Web Service Composition Techniques: A Comparative Study

[1] Sridhar Dilip Sondur [2] Abhilash M [3] Vishwanatha A [4] Dr. G N Srinivasan
[1][2][3][4] Department of ISE, R.V. College of Engineering [RVCE], Bangalore, India
[1] shridharsondur@gmail.com [2] abhi_501@hotmail.com [3] vishwanathchp@gmail.com
[4] gnsrinivasan@rvce.edu.in

Abstract: **Web service composition is combining various individual services into a single service which provides more reliable, value-added features than individual service cannot provide. Many individual web services are available over the internet which provides different services. But, there may be a situation when there is no unique service which completely fulfills user's requirements. Web service composition solves this problem by combining services to satisfy the user's requirements. There are many web service composition techniques but, which one is the efficient one is important. This paper provides an overview on web service composition with its issues. This paper also provides several different web service composition techniques that help in solving these issues. Web Service composition mainly takes into consideration the input and the output parameters along with user's requirements. Since different individual services perform different task by taking different parameters it becomes difficult to compose services. So, to understand which approach is best, various techniques are justified by considering different parameters for evaluation.**

Index Terms**—web service composition, dynamic web service composition, QoS, graph search-based service composition.**

I. INTRODUCTION

Individual web services can be used for providing services to the end user, but the true power of web services technologies is leveraged through composition of web services. Web services can be composed to achieve Enterprise Application Integration (EAI) within one company or to achieve business-to-business (B2B) integration of heterogeneous software and computing systems of different companies. The composed Web Services can be distributed over a network, running on different platforms, implemented in different programming languages, and provided by different service providers. Therefore, Web Services Composition has gained a lot of attention as the composite service presents features that an individual service cannot. The researchers are interested in dynamic service composition to make services reusable and provide progressive adaption of web services. The Semantic Web Service is improvement on the current web services in which the information is given a well-defined meaning, enabling the clients and computers to work in co-ordination [2].

II. PRELIMINARIES

Web Service:

Web services are business applications that have open, internet-oriented, standards-based interfaces. Web services are nothing more than a request/response mechanism that allows the client to simply access/modify the data. These services process different type of requests that can range from answering simple requests to executing important business processes. Different technologies are used to define, describe and discover the web services over the internet such as SOAP (Simple Object Access Protocol) for communication using HTTP protocol, WSDL (Web Service Description Language) used for specification of messages that are used for communication between service provider and requesters, UDDI (Universal Description, Discovery and Integration) for searching existing web services [1].

As shown in Fig. 1, the service provider will create and provide the web service. This service offered by service provider is published in the Service Registry. The service registry acts as a repository for different services providing details about the services present at the registry.

Web Services Composition:

Web Service Composition is a way grouping of individual web services into a composite service for creating high-level business process. When composing web services, the business logic of the client is implemented by many services. As a result, the complexity of the services increases. Web service composition is required because usually there is no single service which can fulfil the user goal completely.

Web Service Composition are of two types, Static or Dynamic. Static Web service composition takes place during design-time when the architecture and the design of the software system are planned. The components to be used are selected, joined together and finally compiled and deployed. This may work well as long as the environment of web services and business partners and Web services components do not or only rarely change. Two main approaches are currently investigated for static service composition. The first approach is Web service orchestration, binds available Web services by adding a coordinator (the orchestrator) that is responsible for invoking and binding the single sub-activities. The second approach, referred to as web service choreography, does not assume the use of a central coordinator but rather defines complex tasks via the definition of the conversation that should be undertaken by each participant. Following this approach, the overall activity is achieved as the composition of peer-to-peer interactions among the collaborating services [2].

