Proficient Similarity Measures For Effective Context Based Web Services

P. Anusha, M.Tech #1, A. Padma Sai Vivek #2, T. Vyshnavi #3, E. Sravani #4, K. Sai Kumar #5, V. Mahendra #6

#1 Assistant professor, Dept Of CSE, Qis College of Engineering and Technology, Ongole, Prakasam (Dt)
#2 Student, Dept Of CSE, Qis College of Engineering and Technology, Ongole, Prakasam (Dt)
#3 Student, Dept Of CSE, Qis College of Engineering and Technology, Ongole, Prakasam (Dt)
#4 Student, Dept Of CSE, Qis College of Engineering and Technology, Ongole, Prakasam (Dt)
#5 Student, Dept Of CSE, Qis College of Engineering and Technology, Ongole, Prakasam (Dt)
#6 Student, Dept Of CSE, Qis College of Engineering and Technology, Ongole, Prakasam (Dt)

Abstract: A Web service composition is an interconnected set of multiple specialized Web service operations, which complement each other to offer an improved tool capable of solving more complex problems. Manual design and implementation of Web service compositions are among the most difficult and error prone tasks. To face this complexity and to reduce errors at design time, the developer can alternatively search and reuse existing compositions that have solved similar problems. Thus the problem of designing and implementing Web service compositions can be reduced to the problem of finding and selecting the composition closest to an initial specification. Comparison of Web service compositions can be done using two possible sources: composition designs (models), and execution logs of compositions. In particular, in this paper a set of similarity measures are described for Web service composition models. The main objective is to measure and assess the degree of closeness between two given compositions of Web services regardless of their modeling language. In this paper, we propose a novel Web services similarity measure approach based on the notion of service composition context. Specifically, we first introduce three types of parameter correlations between service input and output parameters. These correlations can be obtained from existing services compositions. Based on parameter correlations, we propose the service composition context model. Through the composition context of a service, the composition context network is constructed using contexts of all services. Then, we propose to measure the similarity between any two services using the PersonalRank and SimRank++ algorithm by taking the obtained context network as input. By experiments, we analyze characteristics of our proposed method, and demonstrate that its accuracy is much better than the state-of-the-art approaches.

Keywords: Web services, parameter correlation, composition context, services similarity.

I. INTRODUCTION

A Web service composition is an interconnected set of multiple specialized Web service operations, which complement each other to offer an improved tool capable of solving more complex problems that go beyond each individual service capability. Manually designing and implementing Web service compositions [1] is among the most difficult and error prone tasks that any application developer may face. Based on a given initial specification of a complex problem, the common steps that the composition developer must follow are: identify specific sub-problems derived from the complex problem [2]; look for software components (in the form of Web services) that can solve each sub-problem; design the information flow and execution flow for the overall Web service composition; and finally, collect all responses and integrate them into a global composed response to the client. In order to face this complexity and to reduce the design time for Web service compositions, the composition developer could alternatively search and reuse existing compositions that have solved similar problems. Such Web service compositions can be found in repositories of Web service descriptions which publish simple and compound services. Thus the problem of designing and implementing Web service compositions can be reduced to the problem of finding and selecting the composition closest to the initial specification [3]. To achieve this goal, there is the need to use similarity measures to determine how close a given composition with respect to the initial specification is. Similarity measures applied to Web services [5] are not a new subject, as they have been studied and addressed long ago, this is because application developers have faced many times, the problem of searching and selecting simple Web services to meet their specific needs. However, to date little progress has
been made in relation to the construction of public repositories of Web service compositions that provide proven solutions to common problem specifications. Also, little progress has been published in relation to the complex tasks of searching, selecting and matchmaking composed Web services. Regarding the comparison of Web service compositions, this task can be done using two possible resources: the composition designs (models), and the execution logs of compositions. Both source options pose different challenges and difficulties mainly because of the representation format and techniques [6] required to extract information. In case of comparing composition designs or models, Web service compositions could have been modeled, described or implemented in some of the following languages: BPEL4WS, BPMN, EPC , YAWL, WS-CDL among others. In case of comparing execution logs of Web service compositions, it is necessary to mine server logs in order to extract and analyze the sequences of messages issued during execution at the hosting Web server.

