CHAPTER – 3

3 QUALITY OF SERVICE-CENTRIC WEB SERVICE COMPOSITION: ASSESSING WEB SERVICE COMPOSITION IMPACT SCALE TOWARDS FAULT PRONENESS

3.1 OVERVIEW

Service composition in the service-oriented architecture is a significant activity. To achieve the quality of service and secured operations from the web service compositions, they need to verify their impact towards fault proneness before deploying that service composition. Henceforth, here in this chapter, we devised a novel statistical approach to assess the service composition impact scale towards fault-proneness. The designed model explores the higher and lower ranges of the service composition impact scale, which is from the earlier compositions that notify as fault-prone. The experimental results examined from the empirical study indicating that the devised model is significant towards estimating the fault-proneness scope of any service composition from selected service descriptors.

3.2 PROLOGUE

Service-Oriented Architecture (SOA) simplifies information technology-related operational tasks by consumption of ready-to-use services. Such SOA found to realize currently in e-commerce domains such as B2B, B2C, C2B, and C2C, in particular, the web services are one that considered serving under this SOA.

Web services are software components with native functionality that can be operable through the web. Another important factor about this web services is that more than one service can compose as one component by coupled loosely. The standard WSDL is web service descriptive language that let the self-exploration of the web services towards their functionality and UDDI is the registry that allows the devised web services to register and available to required functionality [61].

The composition of web services is the loosely interconnected set of Web service operations that act as a single component, which offers solutions for different tasks of an action. Since the task of the composition is integrating various web services explored through different descriptors, it is the most fault-prone activity. The functionality of service composition includes the activities such as (i) identify the tasks involved in each business operation, (ii) trace related web services to fulfill the need of each task, (iii) couple these services by exploring the order of that services usage, which is based on the expected information flow, (iv) and resolve the given operation by ordering the responses of the web services that coupled loosely as one component.

In order to achieve the quality of service and secure transactions in web service composition and usage, the impact of the composition should estimate before deploying those loosely coupled web services as one component.

The Web service compositions used earlier that could find in repositories and the services involved in those compositions helps to assess the impact of these web services towards fault-proneness.

The current composition strategies [62], [63], [64], [65], [57], [59], [22] are error-prone since these State-of-the-art techniques are not mature enough to guarantee the fault-free operations. However, finding these compositions as fault-prone after deployment is functionally costly and not significant towards end level solutions, also may lead to dangerous vulnerable. Hence the process of estimating the composition scope towards fault proneness is mandatory.

In this chapter, we propose a novel statistical approach to estimate the impact scale of a service composition towards fault-proneness. Our method acts as an assessment strategy for any of existing web service composition approaches.

The chapter is structured as follows. Section 3.3 discusses related work. In section 3.4, the proposed statistical approach is explored, which followed by Section 3.5 that contains the results examined from the empirical study. The conclusion of the proposal and future research directions discussed in Section 3.6.

3.3 ASSOCIATED WORK

Service compositions with malfunctioned web services lead to form the highly fault-prone compositions. Henceforth the web service composition to serve as one component under SOA is complex and needs research domain attention to deliver effective strategies towards the QoS centric service. The model devised in [56] defined set of QoS factors to predict available services. Many of existing quality-aware service selection strategies aimed to select the best service among multiple services available. The model devised in [22] considering the linear programming to find the linear combination of availability, successful execution rate, response time, execution cost and reputation, which is regarding find the optimal service composition towards given business operation. The model devised in [57] is considering the temporal validity of the service factors. The authors in [58] modeled a mixed integer linear program that examines both local and global constraints.

The model devised in [59] is selecting services as a complex multi-choice multi-dimension $0/1$ rucksack problem that tends to define different quality levels to the services, which further considered towards service selection. All these solutions are depended strongly on the positive scores given by users to each parameter. However, it is not scalable to establish them in prospective order.

Though the QoS strategies defined are used in service composition the factor fault-proneness of the service composition as usual. Regarding this, a model devised in [60] explored a mechanism for fault proliferation and resurgence in dynamically connected service compositions. Dynamically coupled architecture outcomes in further complexness in need of fault proliferation between service groups of a composition accomplished by not depending on other service groups.

In a gist, it can conclude that almost all of the benchmarking service quality assessment models are attributed specific, user rating specific or both. Hence the importance of attributes is divergent from one composition requirement to other, and contextual factors influence the user ratings, and another important factor is all of these benchmark models are assessing services based on their performance, but in practice, the functionality of one service may influence by the performance of another service. Henceforth here in this chapter, we devised a statistical approach that estimates the impact scale of service composition towards fault proneness, which is based on a devised metric called composition support of service compositions and service descriptors.

3.4 ESTIMATING THE SERVICE COMPOSITION IMPACT SCALE TWORDS FAULT PRONENESS

The said statistical model works in two aspects. First, it determines the impact of each web service descriptor to form a selected malfunctioned service composition. And then it estimates the higher and lower ranges of the impact scale towards fault proneness, which is from the impact of each service descriptor and each malfunctioned service composition. Then these higher and lower ranges of the impact scale will use to assess the impact of a newly composed service composition towards fault-proneness. This strategy leads to estimate the problem of web service descriptor selection. The business solution expected might represent by several compositions but selecting one of these compositions is strictly by their impact towards fault-proneness. The proposed model is optimal in this regard. The detailed exploration of the proposed model is as follow:

The approach of measuring Composition support (cs) metric is proposed in this chapter. Here in the web service composition of cs we consider the bipartite graph to represent the composition weights.

