Journal of Advances in Science and Technology
Vol. 16, Issue No. 1, March-2019, ISSN 2230-9659

A Novel Approach for Service Composition Using Classification

Anushree Sah[1]* Dr. Manish Varshney[2] Dr. Tanupriya Choudhury[3]

[1] Research Scholar, MUIT, Lucknow, India

[2] Prof. School of Engineering & Technology, MUIT, Lucknow, India

[3] Associate Professor, University of Petroleum and Energy Studies, Dehradun, India

Abstract – Service oriented architecture (SOA) is a collection of services, which can be easily accessed using simple request by using the interaction between two or more services, SOA is usually an architecture, which provides an architecture to interact with services, which had made using other technologies. There are still the challenges in SOA such as Automatic Web service Composition, This research work is based on this very problem.

The process of the work is to find the best match for service so that it can be used to make a complete business process. Further, the framework will be compared using different aspects like execution time, availability, cost, reliability etc. and using different algorithms.

Keywords— Web service composition, SOA, Soft Computing

----------X----------

1. INTRODUCTION

Web services refers to interchange of information among applications over internet. These days internet has become a very big part of our lifestyle, one can consider our social media conversation as a example. Web services help us to achieve the goal of full utilization of internet because it almost the full part of internet which is used. Information technology has been gone through advancement over the decades. Without web services internet will not be of much use or one can say web services without internet is impossible. As it is quite clear that how important web services are for us as an user or an enterprise. In the field of web services evaluation there are few things which still have to be evaluated and automatic web service composition is one of them(Sah, Bhadula, et al., 2018). Service oriented Architecture (SOA) refers to connections between services to form a better solution. SOA is the main component of web services as it provides the blueprint or architecture of web services. SOA is having a big challenge that is known as know Automatic Web service Composition because there is no platform in general available where implementation and testing of algorithms can be performed altogether in a very efficient and organised manner. The requirement of automatic web services composition is necessary in web services. To understand the need Automatic Web service Composition there are many algorithms available to provide the information that how automatically web services can be composed but there is no platform available to implement this or test whether it is as effective(Sah, Dumka, et al., 2018). The main motivation of this research paper to create a framework for web service composition using machine learning.

2. LITERATURE REVIEW

The process of categorizing the web service into different predefined categories using their similarities and feature vectors is web service classification. Managed grouping comprises of two stages— preparation/training and classification. During the preparation stage, a piece of the ordered archive dataset (i.e. services) is taken and in light of which the classifier is prepared to become familiar with the classes. Then, at that point, during the classification stage the classifier concludes at least one classifications for another data with changed exactness levels. As Web Service Description Language (WSDL) are regular language artefact's, there are conditions between the classification of a Web service and its depiction, as far as the phrasing picked by the designer of service and the classifier utilizes this inert data to categorize the service into practically comparable gatherings with specific precision (Sah et al., 2012).

Overall, the class forecast precision of a classifier is reliant upon its inductive inclination's and

speculation error, as various classifiers might pick various examples in the input information to make the classification (Lee et al., 2015). In our work, the classification accuracy is accessed with Naïve Bayes (NB), Support Vector Machine (SVM), K-Nearest Neighbours (KNN), Decision Tree (DT) and Random Forest. We utilized the scikit-learn AI library for Python.

3. PROPOSED METHODOLOGY

"OSC" (Optimal Service Composition) Model - A New Model for Service discovery, selection and composition

A framework can be characterized as a set of abstract classes and their interaction in a well-defined way. Frame working is a concept that comes from object oriented programming. It is a kind of pattern template that defines the interaction between collaborative classes or objects.

Software engineers can reuse systems straightforwardly in their development process; structure reuse is characterized as a combination of information reuse and code reuse (Yu et al., 2015).

This research work presents a novel and progressed model i.e. "OSC" model in which every one of the stages or cycles start in customary design where essential designing and essential particular are displayed in Fig 3.1 below .

