Global Journal of Computer Science and Technology: C
Software & Data Engineering
Volume 14 Issue 9 Version 1.0 Year 2014
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Quality of Service Centric Web Service Composition: Assessing Composition Impact Scale towards Fault Proneness

By Sujatha Varadi & Dr. G. Appa Rao

GITAM University, India

Abstract- Service composition in service oriented architecture is an important activity. In regard to achieve the quality of service and secured activities from the web service compositions, they need to be verified about their impact towards fault proneness before deploying that service composition. Henceforth, here in this paper, we devised a novel statistical approach to assess the service composition impact scale towards fault proneness. The devised model explores the higher and lower ranges of the service composition impact scale, which is from the knowledge of earlier compositions that are notified as fault prone.

Keywords : web service compositions, composition support, service composition impact scale, service descriptor impact scale, web service composition fault proneness.

GJCST-C Classification : H.3.5

QUALITYOFSERVICECENTRICWEBSERVICECOMPOSITIONASSESSINGCOMPOSITIONIMPACTSCALETOWARDSFAULTPRONENESS

Strictly as per the compliance and regulations of:

Quality of Service Centric Web Service Composition: Assessing Composition Impact Scale towards Fault Proneness

Sujatha Varadi [α] & Dr. G. Appa Rao [σ]

Abstract- Service composition in service oriented architecture is an important activity. In regard to achieve the quality of service and secured activities from the web service compositions, they need to be verified about their impact towards fault proneness before deploying that service composition. Henceforth, here in this paper, we devised a novel statistical approach to assess the service composition impact scale towards fault proneness. The devised model explores the higher and lower ranges of the service composition impact scale, which is from the knowledge of earlier compositions that are notified as fault prone. The experimental results explored 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.

Keywords: web service compositions, composition support, service composition impact scale, service descriptor impact scale, web service composition fault proneness.

I. Introduction

Service-Oriented Architecture (SOA) simplifies information technology related operational tasks by consumption of ready-to-use services. Such SOA found to be realized currently in ecommerce 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 web. Another important factor about this web services is that more than one service can be composed as one component by coupled together 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 lets the devised web services to register and available to required functionality [1].

Composition of web services is loosely interconnected set of Web service operations that acts as a single component, which offers solutions for divergent tasks of an operation.

Author α: Assistant Professor, Department of CSE GITAM Schoolof Technology, GITAM University, Hyderabad.
e-mail: varadi.sujatha@gmail.com
Author σ: Professor, Department of CSE GITAM Institute of Technology, Visakhapatnam. e-mail: apparao_999@yahoo.com

Since the task of composition is integrating divergent 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 a given 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 quality of service and secure transactions in web service composition and usage, the impact of the composition should be estimated before deploying those loosely coupled web services as one component.

The Web service compositions used earlier that can be found 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 [2] [3] [4] [5] [6] [7] [8] 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 very expensive and not significant towards end level solutions, also may leads to serious vulnerable. Hence the process of estimating the composition scope towards fault proneness is mandatory.

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

The paper is structured as follows. Section 2 discusses related work. In section 3, the proposed statistical approach is explored, which followed by Section 4 that contains the results explored from empirical study. The conclusion of the proposal and future research directions were discussed in Section 5.

II. Related Work

Service compositions with malfunctioned web services lead to form the highly fault prone

Global Journal of Computer Science and Technology (C) Volume XIV Issue IX Version I Year 2014

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 [9] defined set of QoS factors to predict feasible services. Many of existing quality-aware service selection strategies aimed to select best service among multiple services available. The model devised in [8] considering the linear programming to find the linear combination of availability, successful execution rate, response time, execution cost and reputation, which is in regard to find the optimal service composition towards given business operation. The model devised in. [6] is considering the temporal validity of the service factors. The authors in [10] modeled a mixed integer linear program that considers both local and global constraints.

The model devised in [7] is selecting services as a complex multi-choice multi-dimension rucksack problem that tends to define different quality levels to the services, which further taken into account towards service selection. All these solutions are depends 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 is usual. In regard to this a model devised in [11] 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 be conclude that almost all of the benchmarking service quality assessment models are attribute specific, user rating specific or both. Hence importance of attributes is divergent from one composition requirement to other, and the user ratings are influenced by contextual factors, and another important factor is all of these bench mark models are assessing services based on their individual performance, but in practice the functionality of one service may influenced by the performance of other service. Henceforth here in this paper 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.

III. Estimating the Service Composition Impact Scale Towords Fault Proneness

The said statistical model works in two aspects. First, it estimates 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 o 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 be used 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 represented 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 ( ) metric is proposed in this paper. In regard to measure the composition support, we consider the bipartite graph that represents the composition weights.

a) Assumptions

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

Let set of web service descriptors $wsd_1, wsd_2, wsd_3, ..., wsd_n$, which were involved to form compositions opted

Hereafter the set of such web-service descriptor sets will be referred as

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 will be weighted as follows

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

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

Here in the above equation $ew_{(wsd_i \leftrightarrow wsd_j)}$ 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 $ew > 0$

b) Process

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

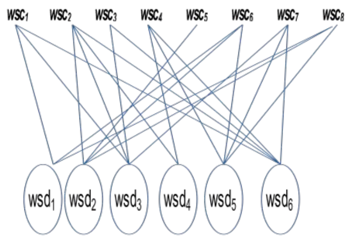

Figure 1: bipartite graph between web service compositions and web-service descriptors

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:

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

Here in the above equation we consider the sum of all edge weights from 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 (fig. 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 [13].

