Quality Assurance of Web Services: Membrane SOA

Vikas Gupta[#], Parminder Kaur[*]

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

[1]vikgup8@gmail.com[s]

[2]parminderkaur@yahoo.com

***Abstract*— Quality of Service (QoS) management is critical for service oriented architectures because web services have different QoS characteristics and the requesters have different requirements. Web services are the building blocks in service oriented architecture and their quality assurance is an important aspect. A web service has to meet both the functional and non-functional requirements of the consumers. In this research work, emphasis is on the non-functional characteristics of the web service that can be determined at the run time. Service consumers require obtaining guarantees related to the web services they invoke and this is referred to as the Quality of Service (QoS). This research work aims to propose and successfully implement a QoS framework that will help us to monitor the quality of SOAP web services in order to prevent violation of the Service Level Agreements (SLA). The tool used to implement the QoS framework is MEMBRANE-SOA is an open source tool for web services monitoring. In this research work with the help of MEMBRANE-SOA web service registry is set up by registering the web services and monitoring them. So according to the demand of current situation we will implement a registry and perform Quality Assurance of services. This thesis aims to focus on detailed process of setting up the registry and monitoring attributes for detecting SLA violations.**

***Keywords*— Service oriented architecture, Web Services, Quality assurance, Quality of Service (QoS), Dynamic Composition, Service level agreements, Membrane Registry.**

I. INTRODUCTION

A web service is a technology that has emerged in the last few years. The important characteristic of web service is that it is a piece of software that can be utilized by a user but that user cannot own that piece of software. By ownership it means that the software cannot be installed but can be used by the user through standard protocols like Simple Object Access Protocol (SOAP), Universal Description, Discovery, and Integration (UDDI) and Web Services Description Language (WSDL).

As web services became popular due to their benefits, it led to the development of a new architecture paradigm called Service Oriented Architecture (SOA). In this architectural style of software development several web services are combined to meet the specified user requirements. Each web service is responsible to perform a concrete task and they are combined to obtain full-operational software.

As mentioned above that a SOA is composed of different web services as per the user requirements, so these services might be able to perform a task in a certain time, might be unavailable at certain periods of time , might have some security policies due to which their availability gets affected. All these attributes which affect the performance of these web services are called Quality attributes and play an important role in choosing a web service in a SOA system.

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. OBJECTIVE OF THIS WORK

The objective of this master thesis is focused on three but interrelated subjects:

- The development of a review regarding what quality attributes should be chosen for the quality assurance of web services in a systematic manner.
- The deployment of a tool MEMBRANE-SOA for analyzing the results of quality attribute and its metrics
- To check whether the web services meet the Service Level Agreements (SLA) mentioned by the service provider.

A. Development of a review

A review is necessary in order to identify and evaluate the most significant available research that is relevant to analyse the Quality attributes for the web services. A review helps us to focus on a particular set of operations and explores their importance. Particularly, we focus on the following aspects:

- Web services definitions with their quality attributes.
- Availability and performance monitoring.
- Root cause analysis of service failures.
- Event history.
- Gathers performance metrics for SLA checks.
- Ranking of the web services on the basis of chosen attributes.

SLA violations are an important part of this work as the services that violate the agreements affect their performance directly. Several parameters are measured to ensure that the web services meet the Service Level Agreements (SLA) which should not be violated to meet the Quality of Service (QoS).

Our work will aim on these final objectives:

- Analyze the existing work done on the quality attributes and models on web services.
- After the analysis of the existing work, proposal of new QoS framework.
- Implementation of the proposed QoS model.

B. Deployment of a monitoring tool

The applied part of this includes the deployment of a tool for

- Web services availability monitoring the QoS of a service.
- Reporting and analyzing the results obtained after monitoring the service.
- Checking the SLA violations of the service on the basis of the availability of the service in a specified period of time.