II RELATED WORK

In the past decade, there have been a lot of research works focusing on Web services similarity measure. These works mainly use static descriptions of services, such as operations, messages, and descriptions. In general, there are three types of works for Web services similarity measure based on static descriptions of services. First, there are many works to measure services similarity based on services’ static syntactic description, such as operations, input and output messages, textual description of its function, such as [7]-[9]. In addition, some works consider the structure of input and output messages of services [10]-[12]. Second, many works, such as [13]-[15], use the ontology-based semantic technology to annotate Web services, such as OWL-S and WSMO [16]. Then, services similarity is measured using the similarity of semantic concepts. Third, some works combine above two approaches to measure services similarity. That is, both the syntactic and semantic descriptions of services are taken into account to measure the similarity of services [17]-[19]. However, there still exist some problems for Web services similarity measure only taking static service descriptions into account. First, as mentioned in the first part, two services may not be able replace each other even if their static descriptions are similar. Second, the fundamental ontology suitable for all Web services is hard to construct under the environment of Internet. As a consequence, it is hard to annotate Web services with ontology [13], [20]. In reality, most services are not annotated with ontology, which limits practical applications of approaches using semantics.

Different from above methods, some researchers believe that inherent essential behaviors of services need be taken into account to measure services similarity. Therefore, they propose to measure services similarity based on dynamic behaviors of services [21]-[24]. These works mainly use theoretical or formal tools such as process algebra and calculus to consider dynamic behaviors of services, and measure services similarity from the perspective of Web services behaviors consistency. These methods improve the accuracy of services similarity measure by using complete formal tools. However, these methods need to model Web services and their behaviors with formal tools beforehand. Furthermore, the verification complexity is very high. In practical applications with many Web services, the availability of these methods is very limited. Meanwhile, it is difficult to achieve full automation of verification [25].

There are some research works, in which core ideas are similar to our work. References [16]-[18] propose services similarity measure based on composition contexts. Through the context of a Web service in a composition, services with similar composition context can be found from existing compositions. From the view of a context model, the composition context of a service in these works focus on services that have control flow dependencies. Specifically, for a service, its composition context can be modeled as a directed graph. In the graph, a vertex represents a service that have control flow dependency with it, and an directed edge represents a control flow between two services, which is an ordered sequence of multiple atomic patterns (Sequence, AND-fork, AND-join, OR-fork, XOR-fork, OR-join). To calculate the similarity of two service contexts, both of them are regarded as the starting of the directed graph of their contexts, respectively. Then, through same neighbor services having equal path length to these two services, the distance between two control flow with the same starting and ending services is calculated through the editing distance.
III METHODOLOGY

The main contributions are as follows. (1) We propose a service composition context model based on correlations between a service input parameter and output parameter. For a service, its composition context is comprised of its neighbor services, which have parameters correlations with it in existing compositions. (2) We construct the Web service context network by all services and their composition contexts, and propose to calculate the similarity between any two services in the composition context network. (3) Through a group of experiments, we analyze characteristics of the proposed method, and compare it with related approaches. The system explores a practical method to measure the similarity between Web services from the perspective of dynamic characteristics of services, which is different from existing works based on static syntactic or semantic descriptions of Web services.

Specifically, we take a Web service’s composition context as the basis of services similarity measure in this work. For a service (an operation of a Web service is called a service in this paper), we regard its neighbors and interactive information with other services in existing compositions as its composition context. We extract the composition context of each service from existing compositions (including a service composition or a service Mashup, hereinafter referred to as a composition), and propose a method to compute the similarity between two Web services based on service composition contexts. The main contributions are as follows. The system proposes a service composition context model based on correlations between a service input parameter and output parameter. For a service, its composition context is comprised of its neighbor services, which have parameters correlations with it in existing compositions. The systems constructed the Web service context network by all services and their composition contexts, and propose to calculate the similarity between any two services in the composition context network.

Through a group of experiments, we analyze characteristics of the proposed method, and compare it with related approaches.