The devised process of identifying web service composition weights using bipartite graph is 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 the each web service composition as a hub in a bipartite graph is initialized as 1, which we represented as matrix (table 1).

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

1
1
1
1
1
1

Let the weights between descriptors and compositions of the given bipartite graph (see fig 1) and form a matrix such that rows represent descriptors (authorities) and columns represent compositions (hubs) and refer that matrix as A,

As referred in HITS [13] algorithm, find each web service descriptor (authority) weight, which is can be done as follows:

$$v = A^{'} X u$$

Here in the above equation 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 bipartite graph. Then the actual weights of the web service compositions (hubs) can be measured as follows:

$$u = A X v$$

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

Then the composition support cs of web-service descriptor wsd can be measured as follows

$$cs_{wsd} = \frac{\sum_{i=1}^{m} \{u_{wsc_i} \exists cw_{wsd \leftrightarrow wsc_i} > 0\}}{\sum_{i=1}^{m} u_{wsc_i}}$$

And then web service composition impact scale towards fault proneness of each service-composition can be found as follows:

$$\sigma_{wsc_i} = 1 - \frac{\sum_{j=1}^{m} \{cs_{wsd_j} \exists wsd_j \in wsc_i\}}{|WSD|}$$

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 be measured as follows:

$$\tau = \frac{\sum_{i=1}^{|SWSS|} \sigma_{wsc_i}}{|SWSS|}$$

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

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

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

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 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 is said to be fault prone if

$$\sigma_{wsc} > \tau_h$$

IV. Empirical Analysisand of the Proposed Model

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

The above said data set contains 294 samples, out of that 250 samples were used to devise the Degree of fault prone threshold and its upper and lower bounds. Further we used the rest 44 records to predict the fault proneness scope. Interestingly, the empirical study delivered promising results. The statistics explored in table 10

Table 10: 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	0.46795646260519363
towards fault proneness	
Higher range of τ	0.5284095974190264
Lower range of τ	0.4075033277913609

Table 2: Exploration of the parameters used in empirical study

Among the considered web service compositions, 244 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

Total number of true positives found is 41, which are having σ greater than lower bound.

a) Performance Analysis

We used accuracy estimation (the percentage of valid predictions by the proposed) as the main 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.

$$pr = \frac{t_+}{t_+ + f_+}$$

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

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

$$rc = \frac{t_+}{t_+ + f_-}$$

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.

$$F = \frac{2 * pr * rc}{pr + rc}$$

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.

V. Conclusion

The model devised in this paper is a method of estimating web service composition impact scale

Global Journal of Computer Science and Technology (C) Volume XIV Issue IX Version I Year 2014

towards fault proneness. This approach is a statistical analysis that derives lower and higher range of service composition impact scale towards fault proneness. In regard to 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 build between web service compositions as hubs and web service descriptors used to compose those compositions as authorities. Further hub and authority weights were calculated as explored in section 3, 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 be used further to estimate the impact of any service composition towards fault proneness. The empirical analysis was conducted on dataset with 296 divergent 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 categorical value set. Further, web-service reputation can also be considered to estimate the impact of a service composition towards fault proneness.

References Références Referencias

1. Papazoglou, M. P., Georgakopoulos, D.: "Service-oriented computing", Communications of the ACM, Vol. 46, No. 10, 2003, pp. 25–28.
2. Aggarwal, R., et al.: "Constraint-driven Web Service Composition in METEOR-S", IEEE Conference on Service Computing, 2004.
3. Lazovik, A., Aiello, M., Papazoglou, M.: "Planning and monitoring the execution of web service requests", International Conference on Service-Oriented Computing, 2003, pp. 335-350.
4. Sirin, E., Hendler, J., Parsia, B.: "Semi-automatic Composition of Web Services Using Semantic Descriptions", In Web Services: Modeling, Architecture and Infrastructure workshop in ICEIS, 2003.
5. Srivastava, B., Koehler, J., "Web Service Composition - Current Solutions and Open Problems", Proceedings of ICAPS Workshop on Planning for Web Services, 2003.
6. Martin-Diaz, O., Ruize-Cortes, A., Duran, A., Muller, C.: "An Approach to Temporal-Aware Procurement of Web Services", International Conference on Service-Oriented Computing, 2005, pp. 170–184.
7. Yu, T., Lin, K.J.: "Service Selection Algorithms for Composing Complex Services with Multiple QoSConstraints", International Conference on Service-Oriented Computing, 2005, pp. 130–143.
8. Zeng, L., Benatallah, B., et al.: "QoS-aware Middleware for Web Services Composition", IEEE Transactions on Software Engineering, Vol. 30, No. 5, 2004, pp. 311–327.
9. Cardoso, J., Sheth, A., Miller, J., Arnold, J., Kochut, K.: "Quality of service for workflows and web service processes", Journal of Web Semantics, Vol. 1, No. 3, 2004, pp. 281–308.
10. Ardagna, D., Pernici, B.: "Global and Local QoS Constraints Guarantee in Web Service Selection," IEEE International Conference on Web Services, 2005, pp. 805–806.
11. Chafl, G., Chandra, S., Kankar, P., Mann, V.: "Handling Faults in Decentralized Orchestration of Composite Web Services", International Conference on Service-Oriented Computing, 2005, pp. 410–423.