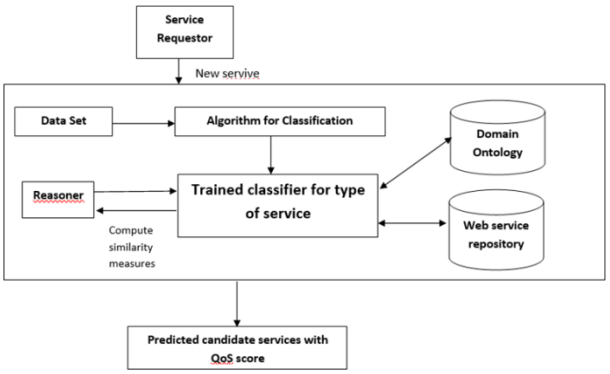

Fig 3.1. Phase 1-Using Classifier for OSC Composition engine

3.1 Overview of model

The proposed framework represents the relationship among discovery, selection and composition tasks of Web services as shown in Fig 3.1. The operation of the framework can be described as follows:

The process of composition starts when a Service Requester demands some service for its functionality from service providers. For this service requestor gives input attributes of service requested, output expected, constraints like Quality of Service for composition of service(Sheng et al., 2014)(Trang et al., 2018). Composition Engine begins to find a bunch of services that give expected output and use QoS constraints asked by requestor from Service Repository. New service once received will first go to composition engine and the composition engine further uses the classifier Decision Tree(DT) for classification(Silic et al., 2015). This is considered after testing different algorithms as shown in result section. For experimental work data set OWL-S Test Collection (OWLS-TC) is used.

Figure 3.1 above shows how the classifier is trained using domain ontology ,reasoned and web service repository and after that the system will produce predicted candidate services with QoS score for the request received from service requestor.

Service Requester can be the system or any individual that sends the service composition request. The functionality requested includes input data, QoS constraints and the desired ouput (Ouni et al., 2017).

A Reasoner utilizes the Domain Ontology information base to figure the likeness proportion of the ideas for semantic matchmaking measure. The Reasoner performs semantic match utilizing ideas of boundaries during the matchmaking system.

The semantic matchmaker is utilized to perform matchmaking between client demand with the arrangement of web services accessible in Web repository(Ben Messaoud et al., 2016). On the off chance that the help is coordinated with the mentioned administration, it passes the outcome to the assistance-separating module, which contains rundown of coordinated services according to the client demand.

Web Services Repository is the archive contains the semantically annotated Web services which is associated with the matchmaking module so the accessible Web services can be found from the archive. QoS data is related with the Web services (Ben Messaoud et al., 2016).

4. RESULTS

For experimental work data set OWL-S Test Collection (OWLS-TC4) is used. The used dataset consist of collection of web service descriptions(WSDL) files from various domains like food, geographu, medical, travel, education, economy etc.

Service vector	Number of features		New service vector
	Before Principal Component Analysis(PCA)	After Principal Component Analysis (PCA)	
SV1	1610	25	SV1A
SV2	1666	20	SV2A

	SV1		SV2	
	Before PCA	After PCA	Before PCA	After PCA
NB	55	59	56	60
SVM	75	78	75	79
KNN	72	75	72	79
DT	78	80	71	82
Random Forests	78	80	72	81

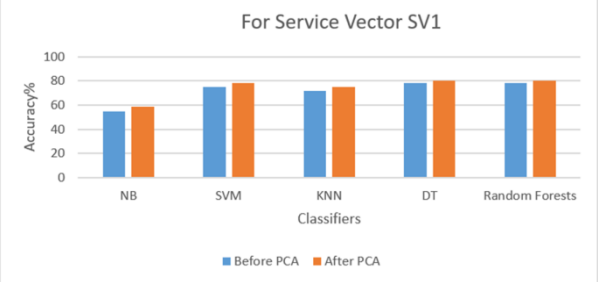

Fig. 4.1. Comparison of result for classification efficiency: Accuracy for Service Vector 1

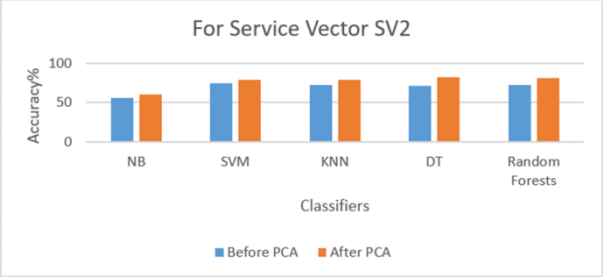

Fig. 4.2. Comparison of result for classification efficiency: Accuracy for Service Vector 2

The outcomes shows that for accuracy% Decision tree classifier shows better outcome. Decision tree classifier had the option to foresee the sort of web service better than other different strategies. This work is first phase. In next stage or in next coming papers consequences of the altered classifiers and composition engine will be discussed and examined. Seeing the above outcomes we can say that Decision tree is predominant and profoundly productive than other classifiers.