IV. PROPOSED QOS MODEL

A quality model is a structured set of quality characteristics of software in our case web services. After the registration of the service the monitoring for quality assurance of the service begins. The quality model is divided into two parts:

1. *Web services monitoring*: In order to ensure the quality assurance of web services, web services monitoring is performed in order to get valuable information for the availability of these services. Availability monitoring is done to get the percentage of time the service was available out of specified period of time. A service with high availability is preferred over a service with low availability. Root cause analysis of failure is done in order to get details of the reason for which the service went down. It can be due to network failure or failing to fetch WSDL of a service. SLA violation is also detected using www.uptime.in which gives us those services which have violated the agreed SLA parameters.

2. *Reporting*: Reporting helps us to maintain a record of the analyzed web services. A dashboard a maintained for these entire 20 web services which gives us the name, description, availability percentage and actions for all the services. After a service is deployed, a lot of events happen during its lifetime which includes service registration, service went up, and service went down. Membrane SOA registry keeps track of all these events in the event log.

The further sections of this thesis provide us with the detailed implementation of this QoS model along with the analysis of the data gathered during the monitoring period.

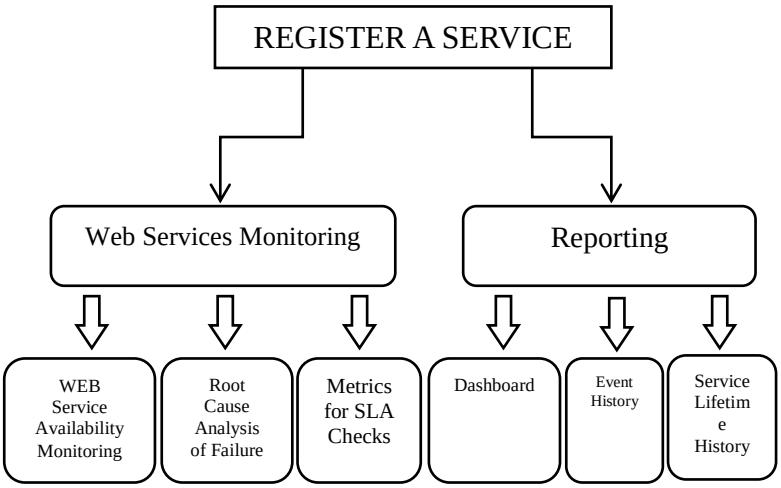

Fig.1. Proposed QoS Model

V. METHODOLOGY

In today's time of software development, organizations are increasingly using web services and are getting more and more dependent on these services. So there is a need for continuously monitoring these services to ensure the quality of SOA. SOA governance should be applied to minimize the risks and streamline the development and use of services.

The tool that helps us to carry out our research work is Membrane SOA Registry. It is a lightweight registry that observes web services to ensure a healthy SOA. The registry monitors the web services continuously and monitors and logs the availability of services. The registry also monitors the WSDL (Web Service Description Language) description of a service and detects if a change has occurred which also affects the quality of web service. The registry is installed in less than 5 minutes and it gathers information of the SOA by monitoring the services and artifacts like WSDL documents and XML schemas.

A. Web Services Monitoring?

In order to guarantee the quality of a service in SOA, there is a need to monitor the availability of services. Membrane SOA Registry monitors the availability of the registered services. It gathers the availability data of the services over a period of time and makes this information available. The availability of web services is critical as more and more systems depend on them. After a service gets registered in the registry, its availability gets monitored automatically.

Membrane SOA registry is an open source web services registry that provides with following monitoring capabilities.

- 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.

B. Parameters analyzed

The analysis of the web services begin after complete registration of the services in the repository. All the services are registered using their WSDL links at the same point in time. Monitoring time is fixed for all the services. Several parameters are there on the basis of which we can check the quality of web services. The non-functional parameters are analysed over here. The attribute chosen for the monitoring of these services is Availability. Availabity is the measure of the percentage of time for which the user is able to access the service. Availabity is an attribute and is not a quantitative thing that can be measured. The metrics are the quantitative sub-attributes that define the attribute chosen. MEMBRANE SOA Registry helps us to monitor the various sub-attributes of this attribute. It includes

- Service availability.
- Service uptime.
- Service downtime.
- Mean time between failures (MTBF).
- Mean time to recovery (MTTR).
- Number of downs.