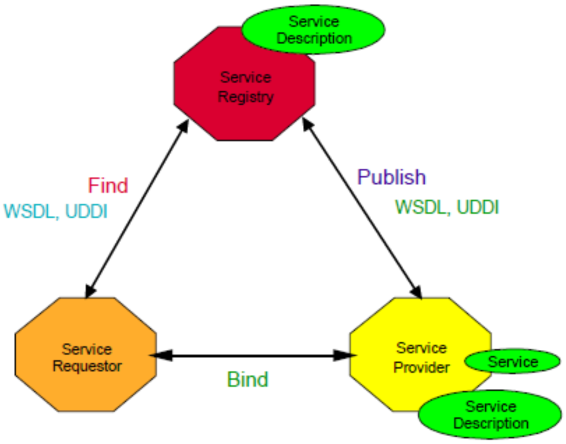
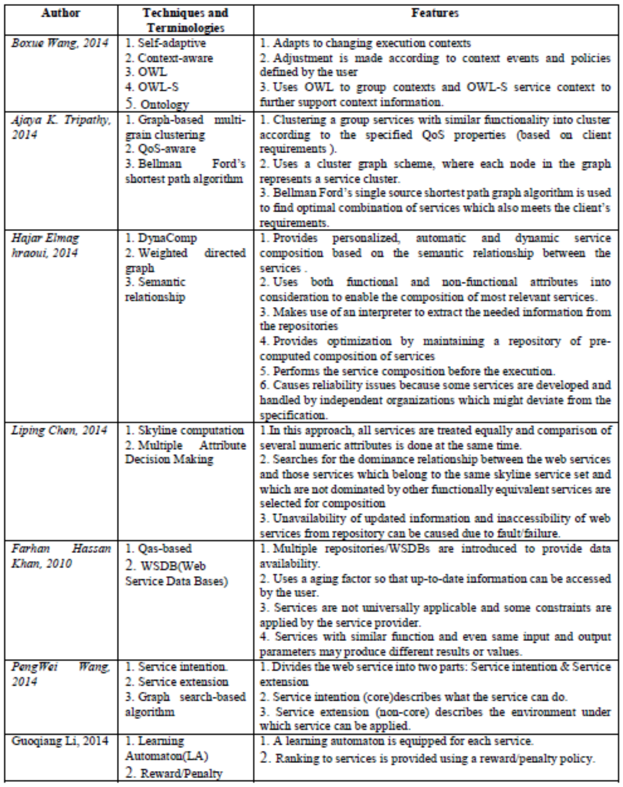


Fig. 1 The web service model

III. SERVICE COMPOSITION APPROACHES

Boxue Wang et. al., [3] proposed a system which is self-adaptive and context-aware that can adapt to changing execution contexts and making adjustment according to context events and policies defined by the user, because web service compositions are executed in many contexts in different environments where the contexts can change which might affect the execution of services. This approach helps the service providers majorly because they don't have to be worry about the context handling. The proposed system uses OWL to group contexts and OWL-S service context to further support context information. The system makes use of ontologies where it groups the services into different types of services according to their input and output parameters' ontologies which helps in simplifying the service management. Also, the system combines different services to create a composite service which meets the users' needs. And, since the contexts of the service execution can change due to any known or unknown reason, it can adapt to changing contexts and adjust by making appropriate changes according to the context events. *Ajaya K. Tripathy et. al.* [4] has proposed a graph-based multi-grain clustering and selection model for service composition where in the discovery of the service also considers the end users' requirements for service selection. A lot of research is being carried out in this field known as QoS-aware (Quality of Service-aware) service composition, using which the service compositions are generated effectively which satisfy the QoS constraints and optimize on that, because each client will have their own QoS criteria. The proposed model introduced makes use of clustering to group services with similar functionality into cluster according to the specified QoS properties. This clustering will help in optimizing the time during the selection of service. This is done using cluster graph scheme, where each node in the graph represents a service cluster. The graph scheme is based on Bellman Ford's single source shortest path graph algorithm to find optimal combination of services which also meets the client's requirements. As an example to this approach, *Hajar Elmaghraoui et. al.* [7] provided a framework called Dyna Comp, a framework in the context of e-government. The framework takes into consideration both the service providers and the users. E-government makes use of information technology to improve the efficiency of the services provided by the government and also the information provided to the citizens, businesses by the government services. The framework proposed provides personalised, automatic and dynamic service composition based on the semantic relationship between the services with the help of a weighted directed graph which will be run before the composition. It uses both functional and non-functional attributes into consideration to enable the composition of most relevant services. In the context of e-government, there are several departments that has its own repositories and they use their own methods to describe the services, which is a problem of interoperability. This problem is solved by DynaComp framework which makes use of an interpreter to extract the needed information from the repositories independent of methods used to describe the services. Dynamic service composition an expensive process, the DynaComp framework provides optimization by maintaining a repository of pre-computed composition of services i.e., performs the service composition before the execution. Even after the consideration of users' requirements when discovering services for composition, there can be reliability issues because some services are developed and handled by independent organisations which might deviate from the specification. To solve this problem *Liping Chen* [6] proposed a method which checks reliability of services and makes sure that the best QoS combinations of services are selected for service composition. Since a huge amount of web services which makes sure the correctness of the web service composition are distributed all over the world the consistent execution of services is not guaranteed. Therefore, to ensure reliability and consistency, the proposed method makes use of skyline computation which basically treats all services equally and compares several numeric attributes at the same time. First, conformance of each participant's local behavior to global interaction is proposed and then dominance relationship between the web services is searched and valid candidate services are selected

