successfully and efficiently orchestrate a composite web service.

Keywords: Qos, Optimization, Refining, WSDL, Business Process, Process Template And BPEL.

---***---

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.

2. RELATED WORK

Erdogan Dogdu, Omer Mescigil implements ActiveBPEL which allows users to specify QoS parameters and uses service selection algorithms on a working BPEL-based execution engine to select appropriate service providers during run-time, so that it improves the engine performance towards higher system throughputs [2]. Sheng SU, Haijie YU proposes a method to compute the cost and values of QoS criteria for concrete composition service with different operation relations. The authors [5] use a genetic algorithm with two layer genes to solve the composition service selection problem. Jean M. Bacon, Brian F. Cooper present an service selection algorithm for QoS-aware composition in dynamic service environments which overcomes the shortfalls like adaptive service composition and time availability for service selection and composition[4]. Junhao WEN et al. describes about the dynamic web services composition using QoS global optimization in BPEL and proved the efficiency of work using an improved branch and bound algorithm [10]. The authors J Ghayathri and Dr. S. Pannirselvam [1] discuss about the selection of efficient and best web service based on users' QoS constraints using the enhanced refining and selection algorithms.

3. METHODOLOGY

The real value of web services lies in the composition of set of related web services. Usually the various service providers expose their capabilities in the form of APIs (Application Program Interfaces) as web services. Based on the requirement and need of the consumer the suitable capabilities are composed together into a single composite service. This can be achieved by the consumer themselves or through third party applications.

3.1 The Service Selection Algorithm With Qos Global Optimization

This previous work [1] of this paper's authors selects the most favorable service through the following processes.

a) *Service tracking*: This process discovers the related services of the required type of service. A set of related services have been selected from the service repository which holds the WSDL files of the service providers. To make the selection process much efficient in addition to the service name matching the operations are also matched with the requirement and based on that the set of relative services are selected.

b) *Service Refining*: This process refines the selected set of related services by applying QoS parameters response time, availability, performance and reliability of web services. The QoS parameters considered in this system are response time, availability, performance and reliability. The services which pass through the refinement constraints form a set of refined services using the service refining algorithm. The normalized weighted values are used to refine the services.

c) *Service Scoring*: For the set of refined services the scoring is calculated by summing up the values of the QoS.

d) *Service Selecting*:   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.

In the normalization process, Equation (1) is used for reliability, availability and performance parameters that require maximization whereas Equation (2) is used for response time that requires minimization.

$$qp = (q - qmin)/(qmax - qmin) \quad \text{----- Eqn.} \quad (1)$$

$$qn = (qmax - q)/(qmax - qmin) \quad \text{----- Eqn..} \quad (2)$$

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:

$$q = (qc - y)/(qmax - y) \quad \text{----- Eqn.} \quad (3)$$

$$q' = (qmax - qc)/(qmax - y) \quad \text{----- Eqn.} \quad (4)$$

qthreshold != y, 1 if qthreshold = y
where qc represents the QoS value for the parameter being considered, y is the default threshold value for the parameter being considered.

3.2. Proposed Model of Dynamic Service Composition In BPEL

To get the composite web service for a business process the consumer has to endure a complex process such as finding the right services among numerous services available, As shown Fig.-1, is the enhanced model of dynamic Web service invocation of business process. In the enhanced model,   Optional service set is similar to the matrix of memory, which store the candidate service set of each service node of BPEL process. In the service selection component, there is the improved branch and bound algorithm.
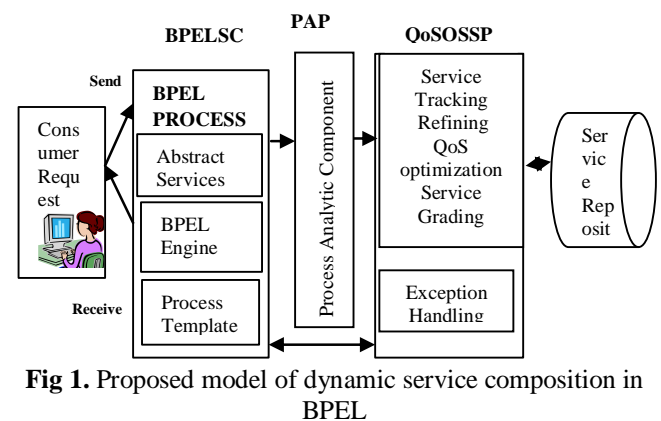

Fig 1. Proposed model of dynamic service composition in BPEL

By the algorithm, the service selection component can select the proper service from many candidate services, then after dynamic service binding, BPEL process meets the constraints and achieves the QoS global optimization. The optimal services list stores the optimal solution generated by the service selection component. When the process is running, if a service fails on any cause, exception handling mechanism can replace the service in time to ensure the process running smoothly.

