

# An Overview of Web Services Composition Approaches

A.Vishnu Vardhan  
Dept of CSE  
Pondicherry University  
Pondicherry, India

M.S.Saleem Basha  
Asst.Professor, Dept. of CSE  
Pondicherry University  
Pondicherry, India

P.Dhavachelvan  
Professor, Dept. of CSE  
Pondicherry University  
Pondicherry, India

## ABSTRACT

This paper presents the brief survey about the existing approaches in web services composition. The main research areas in web services are related to security, quality of service and composition. Among all these areas, web services composition turns out to be a challenging one, because it supports business-to-business or enterprise application integration. It provides an effective solution to complex web application. In recent days with the emergence of semantic web the scope for semantic based web services composition increases as it provides better results compared to the traditional method of discovering candidate services for composition. Along with the semantics the nature of composition .Also needs to be dynamic as the web services and its parameters are changing frequently.

**Keywords** B2B, Semantic web, QoS, EAI.

## 1. INTRODUCTION

Web services are considered as self-contained, self-describing modular applications that can be published, located and invoked across the web [4]. There are mainly two reasons for switching from middleware technologies to web services those are (i) involves whole learning curve and (ii) they don't adopt standard rules and specifications. Another important characteristic for web services are they are loosely coupled.

Due to the dynamic nature of the web, and rapid development across the internet a large number of web services are emerged in the present day internet [2]. However a single web service i.e. published on the web can not satisfy single user request. The increasing number of web services facilitates not only new technology but also poses new challenges on how to compose or collaborate.

As the core technology web service composition provides an effective solution to complex web application. Challenges related to web services composition include constant changes in business rules, high diversity and heterogeneity of web services [7]. Still the research is going on finding the appropriate services from a set of candidate services, building of compound services, invocation of atomic services and execution of compound services. The description of web service is lack of semantic information due to this web service composition is lack of uncertainty .Due to this reason research is towards semantic web.

Semantic Web [3] is the crucial step for Web services composition. The functionality of a Web service needs to be described with further information, either by a semantic explanation of what it does or by a functional annotation of how it behaves [6]. The semantic Web is also an expansion of the current Web in which information is given well defined meaning, as a result better enabling computer and human to work in cooperation .Semantic Web aims to add machine-interpretable information to Web content in order to provide intelligent access.

The Semantic Web has the potential to ease and possibly eliminate this problem by concerning the data exchanged to a set of ontologies which identify the abstract framework that helps with the elucidation of the data. Using these ontologies Web services map the information that they obtain to known concepts and then use that mapping to derive cost of the message.

## 2. BACKGROUND WORK

Before going to the methods present in web services composition, we have to refresh ourselves about the web services frame work and the components that are involved in general composition framework.The basic terminologies used in web services are:

- UDDI-for discovery
- WSDL-for description
- SOAP-for communication
- XML-Uniform data representation

Web service architecture is based on the interaction between the three roles, Service provider, service requester and service broker

### *Web services Framework*

The XML code of behavior is the foundation for a Web Service framework within which computerized, decentralized services can be described, arranged, persuaded and evolved in a make routine fashion. This outline presents a construction for adaptation and a base for protocols that will hold up the requirements of such service-oriented applications. The objective is a scalable, leveled architecture, one that can properly assemble the requirements of both undemanding and fabulously vigorous high-volume deployments. The focus of this representation it defines is a model for communicating, finding and exchanging information that is free of application implementations and the platforms on which applications are organized and developed.

A general framework is familiar with particular functions that need to be attempted in order to carry out decentralized interoperability. It does not determine the accurate technologies used to perform the functions but rather divides the problem space into sub-problems with specified relationships. This functional breakdown allows differing solutions to sub-problems without overlies, conflicts or excluded functionality. This is not to say that all applications have to exhibit the same facilities, slightly that when an attribute is offered it should fit into a general framework and preferably have a standard expression. In fact, they can be categorized into three stacks: wire format (or exchange format), description and discovery. It is also the case that some stacks depend on others. For example, the wire stack will grant the source for the discovery and description stacks. Thus, the “discovery” function may itself be a web service accessed through the wire stack.

**Wire** – those parts that suggest what is sent during a given exchange, the grouping of the data, the envelope and all other metadata entailed for the successful communication of a message.