3.4.1 Assumptions

Let set of service-composites $wsc_1, wsc_2, wsc_3, ...wsc_n$, which found to malfunction compositions

Let set of web service descriptions $wsd_1, wsd_2, wsd_3, ..., wsd_n$, which were involved in preparing compositions opted

Hereafter the set of such web-service descriptor sets will refer as $SWSS$

Let two web-service descriptors wsd_i and wsd_j , wsd_i connected with wsd_j , if and only if $(wsd_i, wsd_j) \in wsc_i$

Build an undirected weighted graph UWG with web-service descriptors as vertices and edges between web-services descriptors. An edge between the two web-service descriptors wsd_i, wsd_j will weight as follows,

$foreach\{wsds \forall wsds \in SWSS\}$

$$ew_{(wsd_i \leftrightarrow wsd_j)} = \frac{\sum_{k=1}^{|SWSS|} \{1 \exists [(wsd_i, wsd_j) \subseteq wsc_k \wedge i \neq j]\}}{|SWSS|} \quad \text{(Eq 3.1)}$$

Here in the above (Eq 3.1) $ew_{(wsd_i \leftrightarrow wsd_j)}$ indicates the edge weight between web-service descriptors wsd_i and wsd_j.

In the process of building a weighted graph, we consider that an edge between any two web-service descriptors exists if and only if the edge weight $ew > 0$.

3.4.2 Process

In the method of detecting the composition support of each web-service descriptor with service-composites, initially, we build a bi-parted graph between web service compositions and the set of web-service descriptors.

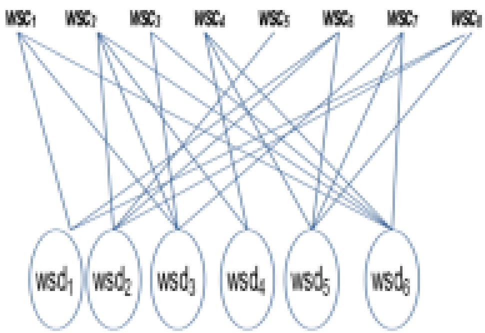

Figure 3.1: bipartite graph between web service compositions and web-service descriptors

If a web-service descriptor wsd_i is part of a web-service composition wsc_i then the weight of the connection between wsd_i and wsc_i will be measured as follows:

$$cw_{(wsd_i \leftrightarrow wsc_j)} = \frac{\sum_{k=1}^{|wsc_i|} \{ew_{(wsd_i \leftrightarrow wsd_k)} \exists [i \neq k \wedge (wsd_i, wsd_k) \in wsc_j\}}{|wsc_j|} \qquad \text{(Eq 3.2)}$$

Here in the (Eq 3.2), we consider the sum of all edge weights from the undirected graph such that there exists an edge between web service descriptor wsd_i and other descriptors of the web service composition wsc_j. The '$|wsc_j|$' indicates the total number of descriptors in web service composition wsc_j.

The graph representation (Figure 3.1) indicates the bipartite relation between web-service descriptors and web service compositions. Composition weights of the different web service compositions represent their importance. Intuitively, a web service composition with high composition weight should contain many of the web-service descriptors with high composition support. The underpinning association of web service compositions and web-service descriptors is that of association between hubs and authorities in the HITS model.

The devised process of identifying web service composition weights using bipartite graph explored below:

Let consider a matrix format of the connection weights of the bipartite edges between web-service descriptors and web-service compositions in given bipartite graph.

The weight of each web service composition as a hub in a bipartite graph initialized as 1, which we represented as a matrix (Table 3.1).

Table 3.1: Initializing the weight of each web service composition as a hub in the bipartite graph with 1 and represented them as a matrix as follows.

1
1
1

1
1
1

As referred to HITS algorithm, find each web service descriptor weight as authority, which it can do as follows:

$$v = A^{'}Xu \quad \text{(Eq 3.3)}$$

Here in the (Eq 3.3) v is the matrix representation of the web service descriptor weights as authorities, $A^{'}$ is the transpose matrix of the matrix A, which is the matrix representation of connection weights between web service compositions as hubs and web service descriptors as authorities in the bipartite graph. Then the actual weights of the web service compositions as hubs can measure as follows in (Eq 3.4):

$$u = AXv \quad \text{(Eq 3.4)}$$

The matrix multiplication between matrix A and matrix v results from the actual weights of the service compositions as hubs.