BPEL Designer
The business expert defines the business process using the BPEL Designer. After defining the business process flow the process logic template containing the process flow logic would be generated by the Designer in the background. This process template would be executed at run time by the BPEL Engine.

Process flow template
The primary task of process flow template is to grab the business process logic. The BPEL specification [11] is used to design the process flow template. At the time of designing the business process it is generated from the BPEL designer   and executed by the BPEL Engine at runtime.

BPEL Engine
The BPEL Engine is integrated within the Application Server. The BPEL Engine includes the functions such as mapping of the data content, error handling, transactionality, security, and so forth.

The proposed model consists client request and services repository along with three computing phases such as a) BPEL Services Composition phase (BPELSCP), Process analysis phases (PAP) and QoS based Optimized service selection phase (QoSOSSP). The flow of operations is given in the Fig. – 2.
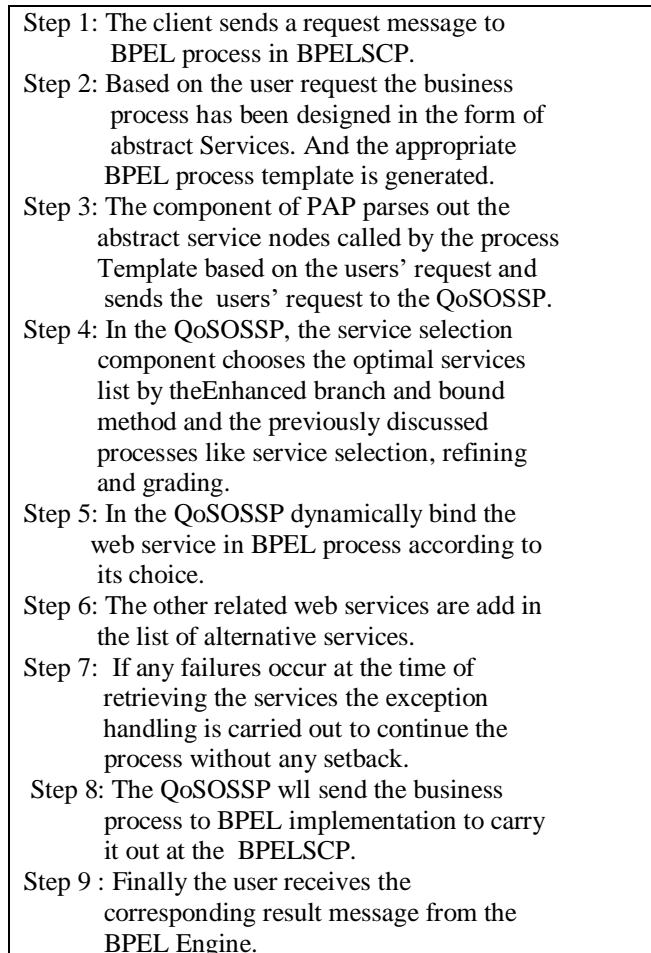
Step 1: The client sends a request message to BPEL process in BPELSCP.

Step 2: Based on the user request the business process has been designed in the form of abstract Services. And the appropriate BPEL process template is generated.

Step 3: The component of PAP parses out the abstract service nodes called by the process Template based on the users' request and sends the users' request to the QoSOSSP.

Step 4: In the QoSOSSP, the service selection component chooses the optimal services list by theEnhanced branch and bound method and the previously discussed processes like service selection, refining and grading.

Step 5: In the QoSOSSP dynamically bind the web service in BPEL process according to its choice.

Step 6: The other related web services are add in the list of alternative services.

