

# A Web Service Composition Framework based on Functional Weight to Reach Maximum QoS

M.Y. Mohamed Yacoab, Abdalla AlAmeen, M. Mohemmed Sha  
Department of Computer Science, College of Arts and Science,  
Prince Sattam Bin Abdulaziz University

**Abstract**—The recent trend in the web world is to accomplish almost all the user services in every field through the web portals of the respective organizations. But a specific task with series of actions cannot be completed by a single web service with limited functionality. Therefore, multiple web services with different functionalities are composed together to attain the result. Web service composition is an approach that combine various services to fulfill the Web related tasks with preferred quality. Composition of such services will become more challenging when these web services are with similar functionalities, varying Quality and from several providers. Hence, the overall QoS (Quality of Service) could be considered as the major factor for composition. Moreover, in most of the compositions the expected QoS cannot be attained when the task is finished. Sometimes the complete task may have affected by a poor performed single web service. So, while composition, at most care should be taken to select a particular web service. Composing web services dynamically is the main method used to overcome these difficulties. However, to reach the actual functionality of the specific task the quality of each individual service is very much necessary. The QoS of a web service normally evaluated using the non-functional attributes, such as response time, availability, reliability, throughput, etc. Also, while composition, the same level of quality is not expected for individual web services that are included in the chain. So, a framework proposed in this research paper, for web service composition by setting appropriate weightage for the non-functional parameters. Experimental results show that implementation of this method will definitely pave the way to reach the maximum performance of the composition with improved QoS.

**Keywords**—Web service; composition of services; non-functional parameters; QoS

## I. INTRODUCTION

Web-oriented services are considered as an application component that makes itself available over the internet. A single web service cannot useful to attain the desired specific task in all the cases. So different services are grouped together to get the work done. The existing web services are grouped together as a chain in some order to attain the target. The main parameters considered while composing such web services may vary, but the target is to reach best result with the expected level of quality. While selecting a particular web service, the quality of the service is measured and its performance in the real time also taken care to maintain the quality [1], [6]. Sometime the external factor such as network traffic can also influence the performance of the service. Any how the ultimate aim to reach the genuine functionality of the desired web-oriented service. When it is a single web service

the QoS is evaluated mainly from non-functional parameter and which is considered as the base factor [11]. It is more complex when combining the services together in some order to reach the entire functionality. A single web service in the group may affect the overall performance of the composed services. So extreme care should be taken to compose the web services together to achieve the expected outcome. Also, it is interesting to see that, the same level of quality for all the web services are not required to gain the actual functionality. It means that the same level of QoS is not essential for all the web services needed to compose [10]. Different frameworks and methodologies are suggested for composing the web services with varying QoS. Most of the methods consider the overall QoS is the major factor after composition of web services rather than considering the individual service quality. So, this work considers the problem in two different aspects. First a pool of web services is grouped based on the individual QoS if these services are up to the mark. Here the QoS is evaluated after setting the appropriate weightage for the non-functional parameters of the service. Secondly, the best arrangement of composition is selected based on the real-time availability of the services from each pool that make up the composition.

The proposed framework and its related methods to compose the web services are defined in Section III. The results are vastly discussed in Section IV and finally concluded in Section V.

## II. RELATED WORK

Web services are always termed with it functional and non-functional properties by its providers. Web service composition is the organization of those services in some order to perform a particular task. Service providers sell their dynamic web service components in the international market so that these services can be used by the customers [7]. The important property to explain a web task is that takes account of the signature, session states and the requirements of specific criteria. Non-functional values are made to assess the cost involved, the quality of service provided and to track issue related to security factors of the web composition [14]. While composing a web service, these measures are considered to check whether the service is up to the expected level of functionality. Several approaches are proposed to compose the services based on its QoS, but most of them focusing on the overall quality after composition.

Yu, T. et al. [1] modelled the problem in two different models such as the analytical model and the chart prototypical

model. The analytical design discusses the key issues on n-dimensional zero-one Knapsack Problem. The chart prototypical model expresses the issues as the Multiple Restriction Best Route value process. These algorithms also proposed to insight their achievements by test cases performance.