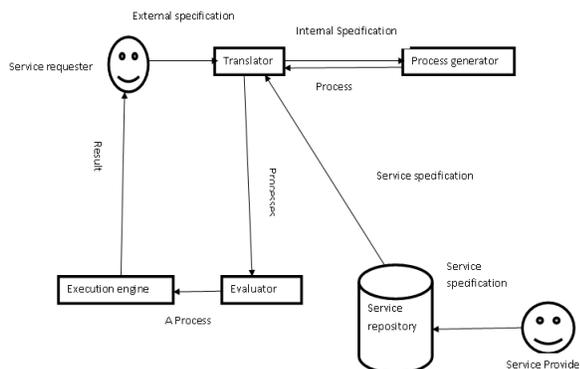
**Description** – the collection of specifications that provide the formal definition of the format, use, or application, of the specs in the “Wire” stack.

Make a note that the description framework is component oriented: the offerings are both layered and ordered. Each technology is built on the ones above, and simpler ones provide useful function by themselves. Therefore, expectation is that schemas and WSDL will be deployed first to afford descriptions for individual messages and message pairs. Over time, tools will become available to support the increasingly rich descriptions enabled by the other standards.

**Discovery** – those specs that provide a means for physical or computerized searching and discovery of the components in the other two stacks.

Coming to the composition the components present in simple, basic frame work are:

- a) Service repository
- b) Translator
- c) Process generator
- d) Evaluator
- e) Execution engine.



**Fig 1: framework of the service composition system**

Most precisely, the process of automatic service composition includes the following phases:

**Production of single service:** initially, the service providers (See figure.1) will publicize their atomic services at a comprehensive market place. There are a number of languages accessible for advertising, for instance, UDDI or DAML-S Service Profile .The fundamental attributes to illustrate a Web service comprise the signature, states and the non-functional values. The signature is characterized by the service’s inputs, outputs and exemptions. The states are detailed by precondition and post condition. Non-functionality values are those characteristics that are used for assess the services, such as the cost, service value and security concerns.

**Transformation of the languages:** Usually service composition systems distinguish between the external and domestic service specification languages. The external languages are utilized by the service users to develop ease of access of the users, with the use of translator (see fig.1).They are usually diverse from the internal ones that are used by the composition process generator. So far, the users have previously get used to the benchmark Web service languages, such as WSDL and DAML-S. Thus the translation components between the standard Web service languages and the domestic languages have to be developed.

**Creation of composition process model:** In the interim, the service requester can also state the requirement in a service specification language. The process initiator habitually takes the functionalities of services as input, and outputs process model that describes the composite service. The process model includes a set of selected atomic services and the control flow and data flow among these services.

**Estimation of composite service:** It is quite familiar that many services have the equivalent or comparable functionalities. So it is probable that the planer or evaluator (see fig.1) generates more than one complex service fulfilling the requirement. In that case, the composite services are assessed by their overall benefits using the information granted from the non-functional attributes. The most frequently used method is utility functions. The requester should spell out weights to each non-functionality attributes and the most excellent composite service is the one who is ranked on top.

**Execution of composite service:** After a unique composite process is selected, the composite service is ready to be executed. Execution of a composite Web service can be thought as a sequence of message passing according to the process model. The dataflow of the composite service is defined as the actions that the output data of a former executed service transfers to the input of a later executed atomic service.

In any environment, whether it may be static or dynamic the best out coming composed plan yields mainly depends on the coordination between the components.

## 2.1 Static Composition approaches:

### 2.1.1 Semi Automatic Code Generation of Static Web Services Composition

BPEL4WS (Business Process Execution Language for Web Services) [1] provides a notation for describing interactions of web services as business processes. Services are integrated by

treating them as partners that fill roles in a BPEL4WS process model.

MuhammadAdeel Talib and ZongkaiYang proposed in [16] a method to semi-automatically generate Web service composition in BPEL4WS language. They mainly work on two goals: first one is to afford composition modeler the flexibility to express less than complete detail without prescribing how much detail is needed and without requiring high specification knowledge. The other one is to allow business user the ability to renew changes in the process application without depending on the developer. In order to achieve these goals they device one system and it captures minimal most required information from the user and stores it in a relational model in order to create a static composition and captures the information required to develop the composition from the composition modeler and the WSDL interfaces of the collaborating partners. This information is stored in a relational model in the form of relations. Once all information is captured, transformation algorithm is applied to map from relational to BPEL model.

