Quality Assurance via Dynamic Composition of Web Services in Service Oriented Architecture

Vikas Gupta[#] Akash Gill , Parminder Kaur[*]

[#] *Department of Computer Science & Engineering, Guru Nanak Dev University*

Amritsar

[1]vikgup8@gmail.com

Akash.gill1687@gmail.com

[2]parminderkaur@yahoo.com

***Abstract*— In service oriented architectural style of software development, web services are the building blocks of the product. Several web services combine to form a software product that affects its performance. In other words we can say that a web service is a method of communication between two electronic devices over the World Wide Web. Undoubtedly, one of the significant concerns in the use of web services is their Quality of Service. Quality of Service is one of the substantial aspects for differentiating between similar service providers. Several parameters are measured to ensure that the web service meets the Service Level Agreements (SLA) which should not be violated to meet Quality of Service. Several web services providing the same user needs are available but it is those quality attributes that distinguish one web service from the other during their dynamic composition at the run time. Web services are combined at run time. So an issue that arises over here is how to choose the best available web service from a number of services at run time. This is done through Dynamic composition of web services at run time. So the issue that arises over here is the Quality assurance of web services via dynamic composition at run time in service oriented architecture without the violation of Service Level Agreements.**

***Keywords*— Service oriented architecture, Web Services, Quality assurance, Dynamic Composition, Service level agreements, Membrane monitor.**

I. INTRODUCTION

Service Oriented Architecture (SOA) is a software architectural style that aims to achieve loose coupling among interacting software agents. Web services play a key role in implementing enterprise-level applications based on SOA. The main advantage of SOA is the dynamic binding of web services to an application with business process. SOA is an architectural style which allows interaction of diverse applications regardless of their platform, implementation languages and locations by utilizing generic and reliable web services that can used as application building block.

Service consumers require obtaining guarantees related to services they use often called as Quality of Service (QoS). Quality of service management is critical for service-oriented enterprise architectures because services have different QoS characteristics, requestors have different requirements, and service interactions are decoupled (YunHee Kang, 2007).

II. RELATED WORK

(YunHee Kang, 2007) presented an extended web services framework based on SOA structure for providing information about quality of web services and build a prototype WebServiceBot for applying quality factors. The non-functional requirements of web services based on QoS parameters has been presented in this paper for the purpose of finding the best available web service during the web service discovery process.

(Farhan Hassan Khan .et al, 2010) proposed the solution of existing problems and proposed a technique by combination of interface based and functionality based rules. The proposed framework also solves the issues related to unavailability of updated information and inaccessibility of web services from repository/databases due to any fault/failure.

(Nizamuddin Channa .et al, 2005) proposed all criteria guiding the selection of services as constraints then choose an optimal set of services to satisfy the customer's requirements. This approach reduces the dynamic composition of web services to a constraint satisfaction problem.

(Kazuto and Mikio., 2006) proposes a value model and its representation language, VSDL (Value-based Service Description Language), of web services, and an architecture of value-added service broker of dynamically composing Web Services based on the value model .

(Liping Liu .et al, 2008) presented a model of web service composition based on particle swarm to resolve dynamic web services selection with QoS global optimal in web services composition. The theory of intelligent optimization of multi-objective genetic algorithm is utilized to produce a set of optimal Pareto services composition process with constraint principle by means of optimizing various objective functions simultaneously.

(Zhang and Gu Qing-rui, 2010) presented a way of dynamic composition of web services based on domain ontology. This way generates a web services composition graphics that is based on domain ontology by using domain ontology and semantic technology, according to graphics can support automatic discovery, dispatch, and compositions of web services.

III. SERVICE ORIENENTED ARCHITECTURE

Service Oriented Architecture (SOA) is an architectural style which allows interaction of diverse applications regardless of their platform, implementation languages and locations by utilizing generic and reliable web services that can used as application building block. SOA includes methodologies and strategies to follow in order to develop sophisticated applications and information systems (YunHee Kang, 2007).

SOA is different from the traditional architectures as it has its own unique architectural characteristics and regulations, which needs to be analysed and clarified so as to apply the information that should be included in the architectural model of SOA correctly to service based application development.