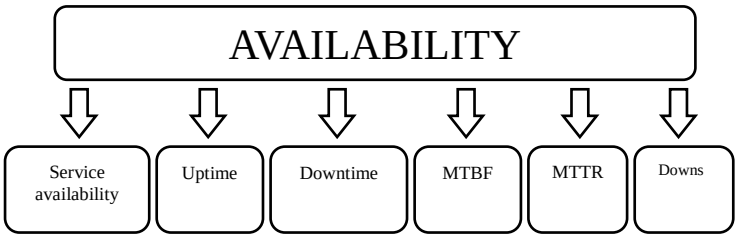

Fig 2. Attributes monitored.

C. Registering a Service

The next step is to register the services that have to be monitored in the registry. Service description is made available in the registry after registering the service. After registering the service the repository gives us valuable information about the service and its availability.

To register a service a couple of steps are needed.

- Copy the WSDL link of the service in the clipboard
- Press the register button.

For example service named Barcode Generator is to be registered and its availability has to be monitored continuously. The WSDL link of the service is mentioned in the clipboard.

WSDL link: http://www.webservicex.net/genericbarcode.wsdl

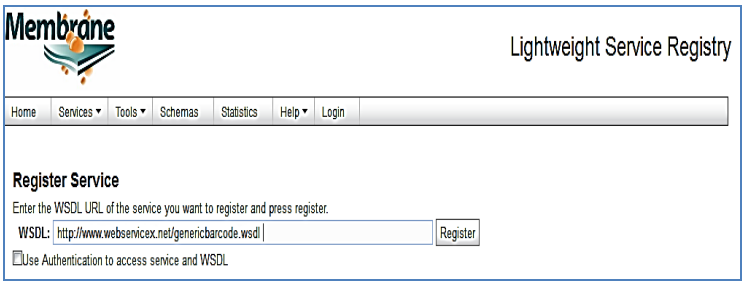

Fig3. To register a service only the WSDL is needed

D. Web Services Availability Monitoring

1. Service Availability

Service availability is the total uptime of a service divided by the total downtime of that service. The availability of the web services is monitored by MEMBRANE by calling them periodically. The time between two ping calls can be

configured for each service periodically. Here the ping interval for each service is 10 minutes i.e. after a period of 10 minutes a ping is made that returns the availability of service in the form of true or false. If the ping returns false, the service gets unavailable and its downtime starts till a successful ping occurs. The ping is sent to all the SOAP Web Services over the HTTP. Availability is directly proportional to successful pings. Larger the number of successful pings, lager is the availability and vice versa. Ping does not harms the service or invoke any business functionality. The service does not have to provide any special operation for ping call.

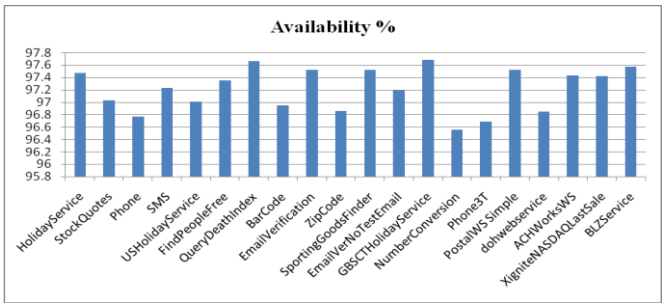

Fig4. Web services availability

When the service is available it returns true and when it is not available it returns false. Non-availability is due to the network error that results in HTTP code 0. The registry gives us the full details of both the HTTP code. The diagram below depicts the ratio of availability to the non-availability of the Barcode web service.