Step 7: If any failures occur at the time of retrieving the services the exception handling is carried out to continue the process without any setback.

Step 8: The QoSOSSP wll send the business process to BPEL implementation to carry it out at the BPELSCP.

Step 9 : Finally the user receives the corresponding result message from the BPEL Engine.

Fig. -2 Flow of operations

The process template is formed based on the abstract services nodes as $(S_1, S_2, S_3 ||_{S4} ---|| \ S_i; \ :S_{i+1} :S_{i+2} --- :S_j \ , \ ^n S_k ,--- ,S_n)$, "||" denotes parallel services, ":" denotes choice of the services, and "^n" denotes n-cycles of the services. Generally, there are four basic service invocation models: sequence, parallelism, choice and cycle. Some calculation methods applied for the four basic models QoS parameters to further abstract each service node of the business process, finally, the business can be abstracted as (W1, W2, W3, _ _ _ , Wn) which is the sequence model [10]. In the step 3, according to the abstract interface of each service node, the QoSSSP inquiries the service repository of the service registration institution which is either publically available or private to the particular provider and deposits the candidate service set in the refined services list. In the step 4, the service selection component chooses the optimal services list by the improved branch and bound method, and then deposits the optimal list in the optimal service list for the invoke activity of BPEL process to invoke. After BPEL process dynamically binds services, its QoS is optimal under

the constraints. In the step 8, when a binding service in the process generates an exception, the model will send the exception message to the exception handling mechanism which selects a similar service to replace the exceptional service and enables the process to continue. The proposed model can realize dynamic Web service composition with QoS global optimization by adding the QoSOSSP and BPELCSP.

According to the calculation method of the QoS value of the optimization model, `cost' value of the composite service `S' is

$$QoS_{cost}(S) = \sum_{i=1}^{n} QoS_{cost}(W_i)$$

according to the weighted normalized values from the formula 1,2,3,4 of the set of services that form the required composite web service. The `cost' value of every candidate service among the candidate service set, for example, the candidate service set $(W_{i1}, W_{i2}, \_ \_ \_ , W_{im})$ of the service node `W_{i}", that is, W_i have m candidate services. Then the QoS value of every candidate service which belongs to a service node's candidate service set in the process, form a two dimensional matrix, given in the Table -1.

Table – 1 List of candidate services

NV(w_{11})	NV(w_{21})	NV(w_{31})	NV(w_{n1})
NV(w_{12})	NV(w_{22})	NV(w_{32}) . . .	NV(w_{n2})
NV(w_{13})	NV(w_{23})	NV(w_{33}) . . .	NV(w_{n3})
NV(w_{1m1})	NV(w_{24})	NV(w_{34}) . . .	NV(w_{n4})
0	NV(w_{2m2})	NV(w_{3m3}) . . . NV(w_{nmn})	

$QoS(W1;W2; \_ \_ \_ ;Wn) =$ in which, the rows of the matrix is M=max$\{m_1, m_2, \_ \_ \_ , m_n\}$, m_i is the number of candidate services of the i-th service node, the columns of the matrix is the number of the service node, which is named `N'. The element $QoS(W_{ij})$ is a data structure, which contains the `Cost' value. The i-th row of the matrix represents the QoS value of i-th candidate service of every service node, if a service node does not have i-th candidate service, then `0' is placed in the position. The j-th column of the matrix represents the QoS value of every candidate service of a service node. The matrix contains sorted values of the QoS parameters. Then the solution space tree for service composition is generated as per the values of the matrix, as shown in Fig.-.3.
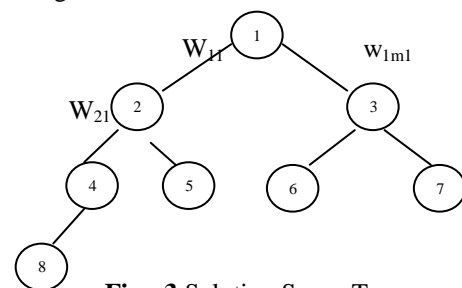

Fig. -3 Solution Space Tree

The solution space tree contains all possible service compositions. In the tree, if a node's child node is more near the left, its $QoS_{cost} (W_{ij})$ is greater. The composite service's constraint is used to judge a node whether which can become an expansion node. Figure-4 depicts the Enhanced

Dynamic web Service Composition Algorithm (EDSCA) with global optimization based on BPEL and QoS constraints, which is realized by the combination of the matrix and the enhanced algorithm based on the priority queue and tree.