But one of the main challenges is that these two models i.e. relational and process models have strong formal grounds (relational and process algebra).The possibility of formally transforming the relational model into a process model is yet to be explored

### *2.1.2 A Reflective Framework to Improve the Adaptability of BPEL-based Web Service Composition*

Yanlong Zhai et al. introduced in [17] the reflective framework it aims to progress the flexibility of BPEL-based web service composition. In an outline, it is stated that the reflective framework endow with a appropriate solution to the adaptive service composition and reliable control flow and data flow correctness. They work out for leveraging computational reflection in support of runtime adaptability in web service composition. The framework is composed of two levels: base-level and meta-level. The base-level is the BPEL based web service composition and the major component of the meta-level is a graph-based meta-model defined to represent the service composition. The proposed structure what they constructed is based on computational reflection and data flow analysis. This framework allows changing the service composition dynamically whenever the execution environment changes or abnormal circumstances happen.

But the shortcoming with these approach is , the monitor and the change analyzer are performed manually. Therefore, it is further work to provide automatic process monitoring and change analysis.

## **2.2 Dynamic Composition approaches**

### *2.2.1 Dynamic Web Services Composition Based on QoS Model*

DONG Rang-sheng et al propose in [14] a QoS model to support dynamic Web services composition, They identify two problems to work out immediately in dynamic Web service composition : (1) How to select the service components. (2) How to coordinate services composition. For that purpose this team propose a QoS model for selection and coordination problems which supports dynamic Web services composition,

then develop a WS \_ TSC algorithm to select the services according to the restrictions on the response time, the rate of the success and the rate of the composition and this approach mainly focuses on Response time T(a), Reliability R(a) and Composability C(a) in Web services. Using the principle of linear programming transformed QoS parameters' formula in Web services composition flow into a single objective function to solve the optimization of Web Services selection. One algorithm is designed based on the model to select services, then assess the QoS performance in the dynamic Web services composition such as the response time, the rate of success and the rate of composition.

The QoS model presented in this approach has a certain degree of generality, so in the future the model and the algorithm can be applied to other service-oriented framework.

### *2.2.2 Solution to Dynamic Web Service Composition related to QoS*

G. Vadelou et al discussed in [13] about a new dynamic web services composition algorithm to solve the composition issues related to data distribution, reliability, availability and QoS. The key interest of this work is that it presents a method for automated and dynamic web service composition by arrangement of interface based and functionality based approaches. It focuses on the data distribution issues, QoS issues and defines how execution problems can be avoided, they device one algorithm to solve their proposed problem. Along with the algorithm, framework is also there in which multiple repositories and Web Service Database have been established in order to make system more trustworthy and ensure data accessibility. By using multiple registries, data availability is guaranteed whereas by using aging factor users can recover up to date information

There expectation in future is that the framework can be extended by crawling the web for searching web services instead of querying the UDDI registries.

### *2.2.3 An Optimal Web Services Integration Using Greedy Strategy*

Wen Ouyang and Min-Lang Chen suggest in [15] a new problem which looks into the possibility of reducing the number of hops of Web services while trying to finish a set of tasks. It also offers a polynomial-time, optimal Web service integration method using greedy strategy. They tries to investigate the Web service integration issues, which consider how to choose a set of web services which can satisfy the functional requirements of a collection of tasks and also trim down the Web service change over time. The idea is based on the concern that the communication time used when hopping between Web services contributes most for response time for most of the time and providing a polynomial-time, optimal greedy solution to integrate the Web services in order to achieve the goal of minimizing the hop count of Web services. So for achieving the objective an optimal algorithm, OPT, to find the way to integrate the web services using greedy strategy in the sense of minimizing the hop count between Web services in order to speed up the response time. This algorithm can find a way to integrate Web services so that the hop count between Web services is minimized. This way, we can speed up the processing time of these tasks.

Finally in the future, they will investigate the effectiveness of different Web services integration methods on other non-functional factors in order to study a more efficient and comprehensive way of integrating the Web Services

## 2.3 Semantics based approaches

### 2.3.1 Enhancing Semantic Web Services Composition with User Interaction

Peter Bartalos et al suggest in [11] an approach concerning users in the semantic web services composition to help overcome difficulties occurring in the composition process. The fundamental idea is to locate the users helpful in situations when preconditions are not satisfied or some input data are not available which are in demand to create a composition. However, a lot of work has been done in the field of semantic web services composition, there are still numerous problems needed to be solved, before the automatic composition can be applied in practice. The vital problem is that tools for automatic composition are immature to handle situations when some web services are out of stock or their descriptions are insufficient to find them [5]. Other problem is associated to the input data for web services which are not always available. In order to overcome these problems they propose an approach to find users able to solve problems occurring within automatic semantic composition. It take advantage of semantic metadata of web services descriptions and user models to realize this proposal for developing a web services composition method makes use of user interaction is to extend some existing efficient automatic approach. One of the best candidates seems to be OWL-S/XPlan. Their approach aims to improve a web service composition, but which is currently practically hard to be used in practice. They want to work in a future to plan to complete our proposal and examine it in more details. In particular, we deal with the definition of the neighbor set N.