L.Zeng [2] proposed a method to build a high quality web task while designing a web composition. First the user requirement is considered, then the quality of web tasks is calculated, finally web tasks are selected to reach the quality as estimated for the composition of web service. The parameters of QoS viz., response instance, cost expenditure, execution duration, reput, successful execution rate and readiness are used here.

Berbner [4], [5] present a structural design called Web task merit of Service Design Extension that helps dynamic linking and binding of Web tasks at dynamic compilation along with monitoring mechanisms. Heuristics problem solving methods are proposed to make use a collection procedure calculates the total task combination. The sequential form procedure that uses several QoS attributes is proposed in this approach.

Jiuyun et al. [12] suggested an immune algorithm to handle the composition problem. Here they consider multiple quality parameters with different business process flow variables to make the attribute such as cost involved in service; it's time for respond, task availability and consistency.

Alrifai et al. [9] propose a heuristics algorithm divides the most important problem to number of small tasks then by finding solution to these as a best possible result can be generated. Finally, the algorithm use data gathering procedure for calculating the total Quality of Service Parameters.

The papers [8], [12] used Genetic algorithms to solve the composition problem. The fitness function used here compare solutions by considering all forms of workflow in business process. The approaches [3], [9], [13], [14] also present the composition of web services based on QoS evaluated from the non-functional parameters. Most of them focus on composing of web services in different aspect mainly the overall QoS, but not much concentration given to the individual web service performance based on its functionality.

### III. COMPOSITION METHDOLOGY

Web services are published by its providers along with the attributes such as signature, states and non-functional parameters. Always non-functional parameters such as response time, reliability, availability, etc. are mainly used for evaluating the quality of a web service [6]. Fig. 1 illustrate the proposed framework to compose the services that help to attain the best possible arrangement of web task to execute an appropriate work. Initially a pool of web services is grouped based on its functionality and individual QoS, to understand whether it will be used in composition chain to complete a specific task. Next appropriate weight is assigned for the non-functional parameters for evaluating the QoS of the service. These services are ranked and the services at the satisfaction level are added in the group.

All these services with specific category are stored in the task service repository. The procedure generator took the functionality attributes of a particular task as initial data input, and generate the system representation that explains the combined task services. The system design model holds the combination of chosen dynamic tasks and their different flow models associated with those services. In contrary we might have found that, many of web task services use to have the identical or equal structure to define a particular web service. In that situation the web architect will design a number of web service tasks that meet out necessary plan of requirement [1]. At this moment, the complex tasks are estimated by the total overall quality from the value stated from the requirement of specific tasks parameters. Normally the person one who request the service, supposed to identify the weight values to each individual requirement of specific tasks parameters.

Finally, the composite service with best quality will ahead in the rank list. Following the evaluation, the individual qualities of the service are maintained as a group. To compose the services, the individual services from each group is selected based on rank and availability. The process generator uses appropriate algorithms that consider the services from each individual group to set the best composition. Once a good composite process is identified and selected, the web service is in the ready state and the attained service to be executed. Carrying out the implementation of a composite web task the sequence of message passing across the service are well explained in the process model [1], [6]. The dataflow in the composition is defined as the actions that the output data from the executed service transfers to the input to the following service.