Table -2 Notations used in the Enhanced Dynamic Service Composition Algorithm (EDSCA)

S.No.	Notation	Description
1	PQ	Priority Queue
2	SRQ	Service Results Queue
3	RQ	Result Queue
4	EN	Extended Node
5	CS	Constraint satisfaction

```
Algorithm : EDCSA()
Input: QoS (W₁, W₂,---, Wn ) A 2-Dimensional Matrix
Output: RQ which is the optimal solution;
{
Step1:// Initialization.
Initialize Tree t , PQ, SRQ, EN with null;
//  EN points the current node
Step 2: //2-Dimensional matrix with selected
services
QoS[M][N] = QoS (W1, W2, _ _ _ , Wn);
Step 3 : // Tree is formed with the constructed matrix
values
t = QoS[M][N];
Step 4:  // Shortest path with satisfied QoS conditions
formed
PQ. push(t.root);
While (! PQ.empty())
{
EN = PQ.pop();
If (CS(EN))
{ // If the current node `cn' satisfies the
given constraints
CS(cn), the node cn,
will be added into the optimal
solution queue.
RQ.push (ENode);
If (EN.child !=null)
{
For i=1 to EN.child.length
{
PQ.push (EN.child[i]);
}
}
}
           }
}
```

Fig. - 4 Enhanced Dynamic web Service Composition Algorithm (EDSCA)

The algorithm EDCSA selects the solutions which satisfies the constraints, so that the storage space of the tree can be reduced. The living node is change into expansion node and it pushes its all child nodes of meeting the QoS constraints

into the priority queue, the rest child nodes will be abandoned. Then select a living node from the priority queue and change it into the expansion node, this process is repeated until the priority queue is empty or the optimal result is found. In addition to this the algorithm handles the failure cases in the form of exception handling mechanism. When an exception happens, the QoSOSSP does not need to inquire the solution space tree for new optimal solution and bind it with the business process again. When the candidate service `$W_{ij}$ ' belonging to the service node `W_i' occurs an error, the error service `$W_{ij}$ ' is replaced with next service `$W_{i.j+1}$', then new service composition is still optimal. This optimal solution finding algorithm finally gives a composite service by meeting the required constraints of the consumer.

4. PERFORMANCE ANALYSIS

Compared with the existing model the matrix has been sorted after placing the services. In the proposed model the sorting of normalized services is done at QoSOSSP itself which reduces the execution time. In the existing method the matrix itself is sorted which increases the time consumption during the execution of process. The Chart – 1 shows the comparison of time consumption of web services composition using the existing method and the proposed method where the number of the process's service nodes is same and the number of every service node's candidate services is same. The proposed BPEL model with enhanced algorithm is more efficient than the existing model.
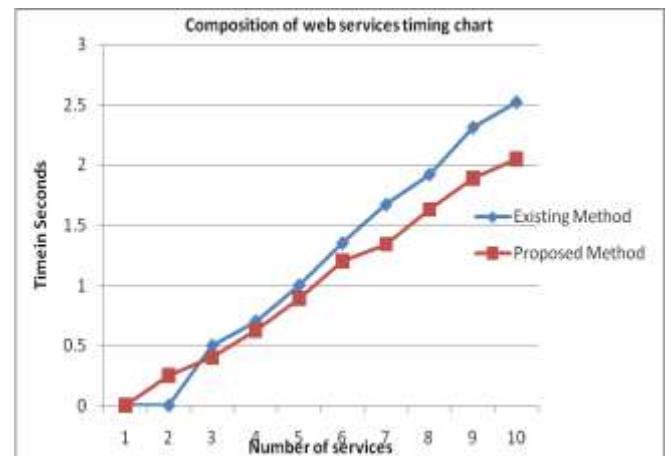

Chart -1 Comparison Chart

5. CONCLUSIONS

In the paper, on the basis of dynamic invocation of web services in BPEL process, the QoS optimization process is included. Through the enhanced method, QoS value of Web services is calculated and the proper services are selected for composition. Finally, after dynamically binding the services, the QoS value of the whole process is optimal under the constraints. As the available number of web services is increasing there is a demand for selecting the appropriate candidate services also proportionally increase. The further research will focus on solving the abovementioned issues.