Then the composition support cs of web-service descriptor wsd can measure as follows in (Eq 3.5):

$$cs_{wsd} = \frac{\sum_{i=1}^{m} \{u_{wsc_i} \exists cw_{wsd \leftrightarrow wsc_i} > 0\}}{\sum_{i=1}^{m} u_{wsc_i}} \quad \text{(Eq 3.5)}$$

And then web service composition impact scale σ towards fault proneness of each service-composition can find as follows in (Eq 3.6):

$$\sigma_{wsc_i} = 1 - \frac{\sum_{j=1}^{m} \{cs_{wsd_j} \exists wsd_j \in wsc_i\}}{|WSD|} \quad \text{(Eq 3.6)}$$

Here in the above equation $|WSD|$ indicates the total number of web-service descriptors involved to create all web service compositions.

Then the web service composition impact scale threshold τ towards fault proneness can measured as follows in (Eq 3.7):

$$\tau = \frac{\sum_{i=1}^{|SWSS|} \sigma_{wsc_i}}{|SWSS|} \quad \text{(Eq 3.7)}$$

Here in the above equation $|SWSS|$ indicates the total number of service-compositions considered

Then the standard deviations of σ each service composition from τ will measure further, which is as follows:

$$sdv_\tau = \sqrt{\frac{\left(\sum_{i=1}^{|SWSS|} \left(\sigma_{wsc_i} - \tau\right)^2\right)}{(|SWSS|-1)}} \quad \text{(Eq 3.8)}$$

Then the Web service composition impact scale low and high ranges towards fault proneness are explored as follows

Lower range of impact scale τ_l is

$$\tau_l = \tau - sdv_\tau$$

Higher range of impact scale τ_h towards fault proneness is

$$\tau_h = \tau + sdv_\tau$$

Service-composite wsc can be said as safe if and only if $\sigma_{wss} < \tau_l$

The impact scale of service composition wsc towards fault proneness is high if and only if

$\sigma_{wsc} \geq \tau_l \;\&\&\; \sigma_{wss} < \tau_h$

The service composition wsc is said to be fault-prone if $\sigma_{wsc} > \tau_h$

3.5 EXPERIMENTAL STUDY

This work explored the credibility of the proposed model on the set of 296 service compositions.

The above said data set contains 296 samples, out of that 240 samples were used to devise the Degree of fault-prone threshold and its upper and lower bounds. Further, we used the rest 56 records to predict the fault-proneness scope. Interestingly, the empirical study delivered promising results. The statistics explored in Table 3.2.

Table 3.2: Statistics of the experiment results

Total Number of web service composites	296
Total number web service descriptors used	140
Total number of edges determined	1560
Total number of bipartite edges found	27776
Service composition impact scale threshold τ towards fault proneness	0.46795646260519363
Higher range of τ	0.5284095974190264
Lower range of τ	0.4075033277913609

Among the considered web service compositions, 240 web service compositions were used to estimate the service composition impact scale towards fault proneness.

Total web service composites used to test the accuracy of the impact scale are 56.

Total number of false negatives are 11 that is web service composites found with σ less than lower bound are 11.

The total number of true positives found is 41, which are having σ higher than the lower bound.

3.5.1 Performance Analysis

We used accuracy estimation (the percentage of valid predictions by the proposed) as the primary performance measure. In addition to measuring accuracy, the precision, recall, and F-measure were used to analyze the performance; these are defined using following equations.

$$pr = \frac{t_{+}}{t_{+} + f_{+}} \quad \text{(Eq 3.9)}$$

Here in above Equation the pr indicates the precision, t_{+} suggests the true positives and f_{+} suggests the false positive.

As per the empirical study conducted the t_{+} found here are 41 and $f_{+}$are 0, henceforth precision is 1.

$$rc = \frac{t_{+}}{t_{+} + f_{-}} \quad \text{(Eq 3.10)}$$

Here in above Equation, the 'rc' indicates the recall, f_{-} indicates the false negative. As per the results explored in empirical study $f_{-}$are 11, hence the rc value is 0.788.

$$F = \frac{2 * pr * rc}{pr + rc} \quad \text{(Eq 3.11)}$$

Here in the above Equation, $'F'$ indicates the F-measure. And the F-measure found from the results of the empirical study is 0.88143

As per the results explored, the proposed model is accurate to the level of 79%. The failure percentage is 21%, which is not negligible but considerably performed well.

3.6 CHAPTER SUMMARY

The model devised in this chapter is a method of estimating web service composition impact scale towards fault-proneness. This approach is a statistical analysis that derives lower and higher range of service composition impact scale towards fault-proneness. Regarding this, initially, an undirected graph that connects the involved web service descriptors as vertices with weighted edges. The edge weight of to vertices is the ratio of service compositions contains services from both descriptors act as vertices to a selected edge. Further, a bipartite graph builds between web service compositions as hubs and web service descriptors used to compose those compositions as authorities. Further hub and authority weights calculated as explored in section 3.4, and further these weights were used to estimate the service composition impact scale towards fault-proneness. The estimated service composition impact scale higher and lower range values can used further to determine the impact of any service composition towards fault-proneness. The empirical analysis conducted on the dataset with 296 different web service compositions. The explored results are indicating the significance of the proposed model. In future to improve the accuracy of the devised model, the correlation of the service descriptors will be estimated, which is done by considering the web-services of each descriptor as the specific value set. Further, web-service reputation can also recognize to evaluate the impact of a service composition towards fault-proneness.