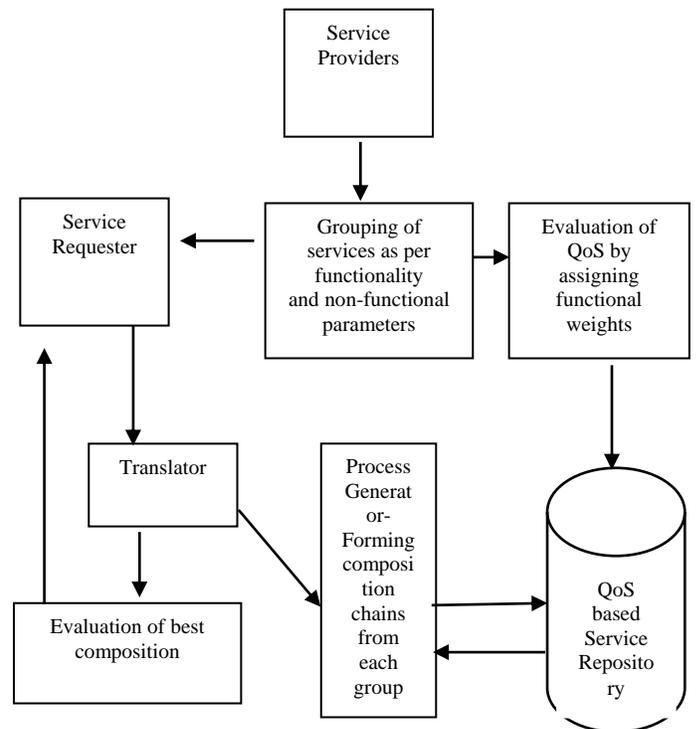


Fig. 1. Functionality based web service composition framework.

The methodology of functionality based composition of web services include the following steps.

**A. Identify the Functionality Equivalent Web Services in each Group that Are Needed to Compose**

The first step is to identify the group of web services that are to be included in the chain of composition. In each group the services that can satisfy functionality only be added. In this step the atomic web services collected into a specific category based on the minimum expected QoS and functionality. For that the non-functional parameter values are verified with the expected level to reach the actual functionality [11].

For example, the services that need to complete the task for online book purchasing system may be as follows:

*Book Ordering service* → *Warehouse service* → *Payment service* → *Delivery service*

The services with the above four categories are grouped based on its QoS and evaluated by fixing appropriate functional weights.

Consider  $G_1WS_1, G_1WS_2, \dots, G_1WS_n$  are the services in the first category. Similarly, in each category the services in other groups are also collected.

The non-functional parameters  $P_1, P_2, P_3, P_4$  and  $P_5$  are considered to evaluate the QoS for individual services are as follows:

- $P_1$  - Response time     $P_2$ - Availability
- $P_3$  - Throughput       $P_4$ - Successibility
- $P_5$  - Reliability

Table I lists all the average non-functional attribute values for 10 web services in the group  $G_1$  are as follows:

TABLE I. WEB SERVICE NON-FUNCTIONAL VALUES

S.no	Web services	P1 (ms)	P2 (%)	P3 (Inv/s)	P4 (%)	P5 (%)
1	$G_1WS_1$	290	90	5	95	80
2	$G_2WS_2$	210	95	4	99	60
3	$G_3WS_3$	135	65	8	65	75
4	$G_4WS_4$	140	80	3	80	70
5	$G_5WS_5$	255	80	9	80	75
6	$G_6WS_6$	145	99	25	99	70
7	$G_7WS_7$	165	85	30	85	80
8	$G_8WS_8$	130	90	13	99	75
9	$G_9WS_9$	160	95	2	95	80
10	$G_{10}WS_{10}$	125	90	8	90	70

**B. Ranking of the Web Services based on QoS after Setting the Weight to the Non-Functional Parameters**

The performance of the composition is directly depending on the extent to which the atomic services reach its functionality. So, the appropriate the weightage is fixed by the requester of the service to the individual non-functional parameter [14]. The same weight is assigned in each category but not required to be for the other web services in different categories.

For example,  $G_1$  is a group that contains the services with similar functionalities. Here priority is given to those services based on a particular non-functional parameter that influence the service to reach the actual functionality. Based on that a particular weightage is given to all the non-functional parameter with expected level [5].

Let  $G_1WS_i, G_2WS_i, \dots, G_5WS_i$  are the categories of services included in the composition in order.

The weights assigned to the non-functional parameters  $P_1, P_2, P_3, P_4$  and  $P_5$  are represented as in Table II.

TABLE II. WEB SERVICE NON-FUNCTIONAL WEIGHTS

S. no	Service Name	W1 (0-1)	W2 (0-1)	W3 (0-1)	W4 (0-1)	W5 (0-1)
1	$G_1WS_i$	0.4	0.69	0.76	0.52	1
2	$G_2WS_i$	0.55	1	0.68	0.91	0.84
3	$G_3WS_i$	0.91	0.87	0.93	0.78	1
4	$G_4WS_i$	1	0.73	0.64	0.82	0.85
5	$G_5WS_i$	0.83	0.69	0.84	1	0.85

The QoS of each web service in the categories can be calculated as follows:

$$QoS(G_iWS_j) = \frac{1}{m} \sum_{j=1}^m W_j \cdot P_j$$

Where,  $m (1 \leq j \leq m, 1 \leq i \leq n)$ .