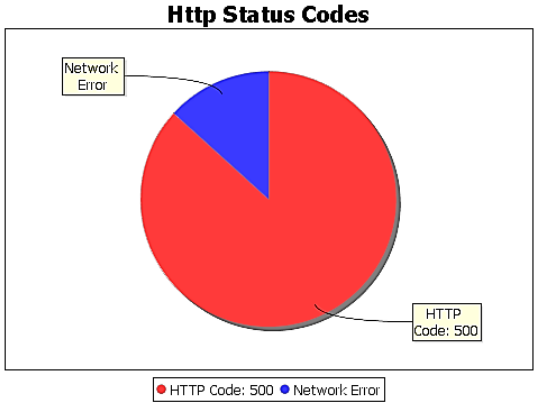

Fig5. HTTP Status Codes

2. *Service Uptime*

Service uptime is the time the service was available. The total uptime is the sum of all uptimes. During the monitoring period the service can get unavailable due to unsuccessful a ping which is known as downtime. So the uptime of a service includes Total Monitoring Period-downtime. Uptime of a service is directly proportional to availability. Larger the availability of a service, larger is the uptime of the service. A web service with larger uptime is always preferred over a service with lower uptime.

Uptime = Total Monitoring Period - Downtime

3. *Service Downtime*

Service downtime is the time the service was not available. The total downtime is the sum of all downtimes. Downtime occurs when a ping fails to get the WSDL of the service. A service with larger downtime is not preferred.

Downtime = Total Monitoring Period – Uptime

4. *Mean time between failures (MTBF)*

Mean time between failures is the elapsed time between the failures of a service during the monitoring. MTBF is calculated as the arithmetic mean or the average time between the failures of a web service. In other words MTBF is the average uptime from recoveries to failure.

5. *Mean time to Recovery (MTTR).*

The average time it takes for a service to recover from failure. The MTTR should be as small as possible. Smaller value of MTTR shows that the service is able to recover from failure at a fast rate as compared to a service with larger MTTR. The reason for the failure of the service includes network failures, ping failures and the registry not able to fetch WSDL of the service.

6. Number of downs.

Counts how often a service went down. In the monitoring period there number of times a service goes down. The registry counts the number of times services goes down. This is managed by event management. In the event management there is the details of the registering the service, the time the service gets up, the service gets down.

VI. RESULTS AND DISCUSSION

A. *Relationship between availability and Uptime*

Availability of services is directly proportional to the uptime of the service. Larger the availability of the service, larger is the uptime and hence more is the QoS of that service. On the x-axis availability in percentage (%) is plotted and on the y-axis uptime in hours (Hrs) is plotted. Then a graph is plotted that depicts that Availabity is directly proportional to uptime.

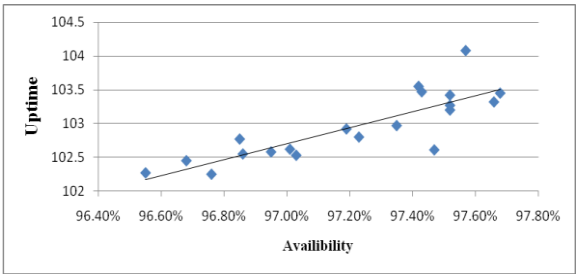

Fig6. Availability α Uptime

B. Relationship between availability and Downtime

Availability of services is inversely proportional to the downtime of the service. Larger the availability of the service, smaller is the downtime and hence more is the QoS of that service. On the x-axis availability is plotted and on the y-axis downtime time in hours (Hrs) is plotted. Then a graph is plotted that depicts that availabity is inversely proportional to downtime. Downtime should be as low as possible and it can be done by allowing the service to fail less frequently. The reasons for the downtime of the service are the registry failing to retrieve the WSDL of the service and the failure of the ping message.