### 2.3.2 Web Services Composition Method Based on OWL

Jike Ge Yuhui Qiu and Shiqun Yin propose in [10] a Web services composition method based on OWL ontology, and devise a system model for services composition. Work is concentrated mainly on to provide semantic capabilities for semantic Web services. For that purpose they extend or enhance the WSDL capabilities. This makes it feasible for the automatic Web services composition. These two people formulated three rules for the web services composition:

- Catalog composition
- Binding composition
- Operation sequence

For the implementation part, it takes place in a stage wise process first of all with the help of framework one plan is produced after that it sends to the service matching algorithm.

Based on the results given by the algorithm they generate a detailed description of composite service. But the current composition strategies are almost sequential, and a composite service might require a few services to be executed in parallel. It might be possible to improve the method to start parallel service matching to handle parallel service flows in the services composition.

### 2.3.3 Towards efficiency of QoS-driven semantic web service composition for large-scale service-oriented systems

Bin Xu et al discuss in [9] about to provide QoS efficiently and effectively for huge range composition problems. Up to now Optimization of QoS-driven semantic Web service composition is known to be NP-hard. So they are going to address the following problem: given a set of available services, how to efficiently build a QoS-guaranteed SOS. There are four reasons why this task may be difficult:

- Scalability
- Semantics
- Narrow focus on functionalities
- Complexity

They are mainly highlighted in the paper is that the response time for a SOS is not only influenced by the response time of services used in it, but also the execution plan through which the services are organized. For the execution plan, there are three basic flow structures in SOS including sequence, parallel and switch. They design a QoS-Driven Algorithm (QDA) for shortest sequence composition it is a stage-by-stage algorithm to achieve an optimal solution. This algorithm solves the previous problem i.e. NP Hard problem and it runs in a polynomial time.

They want to handle a large-scale service composition in a very short time and another goal is to integrate functional and non-functional consideration together to achieve the optimal solution that satisfies QoS requirements. As the semantic Web develops and QoS becomes more and more important in e-business, these issues must be addressed in the near future.

### 2.3.4 Ontology-driven service composition for end-users

Hua Xiao et al suggest in [12] an approach which can compose services on the fly to meet the situational needs of end-users. i.e. in a nut shell involving end users in a web services composition.

But achieving this involves three challenges.

- Complexity of service descriptions.
- Complexity of service composition languages.
- Limited support for composing services on the fly.

They come up with an approach it is also a stage wise process in a first step they present a tag-based service description schema which permits non-IT professional users to easily understand the description of services and add their own descriptions using descriptive tags after search for ontology and search for services are used. Last stage is a crucial stage i.e. Generation of Ad hoc process.

For the implementation part i.e. to generate ad hoc process they use IBM WSSR to register and manage web services. So the prototype contains three parts Goal Editor, Process Editor and Service Selection Panel. With the help of these three the user can create their own composite services. To generate better ad-hoc processes, it might be helpful if we can find and merge the information from more than one ontology.

**Table 1: Various web service composition approaches**

| <b>Title of the paper</b>  | <b>Algorithm used</b>                                | <b>Advantages</b>   |
|--|--|---|
| Semi Automatic Code Generation of Static Web Services Composition  | Transformation algorithm                             | Composition modeller can express in a simple way<br>At the time of updating user no need to depend on the developer                       |
| A Reflective Framework to Improve the Adaptability of BPEL-based Web Service Composition                   | Verification algorithm                               | Dynamically updating of service composition   |
| Dynamic Web Services Composition Based on QoS Model  | WS_TSC   | Focus is mainly on response time, reliability and composability   |
| Solution to Dynamic Web Service Composition related to QoS   | QoS based dynamic web services composition algorithm | Data availability is guaranteed<br>Users can recover up to date information   |
| An Optimal Web Services Integration Using Greedy Strategy  | OPT algorithm is used                                | It can minimize the number of hops<br>Processing speed is increased   |
| Enhancing Semantic Web Services Composition with User Interaction  | Heuristics algorithm                                 | It makes use of semantic meta data to describe web services   |
| Web Services Composition Method Based on OWL   | Service matching algorithm                           | Concentrates on QoS parameters<br>Extends WSDL with semantic capabilities   |
| Towards efficiency of QoS-driven semantic web service composition for large-scale service-oriented systems | QDA algorithm  | Works well for large scale composition problems<br>Equal preference to both functional and non-functional properties of composite service |
| Ontology driven service composition for end users  | Searching for Ontology, Searching for services       | Support end users in service composition<br>Hides the complexity of SOA standards   |