After evaluating the QoS, the services are ranked in each category and stored in the service repository.

**C. Dynamic Composition of Web Services from the Pool based on the Availability**

In this step the process generator generates all possible combinations of web services from each group that are considered to complete the task (Fig. 2). As a result, based on the availability of services, the maximum possible compositions are prepared by the process generator. Also, the overall QoS of the generated combinations are prepared and the best available composition is selected to complete the task.

G1                      G2                      G3                      Gn

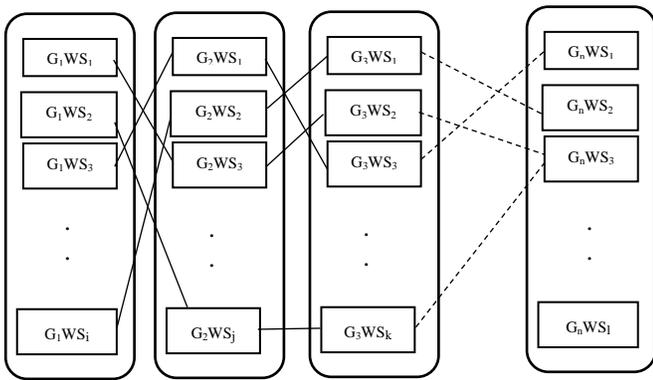


Fig. 2. Generation of composition chains.

Let  $C_1, C_2, C_3 \dots C_n$  are the possible compositions generated by the process generator.

From that, each  $C_i$  may contain the services  $G_1WS_j, G_2WS_j \dots G_nWS_j$ .

Here the  $WS_j$  is selected from each group with best QoS and its availability.

The overall QoS of the composition is:

$$QoS(C_i) = \frac{1}{n} \sum_{m=1}^n QoS(G_i, WS_j)$$

Where  $m (1 \leq i \leq n)$

The process generator algorithm ranks all these compositions as per the evaluated QoS. The best composition is selected from this and executed to complete the task.

#### IV. EXPERIMENTAL RESULTS

The number of services in each category may depend on the non-functional parametric values that satisfy the functionality of the web services. The QoS of the services are evaluated and grouped in specific categories to form the composition.

Table III shows the evaluated QoS of all the services that are in the group  $G_1$ .

TABLE III. EVALUATED QoS OF WEB SERVICES

$G_1WS_1$	$G_1WS_2$	$G_1WS_3$	$G_1WS_4$	$G_1WS_5$
0.338	0.384	0.646	0.502	0.515
$G_1WS_6$	$G_1WS_7$	$G_1WS_8$	$G_1WS_9$	$G_1WS_{10}$
0.707	0.447	0.665	0.706	0.585

TABLE IV. OVERALL QUALITY OF COMPOSITIONS

$C_1$	$C_2$	$C_3$	$C_4$	$C_5$
0.562	0.658	0.455	0.682	0.732
$C_6$	$C_7$	$C_8$	$C_9$	$C_{10}$
0.714	0.566	0.489	0.457	0.791
$C_{11}$	$C_{12}$	$C_{13}$	$C_{14}$	$C_{15}$
0.356	0.671	0.744	0.622	0.693

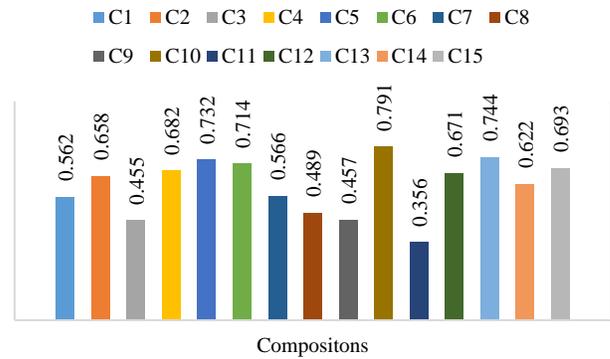


Fig. 3. Performance of composition chains.