CONCLUSION AND FUTURE WORK

Reusing the current programming modules will decrease the development time and cost as the module is not needed to be written from the core. As previous developed modules are reused, it saves the plenty of time of the software engineer that can be used in another work. Moreover, as the reusable parts are pre-tried, the shot at bugs and mistakes is decreased. Accordingly, the expense of investigating and testing is decreased, which are some of the hard and slow aspect of programming. With the increment in programming program size and intricacy, it is a significant worry to diminish complexity and cost, in pace with keeping up with modifiability and unwavering quality. This work suggests an adjusted and novel model. There are a few sub-stages related with each stage that are followed to accomplish quality and dependability in the end result. Worked on quality, effectively maintainable, further developed evolvability, and lower imperfection thickness are the significant advantages of utilizing this model. In future, more effective and complex algorithms can be used to enhance the working of the composition.

REFERENCES

Ben Messaoud, W., Ghedira, K., Ben Halima, Y., & Ben Ghezala, H. (2016). Survey of web service composition. 2015 5th International Conference on Information and Communication Technology and Accessibility, ICTA 2015. https://doi.org/10.1109/ICTA.2015.7426900

Lee, J., Lee, S. J., & Wang, P. F. (2015). A Framework for Composing SOAP, Non-SOAP and Non-Web Services. IEEE Transactions on Services Computing, 8(2), pp. 240–250. https://doi.org/10.1109/TSC.2014.2310213

Ouni, A., Kessentini, M., Inoue, K., & Cinnéide, M. (2017). Search-Based Web Service Antipatterns Detection. IEEE Transactions on Services Computing, 10(4), pp. 603–617. https://doi.org/10.1109/TSC.2015.2502595

Sah, A., Bhadula, S. J., Dumka, A., & Rawat, S. (2018). A software engineering perspective for development of enterprise applications. Handbook of Research on Contemporary Perspectives on Web-Based Systems, pp. 1–23. https://doi.org/10.4018/978-1-5225-5384-7.ch001

Sah, A., Dumka, A., & Rawat, S. (2018). Web technology systems integration using SOA and web services. Handbook of Research on Contemporary Perspectives on Web-Based Systems, pp. 24–45. https://doi.org/10.4018/978-1-5225-5384-7.ch002

Sah, A., Rawat, S., & Pundir, S. (2012). Design, Implementation and Integration of Heterogeneous Applications. International Journal of Computer Applications, 54(5), 11–16. https://doi.org/10.5120/8561-2157

Sheng, Q. Z., Qiao, X., Vasilakos, A. V., Szabo, C., Bourne, S., & Xu, X. (2014). Web services composition: A decade's overview. Information Sciences, 280, pp. 218–238. https://doi.org/10.1016/j.ins.2014.04.054

Silic, M., Delac, G., & Srbljic, S. (2015). Prediction of atomic web services reliability for QoS-aware recommendation. IEEE Transactions

on Services Computing, 8(3), pp. 425–438. https://doi.org/10.1109/TSC.2014.2346492

Trang, M. X., Murakami, Y., & Ishida, T. (2018). Policy-aware service composition: Predicting parallel execution performance of composite services. IEEE Transactions on Services Computing, 11(4), pp. 602–615. https://doi.org/10.1109/TSC.2015.2467330

Yu, Q., Chen, L., Li, B., & Li, J. (2015). Ant colony optimization applied to web service compositions in cloud computing. Computers and Electrical Engineering, 41(C), pp. 18–27. https://doi.org/10.1016/j.compeleceng.2014.12.004

Corresponding Author

Anushree Sah*

Research Scholar, MUIT, Lucknow, India