Table 1: Summarizing different service composition techniques



Author	Techniques and Terminologies	Features
Boxue Wang, 2014	1. Self-adaptive 2. Context-aware 3. OWL 4. OWL-S 5. Ontology	1. Adapts to changing execution contexts 2. Adjustment is made according to context events and policies defined by the user 3. Uses OWL to group contexts and OWL-S service context to further support context information.
Ajaya K. Tripathy, 2014	1. Graph-based multi-grain clustering 2. QoS-aware 3. Bellman Ford's shortest path algorithm	1. Clustering a group services with similar functionality into cluster according to the specified QoS properties (based on client requirements). 2. Uses a cluster graph scheme, where each node in the graph represents a service cluster. 3. Bellman Ford's single source shortest path graph algorithm is used to find optimal combination of services which also meets the client's requirements.
Hajar Elmag hraoui, 2014	1. DynaComp 2. Weighted directed graph 3. Semantic relationship	1. Provides personalized, automatic and dynamic service composition based on the semantic relationship between the services . 2. Uses both functional and non-functional attributes into consideration to enable the composition of most relevant services. 3. Makes use of an interpreter to extract the needed information from the repositories 4. Provides optimization by maintaining a repository of pre-computed composition of services 5. Performs the service composition before the execution. 6. Causes reliability issues because some services are developed and handled by independent organizations which might deviate from the specification.
Liping Chen, 2014	1. Skyline computation 2. Multiple Attribute Decision Making	1.In this approach, all services are treated equally and comparison of several numeric attributes is done at the same time. 2. Searches for the dominance relationship between the web services and those services which belong to the same skyline service set and which are not dominated by other functionally equivalent services are selected for composition 3. Unavailability of updated information and inaccessibility of web services from repository can be caused due to fault/failure.
Farhan Hassan Khan, 2010	1. Qas-based 2. WSDB(Web Service Data Bases)	1. Multiple repositories/WSDBs are introduced to provide data availability. 2. Uses a aging factor so that up-to-date information can be accessed by the user. 3. Services are not universally applicable and some constraints are applied by the service provider. 4. Services with similar function and even same input and output parameters may produce different results or values.
PengWei Wang, 2014	1. Service intention. 2. Service extension 3. Graph search-based algorithm	1. Divides the web service into two parts: Service intention & Service extension 2. Service intention (core)describes what the service can do. 3. Service extension (non-core) describes the environment under which service can be applied.
Guoqiang Li, 2014	1. Learning Automaton(LA) 2. Reward/Penalty	1. A learning automaton is equipped for each service. 2. Ranking to services is provided using a reward/penalty policy.

For composition which belongs to the same skyline service set and are not dominated by other functionally equivalent services. Finally, Multiple Attribute Decision Making approach is used making optimal QoS-based service selection. Even after meeting the QoS criteria for service composition, there can be problems related to unavailability of updated information and also inaccessibility of web services from repository/databases because of fault/failure. *Farhan Hassan Khan et. al.* [11] proposed a QoS-based framework which helps in solving these problems. The proposed framework provides updated information by adding aging factor to the repository/WSDBs (Web Service Data Bases) and problem of inaccessibility is solved by replicating the repository. Multiple repositories/WSDBs are introduced in order to make system more reliable and provide data availability. By using aging factor the up-to-date information can be accessed by the user. The above proposed approaches dealt with the service composition based on input and output parameters along with users' requirements (i.e, QoS criteria), but most services are not universally applicable and some restrictions are applied by the service provider. These restrictions are nothing but constraints that are imposed on these services. The inputs and the output and users' requirements are major factors in selection on services for composition but the values of these parameters can be different which mainly some contextual conditions are. Because of this, different services may be selected for composition. But sometimes some services with similar function and even same input and output parameters may produce different results or values. *Guoqiang Li et. al.* [8], proposed an adaptive Learning Automaton (LA) based framework for web service composition. Leaning Automaton refers to a finite state machine where interactions happen with the environment and subsequently learn the optimal action required to complete the task.