IV ARCHITECTURE

Fig: System Architecture

COMPONENTS

Admin Server
In this module, the Admin has to login by using valid user name and password. After login successful he can do some operations such as View all users and authorize, View all publishers and authorize, List All Posts with ranks, View all entity details, View all users searched keyword with keyword score, View all entity details in tree format, View all entity details in Knowledge Graph, View all similarity Metrics, View Entity Scores in chart.

View and Authorize Users
In this module, the admin can view the list of users who all registered. In this, the admin can view the user’s details such as, user name, email, address and admin authorizes the users.

View Chart Results
View Entity Scores in chart.

Publisher
In this module, there are n numbers of users are present. Publisher should register before doing any operations. Once publisher registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful publisher will do some operations like View
Profiles, Upload contents, View all Your Published Details.

**End User**
In this module, there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user will do some operations like View your profiles, Search entity, View all search history, Search top k entity topic with its details, View all your searched keyword with keyword score.

**V CONCLUSION**
In this paper, we propose a practical approach to measure Web services similarity from the perspective of dynamic features of services, which is different from traditional methods based on static descriptions of Web services. Dynamic interaction information of services, which contains neighbor services and correlations between their input and output parameters in existing compositions, are taken into account to measure services similarity. On the basis of this idea, we first propose a Web service composition context model based on parameter correlations. According to the context model, we build the composition context of each service from existing service compositions, and construct a global context network of all services. Using the context networks, we measure the similarity of any two services by the PersonalRank and SimRank++ algorithms. The proposed approach makes use of dynamic service composition contexts. However, if some services published on the Web have not been composed or invoked, we cannot get their composition records. Under such circumstances, we cannot use the proposed method to measure the similarity between these services and other services, whereas methods using syntactic and semantic descriptions of services can still work. This problem can be regarded as the "cold start" problem of our approach. In order to solve this problem and measure Web services similarity in a more comprehensive and accurate manner, we should combine our approach with traditional methods to study Web services similarity measure combining static and dynamic characteristics of services. This is one of our works in the future. In addition, the proposed service parameter correlation is the one-to-one correlation between service parameters. In order to deal with the heterogeneity of service parameters in the future, we should deal with more complex correlations, such as one-to-many or many-to-many correlations between service parameters.

**VI REFERENCES**


**Authors Profile**

**P. Anusha, M. Tech** working as an Assistant Professor of CSE Department in QIS College of Engineering and Technology (Autonomous), Ongole, Andhra Pradesh, India.

**A. Padma Sai Vivek** pursuing B Tech in computer science engineering from Qis college of Engineering and Technology (Autonomous & NAAC ‘A’ Grade), Ponduru Road, Vengamukkalapalem, Ongole, Prakasam Dist, Affiliated to Jawaharlal Nehru Technological University, Kakinada in 2016-20 respectively.

**T. Vyshnavi** pursuing B Tech in computer science engineering from Qis college of Engineering and Technology (Autonomous & NAAC ‘A’ Grade), Ponduru Road, Vengamukkalapalem, Ongole, Prakasam Dist, Affiliated to Jawaharlal Nehru Technological University, Kakinada in 2016-20 respectively.
E. Sravani pursuing B Tech in computer science engineering from Qis college of Engineering and Technology (Autonomous & NAAC ‘A’ Grade), Ponduru Road, Vengamukkalapalem, Ongole, Prakasam Dist, Affiliated to Jawaharlal Nehru Technological University, Kakinada in 2016-20 respectively.

K. Sai Kumar pursuing B Tech in computer science engineering from Qis college of Engineering and Technology (Autonomous & NAAC ‘A’ Grade), Ponduru Road, Vengamukkalapalem, Ongole, Prakasam Dist, Affiliated to Jawaharlal Nehru Technological University, Kakinada in 2016-20 respectively.

V. Mahendra pursuing B Tech in computer science engineering from Qis college of Engineering and Technology (Autonomous & NAAC ‘A’ Grade), Ponduru Road, Vengamukkalapalem, Ongole, Prakasam Dist, Affiliated to Jawaharlal Nehru Technological University, Kakinada in 2016-20 respectively.