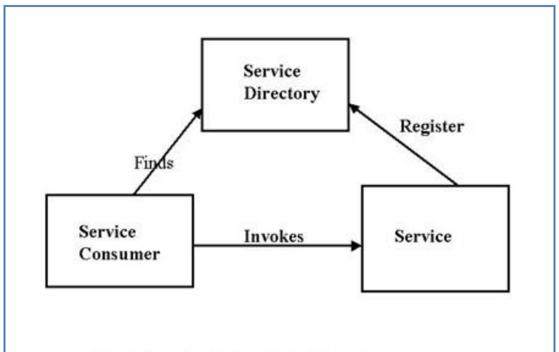

Fig 1. Service Oriented Architecture

Fig 1. Service Oriented Architecture

A. Web services in SOA.

Web services and e-services have been announced as the next wave of Internet-based business applications that will dramatically change the use of the Internet. Before the existence of this technology, the use of services from the World Wide Web was through a browser and a Web server using the HTTP protocol. Nowadays, this new model of interaction in the web given by web services has involved a primary subject of study which has had several contributions in the last few years.

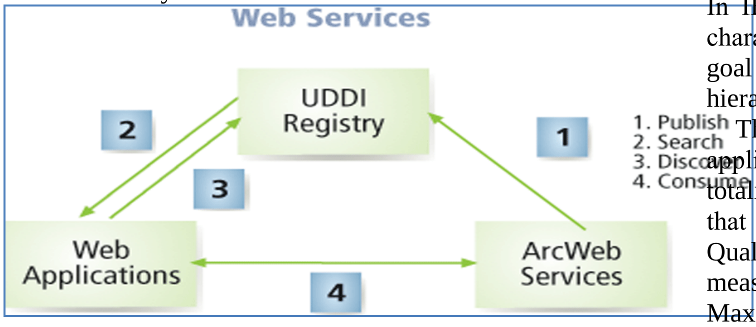

Fig 2. Actors of Web Service Model

The main actors of this new model are web service providers and web service consumers. Web Service providers are the ones who develop the service and publish the Web Service Description Language (WSDL). A WSDL is an XML interface used to invoke the service in a programming-language independent manner. Service providers are also responsible to register their services in a public service registry called Universal Description, Discovery, and Integration (UDDI). This registry would be used afterwards for the clients to discover the web services which fulfil their requirements (YunHee Kang, 2007).

B. Quality of Service.

A Software Quality model is a structured set of Quality Characteristics of Software.
There exist several Quality models for Software Systems; one of the most relevant was established in ISO9126, classifying the software quality in a structured set of characteristics and sub-characteristics as follows:

a) Functionality: Suitability, Accuracy, Interoperability, Compliance, Security
b) Reliability: Maturity, Recoverability, Fault Tolerance
c) Usability: Learnability, Understandability, Operability
d) Efficiency: Time Behaviors, Resource Behavior
e) Maintainability: Stability, Analyzability, Changeability, Testability
f) Portability: Installability, Replaceability, Adaptability, Conformance.

However, it is mentioned, that not all of these software quality attributes presented in are applicable to Web Services. For instance, installability can obviously not be applied to a web service. In this sense, those Quality Characteristics related to web services are known as Web Service Quality Characteristics or also Web Service Quality Factors.

Most of these quality sub-characteristics/sub-factors are also known in the literature of web services as Quality Attributes. In IEEE610 a quality attribute is defined as “A feature or characteristic that affects an item’s quality.” In a nutshell, the goal of quality Models is to group all quality attributes into a hierarchy of quality characteristics (Kazuto and Mikio., 2006). The sum of all these quality characteristics and attributes applied to a Web Service is defined as QoS. QoS is “the totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs”. A Quality Attribute is, nevertheless, not a quantitative measurement. The attributes to be examined include Maximum response time, Latency, Average Execution Time etc. These kinds of sub-attributes are known as Quality Metrics. As defined in “a quality metric is a quantitative measurement of the degree to which an item possesses a given quality attribute”.