These services are ranked as per the QoS and listed in the group. The same process will be followed for all the services that are grouped in the categorized chain. Finally, the process generator develops all the possible compositions  $C_1, C_2 \dots C_n$ . From this, top listed compositions that are up to the expectation form the requester are selected by the evaluator. When the task is invoked the evaluator executes the best composition based on its availability. Table IV lists the overall QoS values evaluated for the compositions.

The performance level of all the combinations generated by the process generator is shown in Fig.3.

#### V. CONCLUSION

The functional requirement of the web services involved in a particular task may vary from service to service based on its performance. Also, it is not sure that all the web services in the composition are at the same level of QoS because it will be the overhead in terms of cost. So appropriate QoS is fixed before composing the web services. Selecting an individual service up to the expectation means attaining the actual functionality of that service. This work presents the method that fix weightage to the non-functional parameters that influence the actual performance of the service. After selecting a pool of such service from each domain they are ranked based on quality. When composing a particular chain, the services are dynamically created based on its availability. Finally, the best composition is executed to complete the task to achieve the maximum performance.

#### ACKNOWLEDGEMENTS

This research project was supported by the Deanship of Scientific Research at Prince Sattam Bin Abdulaziz University, KSA under specialized research project grant no. 2017/01/7561.

#### REFERENCES

- [1] T. Yu "Service Selection Algorithms for Composing Complex Services with Multiple QoS," In Proc of ICSOC, pp. 130-143, 2015.
- [2] L. Zeng, "Quality Driven Web Service Comp.," Proc. of 12th Intl WWW Conference, 2013.
- [3] Yu, and S. R.-Marganec, "Service composition using selective appr. Based on backwards composition context Service", IEEE Intl, pp. 419 – 426, Oct 2009.
- [4] R. Berbner, "QoS-aware Web Services Composition using Heuristics approaches," Intl, pp. 72 – 82, Dec 2006.

- [5] R. Berbner, "Design Arch. for a Quality Service management of Web task oriented Work flow," Networking and Electronic Communication Research Conference (NAEC 2005).
- [6] Jinghai Rao and Xiaomeng Su, A Survey of Automated Web Service Comp. Methods, Semantic Web Services and Web Process Comp., First International Workshop, SWSWPC 2004, San Diego, CA, USA, July 6, 2004
- [7] Ikbel Guindra, "Heuristic Based Time-Aware Service Selection Approach," International Conference on Web Services, Jun 2015, New York, United States. 2015, Web Services (ICWS), 2015 IEEE Intl Conf.
- [8] Xu, S. Reiff-Marganiec, "Immune Algorithm implemented using Empirical Web Task Arrangement," IEEE -Web Services ICWS '08Intl, pp. 238 – 245, Nov 2008.
- [9] M. Alrifai, T. Risse, "Combining Global Optimization with Local Selection for Efficient QoS-aware Service Comp.," Proc of the 18th international Conference on World Wide Web, ACM, 2009.
- [10] Sha MM, Manesh T, Mohamed Mustaq, A Structure for Assessing the Quality of Service and Cost of Web Task Built on Its Well-designed Presentation Intl Journal of Computer, Electrical, Automation, Control and Engineering Vol:10, No:1, 2016.
- [11] C. Zhao, "A Active Web Task Algorithm. Built on the Grouping of Ant Colony algorithm and Genetic procedure," Journal of Comp Information Sys, pp. 2617-2622, 2010.
- [12] Jiuyun Xu Stephan, Reiff-Marganiec, ICWS '08. IEEE Intl Conference on, Proc, pp. 238-245. September 23 - 26, 2008
- [13] W. Zhen-wu, "An Method for Web task built on QoS and Separate Unit Swarm Optimization," Eighth ACIS Intl Conference on Software Engineering, IEEE, 2007.
- [14] Mohemmed Sha, Vivekanandan K. "Selection of Web Services Based on Providers Reputation," International Journal of Innovation Technical Expl Eng. 2013; 3(2):140-143.