In this approach, an automaton is equipped for each service to monitor the service. And a reward-penalty scheme is used to update the usefulness (ranking) of the component services in real time. The framework is basically divided into three components, (i) Automaton, (ii) Environment (iii) Reward/Penalty. The environment takes inputs from the automaton actions. The automaton takes inputs from the responses from the environment. Depending on the success or failure in execution of the service in real time, the service we will either rewarded or penalized. *PengWei Wang et. al.* [5] defined and adapted a formal expression to use constraint based service composition. The proposed approach first divides the web service into two parts: service intension, which deals with core of the service i.e., describing what the service can do and service extension, deals with noncore concepts such as environment under which service can be applied and QoS information and a graph search-based algorithm is used along with two pre-processing methods, (i) Prepackaging method, (ii) abstraction and refinement method

The approach is as follows:

- ❖ Conditions that must be met for correct execution are specifies i.e., service constraints.
- ❖ Considering these constraints, the proposed solution can be applied for service composition.
- ❖ The graph search-based algorithm to generate feasible solutions with according to the user's criteria.

IV. CONCLUSION

In this paper, we studied various papers which provide different approaches and solutions to web service composition. The main focus was on dynamic web service composition techniques which makes use of different graph search-based algorithms. The limitations were mentioned for each methods, also frameworks to overcome them were highlighted. The approaches are mainly based on QoS aware, Self-adaptive, Context-aware, Semantic relationships, functional / non-functional , automaton and Graph search-based algorithms. We also discussed briefly, the comparison of the different service composition techniques that are proposed. Our future work is to perform a experimental analysis on these approaches and evaluate the optimal method for QoS based service composition.

REFERENCES

[1] Monowar Hussain and Arnab Paul, "A Survey on Graph based Web Service Discovery and Composition Techniques" IJEAT, Volume-3, Issue-5, June 2014

[2] Yashpalsinh Jadeja, Kirit Modi, and Ankur Goswami , "Context Based Dynamic Web Services Composition Approaches: a Comparative Study" International Journal of Information and Education Technology, Vol. 2, No. 2, April 2012.

[3] Boxue Wang, Xinhuai Tang, "Designing a Self-adaptive and Context-aware Service Composition System", IEEE, 2014.

[4] Ajaya K. Tripathy, Manas R. Patra, Mohiuddin A. Khan, Huda Fatima, Pravati Swain, "DynamicWeb Service Composition with QoS Clustering", IEEE International Conference on Web Services, 2014.

[5] PengWei Wang, ZhiJun Ding, ChangJun Jiang, and MengChu Zhou, "Constraint-Aware Approach to Web Service Composition", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, VOL. 44, NO. 6, JUNE 2014.

[6] Liping Chen, "Ensuring Reliability and QoS Optimizing for Web Service Composition", Tenth International Conference on Computational Intelligence and Security, 2014.

[7] Hajar Elmaghraoui, Laila Benhlima, Dalila Chiadmi, "DynaComp: a Framework for dynamic composition of semantic web services", 2014 IEEE.

[8] Guoqiang Li1,2, Dandan Song1, Lejian Liao1, "Learning Automata-based Adaptive Web Services Composition", 2014 IEEE.
[9] Yulong liu, Lei Wu, Shijun Liu, "A Novel QoS-Aware service composition approach on path decomposition", 2012 IEEE Asia-Pacific Services Computing Conference.

[10] Haibo Zhao and Prashant Doshi, "Towards Automated RESTful Web Service Composition", 2009 IEEE International Conference on Web Services.

[11] Farhan Hassan Khan, M.Younus Javed, Saba Bashir, Aihab Khan, Malik Sikandar Hayat Khiyal, "QoS Based Dynamic Web Services Composition & Execution ", (IJCSIS) International Journal of Computer Science and Information Security, Vol. 7, No. 2, February 2010.