C. Service Level Agreements

A service level agreement is an agreement regarding the guarantees of a web service. It defines mutual understandings and expectations of a service between the service providers and service consumers. The service guarantees are about what transactions need to be executed and how well they should be executed. An SLA may have the following components:

- *Purpose* - describing the reasons behind the creation of the SLA
- *Parties* - describes the parties involved in the SLA and their respective roles (provider and consumer).
- *Validity period* - defines the period of time that the SLA will cover. This is delimited by start time and end time of the term.
- *Scope* - defines the services covered in the agreement.
- *Restrictions* - defines the necessary steps to be taken in order for the requested service levels to be provided.
- *Service-level objectives* - the levels of service that both the users and the service providers agree on, and usually include a set of service level indicators, like availability, performance and reliability. Each aspect of the service level, such as availability, will have a target level to achieve.
- *A penalty - spells out what happens in case the service provider under-performs* and is unable to meet the objectives in the SLA. If the agreement is with an external service provider, the option of terminating the contract in light of unacceptable service levels should be built in.
- *Optional services* - provides for any services that are not normally required by the user, but might be required as an exception.
- *Exclusions* - specifies what is not covered in the SLA.
- *Administration* - describes the processes created in the SLA to meet and measure its objectives and defines organizational responsibility for overseeing each of those processes (Li-jie Jin , Vijay Machir"aju and Akhul Sahai).

IV. PROPOSED WORK

The base model for dynamic composition of web services via QoS in Service Oriented Architecture has been taken from(Nizamuddin Channa .et al, 2005) In this research work an architecture for dynamic composition has been proposed that aims at reducing the complexity and time needed to generate, and execute a composition and improving its efficiency by selecting the best possible services available at current time

The basic idea of this architecture is: service registry is a searchable registry of service where service providers publish their service description and service consumer find service and obtain binding information for services during development for static binding or during execution for dynamic binding. With the help of discovery engine and constraint optimizer a user can select the best available web services for composition.

The objective of this research is focused on three different but related subjects:

(1) The development of a review regarding to the Quality Attributes for web services in a systematic manner and what are the quality attributes that should be considered to evaluate the performance of web services.

(2) The Deployment of a tool for monitoring SOA Systems capable to be used in several frameworks such as for Self-Adaptive SOA Systems and for Web Service Discovery Systems. The tool used is **MEMBRANE SOA** that will help in analyzing web services, their performance and their behavior over a time period.

(3) To analyze the dynamic composition of web services at run time, propose and compare different algorithms or methodologies that focus on correct selection of web services and their composition at run time.

(4) Comparison of different tools and methodologies for Quality of Service of web services.

A. What is Dynamic Composition?

Dynamic composition is composition of web services at the run time as per the user requirements and needs. QoS parameters should not be violated as they ensure the quality of the web services to be combined. Dynamic composition should be deadlock free i.e. the combined web services should operate as per process.

B. General algorithm for composition of web services.

Our proposed service composition method is based on two standard Web service languages: Web Service Description Language (WSDL) and Web Service Choreography Interface (WSCI). WSDL describes the entry points for each available service, and WSCI describes the interactions among WSDL operations. WSCI complements the static interface details provided by a WSDL file describing the way operations are choreographed and its properties. This is achieved with the dynamic interface provided by WSCI through which the inter-relationship between different operations in the context of a particular operational scenario [7].

The flow of the composition procedure is as follows: First, get the WSDL of the Web service components from UDDI. Then, through the messages between the Web services, obtain the WSCI of the components. Afterwards, examine the input and output of the components through WSDL and determine the interactions between different components to provide the service through WSCI. Finally, perform the composition of the Web service with the information obtained in the composition procedure. The detailed composition algorithm is shown in Algorithm 1.

When it gets the required output, algorithm searches the Web services in the WSDL. In the operation tag of the WSDL, the output information is stated. When the desired output is found, thatWeb service component (*CPn*) is inserted as the root of the tree. Then, if the input of that operation matches the required input, the searching is finished and the input is inserted as a child of the *CPn*. Otherwise, it will search the action in WSCI in finding matches to the operation in *CPn*. After the action is completed, previous action can be determined. Then, it can find the operation prototypes in the WSDL. If the input of this operation matches the required input, then the composition is finished. Otherwise, loop will iterate until the root of the WSCI is reached.