### 3. CONCLUSION

We considered in this paper about existing techniques addressing the problem of web services composition. Even though the diverse methods provide different level of automation in service composition, we cannot say the higher automation the better. Because the Web service environment is highly multifaceted it is not feasible to generate everything in an automatic way. Out of all those approaches semantics based approaches is emerging one and the research is going on this area. We anticipate that, especially in practical applications, additional requirements will be imposed onto a service composition engine. Such requirements will include quality of service (QoS). In order to achieve this QoS requirements Semantics based approaches is the best one.

### 4. REFERENCES

- [1] T.Andrews et al. Business Process Execution Language For Web Services (BPEL4WS), Version 1.1.
- [2] Milanovic, N., Malek, M.: “Current solutions for web Service Composition,” IEEE Internet Computing, vol.8, no.6, pp. 51-59,nov2004
- [3] Yasmine Charif and Nicolas Sabouret :”An Overview of Semantic Web Services Composition Approaches,” First International Workshop on Context for Web Services (CWS 2005),vol.146,no.1,pp.33-41, jan 2006
- [4] J. Rao, and X. Su, “A Survey of Automated Web Service Composition Methods,” In Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition,SWSWPC’2004, LNCS, San Diego, USA, Springer-Verlag, pp. 43-54,July 2004.
- [5] Fujii, K., Suda, T.:” Semantics-based dynamic Web service composition,” the IEEE Journal on Selected Areas in Communications (JSAC), special issue on Autonomic Communication Systems, Vol. 23, No. 12, pp. 2361 - 2372, Dec. 2005.
- [6] T. Berners-Lee, J. Hendler, and O. Lassila, “The Semantic Web,” Scientific American, vol. 284, no. 5, pp.34-43, May 2001.
- [7] Web Service Challenge. <http://www.ws-challenge.org>
- [8] Alrifai M, Risse T “ Combining global optimization with local selection for efficient QoS-aware service composition,” 18th International World Wide Web Conference, April 20th-24th, 2009
- [9] Bin Xu · Sen Luo · Yixin Yan · Kewu Sun “Towards efficiency of QoS-driven semantic web service composition for large-scale service-oriented systems,” Service Oriented Computing and Applications,vol.5,no.2,pp.1-13,June 2011
- [10] Jike Ge Yuhui Qiu Shiqun Yin : “ Web Services Composition Method Based on OWL,” 2008 International Conference on Computer Science and Software Engineering,vol.3,pp.74-77,2008
- [11] Peter Bartalos and Maria Bielikova “Enhancing Semantic Web Services Composition with User Interaction,” IEEE International Conference on Services Computing,vol.4,pp.503-506,2008.

- [12] Hua Xiao · Ying Zou · Ran Tang · Joanna Ng · Leho Nigul “Ontology-driven service composition for end-users,” *Service Oriented Computing and Applications*, vol.5, no.1, pp1-23, mar 2011
- [13] G. Vadelou, Dr. E. Ilavarasan “Solution to Dynamic Web Service Composition related to QoS”, IEEE, 2011
- [14] DONG Rang-sheng et al “Dynamic Web Services Composition Based on QoS Model,” *Intelligent Computing and Integrated Systems (ICISS)*, 2010 International Conference, vol.11, pp 823-826, oct 2010.
- [15] Wen Ouyang, Min-Lang Chen “An Optimal Web Services Integration Using Greedy Strategy,” *apsc*, pp.568-573, 2008 IEEE Asia-Pacific Services Computing Conference, 2008
- [16] Muhammad Adeel Talib, Zongkai Yang “SemiAutomatic Code Generation of Static Web Services Composition,” *Engineering, Sciences and Technology, Student Conference*, vol.37, pp 132-137, oct 2004
- [17] Yanlong Zhai, Hongyi Su, Shouyi Zhan “A Reflective Framework to Improve the Adaptability of BPEL-based Webservice Composition,” vol. 1, pp.343-350, 2008 IEEE International Conference on Services Computing 2008 .