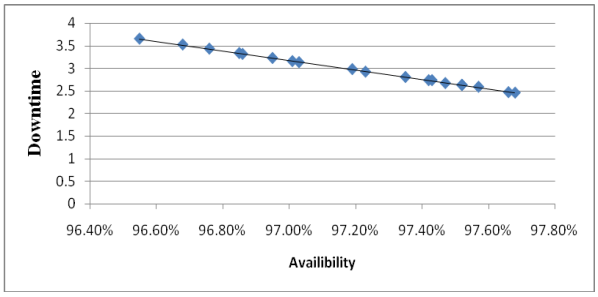

Fig7. Availability ∝1/Downtime

C. SLA Violations

SLA is the formalized contract between the service providers and service consumers that are used to define quality of service (QoS) properties. The violation of the SLA by the service provider usually results in a penalty to compensate the service consumer. Therefore, a measure for quantifying the danger of SLA violation is needed as a part of service level management system.

Membrane SOA Registry gave us the downtime for the services. SLA gives us the minimum downtime that is allowed. If a service has more downtime than the allowed downtime by SLA then it violates the agreement. SLA violation has an impact on the QoS of service. Allowed downtime is calculated by the tool http://uptime.is/advanced?sla=99.9.In our research work it is found that all the services violate the SLA allowed downtime and hence have low QoS. When a SLA is violated by the service provider, the service provider has to compensate to the service consumer. The downtime should be as low as possible for the quality of the services as larger downtimes make the consumer believe that the service is not available.

Web Service	Downtime(Hrs)	Allowed Downtime(Hrs)	SLA violated
WS1	2.672	2.635	Yes
WS2	3.133	3.085	Yes
WS3	3.432	3.373	Yes
WS4	2.924	2.884	Yes
WS5	3.158	3.109	Yes
WS6	2.806	2.756	Yes
WS7	2.471	2.44	Yes
WS8	3.227	3.175	Yes
WS9	2.633	2.583	Yes
WS10	3.317	3.226	Yes
WS11	2.629	2.583	Yes
WS12	2.981	2.926	Yes
WS13	2.46	2.413	Yes
WS14	3.653	3.585	Yes
WS15	3.524	3.453	Yes
WS16	2.631	2.583	Yes
WS17	3.336	3.279	Yes
WS18	2.736	2.676	Yes
WS19	2.736	2.687	Yes
WS20	2.59	2.531	Yes

VII. CONCLUSIONS.

Quality assurance (QoS) of web services plays an important role in application development using Service Oriented Architecture (SOA). In Service Oriented Architecture the components are the web services. There may be many service providers providing web services having same functionality. Now the question arises is how to choose best available service and on which criteria. Web services are defined by quality attributes on the basis of which they are monitored. The attribute chosen in this research work is availability and the metrics that define this attribute are Service Availability, Uptime, Downtime, Mean Time between Failure (MTBF), and Mean Time to Recovery (MTTR) and the Number of downs. After analyzing all these metrics, it is concluded that availability is directly proportional to the uptime of the web service and availability is inversely proportional to downtime. The services with low availability are usually considered as non-available. The services with low availability are not good for the application using SOA. Larger the downtime, larger is the time the client has to wait for the service. The services have to meet the Service Level Agreements (SLA) that serves as a contract between service provider and the consumer. This research work provides us a mechanism to detect SLA

violations of monitored services on the basis of downtime. During the monitoring period, if the services are having more downtime than allowed, they are said to be having violating the SLA.

Future work needs to be done in order to develop monitoring tools that can help in monitoring all the attributes at the same time. The tool used in this research work (MEMBRANE- SOA) provides us with only one attribute and its metrics which is not sufficient for complete Quality Assurance of Web Services. In order to get complete Quality Assurance, several Non-Functional and Functional parameters are to be monitored that will result in a complete QoS Framework. The tool should also incorporate some ranking mechanism on the basis of which they can be ranked as per their attributes values. The ranking mechanism can be on the basis of one attribute or combining two or more attributes. These will help in more reliable SOA applications. s

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.