If the desired input is still not found, the algorithm searches for the operations in other WSDL whose output is equal to the input of *CPn*. If the next Web service component found is *CPm*, then *CPm* is inserted as the child of *CPn*. The algorithm performs the searching iteratively and continues to build the tree until all the inputs match the required input.

```
Algorithm 1 Algorithm for Web service composition
Require: I[n]: required input, O[n]: required output
1: CPn: the nth Web services component
2: for all O[i] do
3: Search the WSDL of the Web services, and find the
CPn 's operation output = O[i]. Then, insert CPn
into the tree.
4: if the input of the operation = I[j] then
5: Insert the input to the tree as the child of CPn.
6: else
7: Search the WSCI of CPn, WSCI.process.action =
operation.
8: Find the previous action needing to be invoked.
9: Search the operation in WSDL equal to the action.
10: if input of the operation = I[i] then
11: Insert input to the tree as the child of CPn
12: else
13: go to step (8)
14: end if
15: end if
16: until reaching the root of WSCI and not finding the
correct input, search other WSDL with output = I[j],
insert CPm as the child of CPn and go to step (7) to
do the searching in WSCI of CPm.
17: end for
```

C. Membrane-SOA.

Membrane SOA is an open source tool available at http://www.membrane-soa.org/ that helps us to analyze web services based on quality attributes. Membrane SOA offers lightweight tools that help to run a successful and agile SOA. This tool is composed of different components with different functionalities.

a) MEMBRANE MONITOR-Capture, analyze and manipulate HTTP and SOAP messages.

b) MEMBRANE SOA REGISTRY-Membrane SOA Registry is an open source Web Services Registry providing:

- Availability and performance monitoring
- Monitoring of WSDL changes and versioning
- A generic Web Services client for testing
- A Web Service lifetime history
- Alerting over an Atom newsfeed
- Web 2.0 features like tagging and rating
- Dependency Management
- Reporting
- Endpoint Management
- Real-time performance monitoring.

Therefore, the final objectives of this tool are:

(1) To provide QoS information on run-time to detect SLA violations for Self-adaptive SOA Systems

(2) To provide QoS information on run-time in order to give trustworthy quality Information for a Service Ranking tool.

V. CONCLUSIONS

In this research work, the basic ideas of web services, their dynamic composition and quality assurance of web services are presented. Our service composition research aims at reducing the complexity and time needed to generate, and execute a composition and improving its efficiency by selecting the best possible services available at current time. Different methodologies for dynamic composition of web services will be analyzed and we will propose a methodology that will help the user in an interactive composition. The major issue regarding web services is their quality and their composition which leads to best service selection. QoS parameters also play an important role in determining which service suits the user requirements. This research work tries to eliminate the earlier limitations as discussed with respect to dynamic composition of web services in SOA via proposed algorithm for web service composition. Our future work includes the practical implementation of Membrane-SOA(an open source tool) in order to analyze web services based on quality attributes like maximum response time , latency , average execution time etc.

.

ACKNOWLEDGMENT

The efforts, contributed by research students, associated with this work, are highly acknowledged with great thanks.

REFERENCES

1. Farhan Hassan Khan, M.Younus Javed, Saba Bashir. 2010, "*QoS based dynamic web service composition & execution*, 2010; in proceedings of International Journal of Computer Science and information security , Vol.7. No.2, February 2010.

2. Kazuto Nakamura ,Mikio Aoyama..2006, "Value based dynamic composition of web services" in Asia Pacific Software Engineering Conference.

3. Li-jie Jin , Vijay Machir"aju and Akhul Sahai," Analysis on Service Level Agreement of web Services".

4. Liping Liu , Anfeng Liu , Ya Gao 2008, " Improved algorithm for dynamic web service composition*"* in proceedings of the The 9th International Conference for Young Computer Scientists .

5. Information Security, Vol. 7,No.2 ,February 2010.
Nizamuddin Channa1, Shanping Li1, Abdul Wasim Shaikh and Xiangjun Fu. 2005, "Constraint Satisfaction in dynamic web service composition*"*, Database and expert system Applications.

6. Pat. P. W. Chan and Michael R. Lyu. 2008, " Dynamic web service composition: A new approach in building reliable web service*"* in 22nd International Conference on Advanced Information Networking and Applications.

7. YunHee Kang. 2007, "Extended Model Design for Quality Factor based Web Service Management" in Proceedings of the Future generation communication and networking, Volume2 PP 484-487.

8. Zhang Hai-tao, Gu Qing-rui 2010, *"A dynamic web services composition and realization on the Base of semantic"* in proceedings of the 2nd international conference on Future Computer and Communication pp 624-627.