ACKNOWLEDGEMENT

REFERENCES

[1]. J. Ghayathri , Dr.S. Pannirselvam, "Selection of paramount web service using sentient QoS parameters", International Journal of Computer Applications, 2015.

[2]. Erdogan Dogdu, Omer Mescigil, TOBB University of Economics and Technology, Computer Engineering Department, Ankara, Turkey, " Dynamic Selection of Web Services Using QoS Constraints in BPEL", Proc. of the 17th Int. Conf. on Software Engineering and Data Engineering (SEDE'08), 2008.

[3]. Shujun Pei , Deyun Chen, Harbin University of Science and Technology, Harbin, China, "Research on Dynamic Web Services Composition Framework Based on Quality of Service", Information Technology Journal (10) 8:1645-1649, 2011

[4]. Jean M. Bacon, Brian F. Cooper, "QoS-aware Service Composition in Dynamic Service Oriented Environments.", Middleware 2009 - ACM/IFIP/USENIX, 10th International Conference, Nov 2009, Urbana, IL, United States. Springer, 2009,

[5]. Sheng SU, Haijie YU , University of Electronic Science and Technology of China, China, "Composition Service Selection with Complex Service Participant Relationships and Optional QoS Values ", Journal of Computational Information Systems 7:3 , 707-720, 2011

[6]. Mohammad Alrifai, Thomass Risse, L3S Research Center, University of Hannover, Germany, "Combining Global Optimization with Local Selection for Efficient QoS-aware Service Composition", WWW 2009, April 20–24, 2009, Madrid, Spain.ACM 978-1-60558-487-4/09/04.

[7]. Maya Rathore and Ugrasen Suman, School of Computer Science & IT, Devi Ahilya University (DAVV), Indore (M.P.), India, "QoS Broker based Architecture for Dynamic Web Service Discovery and Composition" , International Journal of u- and e-Service, Science and Technology Vol. 7, No. 6 (2014).

[8]. Florin Leon, Costin Badica, "A Freight Brokering System Architecture Based on Web Services and Agents", Proceedings of the 7th International Conference IESS 2016, Bucharest, Romania, Lecture Notes in Business Information Processing, vol. 247, pp. 537-546, Springer International Publishing Switzerland, 2016.

[9]. Dinesh Ganesarajah, Orbis, NDS, London, UK, Emil Lupu Department of Computing, Imperial College, London, UK, "Workflow-based composition of web-services: a business model or aprogramming paradigm?", Proceedings of the 6th International Enterprise Distributed Object Computing Conference (EDOC 2002), 17-20 September 2002, Lausanne, Switzerland, Proceedings. IEEE Computer Society 2002.

[10].Junhao WEN, Yu HUANG, Zhiqiang SHU, Peng LI, College of Computer Science, Chongqing University, Chongqing , China, "Research on Dynamic Web Service Composition with QoS Global Optimization in BPEL", Journal of Computational Information Systems 8: 2 , 2012.

[11].Kumar Raj Moorthy, "Introduction to BPEL", www.developer.com/author, May 2006

[12].Chiaen Lin, Krishna Kavi, Department of Computer Science and Engineering, Unitersity of North Texas, Denton, TX 76203 USA, "A QoS-Aware BPEL Framework for Service Selection and Composition Using QoS Properties", International Journal on Advances in Software, vol 6 no 1 & 2, 2013.

BIOGRAPHIES

J. Ghayathri, MCA, M.Phil, Research Scholar in Ph.D., Associate Professor in Computer Science (PG), Kongu Arts and Science College (Autonomous), Erode, Tamil Nadu, India.

Dr. S. Pannirselvam, M.Sc., M.Phil., Ph.D., Head & Associate Professor in Computer Science, Erode Arts and Science College (Autonomous), Erode, Tamil Nadu, India. Area of Specialization is Digital Image Processing and Data Mining 30 Years of Experience in the field of Computer Science. The author published 21 papers in various International Journals and 12 papers in National journals