International Journal of Advanced Technology in Engineering and Science
Vol. No.3, Special Issue No. 01, November 2015
www.ijates.com
ijates
ISSN 2348 - 7550

REQUIREMENTS AND APPROACHES OF QUALITY OF SERVICE BASED WEB SERVICES – AN OVERVIEW

P. Karthikeyan[1], Dr. G.J. Joyce Mary[2]

[1]Doctoral Research Scholar, [2]Associate Professor, Department of Computer Science Engineering, PRIST University, Thanjavur, Tamil Nadu, (India)

ABSTRACT

Due to the evolution of web services, the global scenario today is witnessing a software revolution which has made our society to depend purely on it. The services provided by it have attracted many users in our modern world to rely on web services to fulfill their needs. This in turn has increased its responsibility and has thus made its architecture very complex. It's first and foremost responsibility is to provide Quality of Service (QoS). The sustainability of any service providers is decided by the quality of the service provided by them. QoS is nothing but a set of properties and attributes that are requires for the system to satisfy the needs of the network. It is usually a synonym behind performance. This paper mainly focuses on the requirements and the different approaches used in QoS based web services.

Keywords: Web Services, Quality of Service (Qos), Requirements, Approaches

I. WEB SERVICES – AN INTRODUCTION

Today's internet facilities have developed very much such that people choose their products and services based on their requirement, quality and cost via the internet. This is the service offered in the web and is termed as web services. [1]. Web services are modular, self-describing, dynamic and loosely-coupled software applications that can be located and used across the Internet using a set of standards such as SOAP, WSDL and UDDI. [2]. According to wikipedia, "a web service is a method of communication between two electronic devices over a network." And according to W3W "It is a software system that provides interoperability between machines interacting over a network".

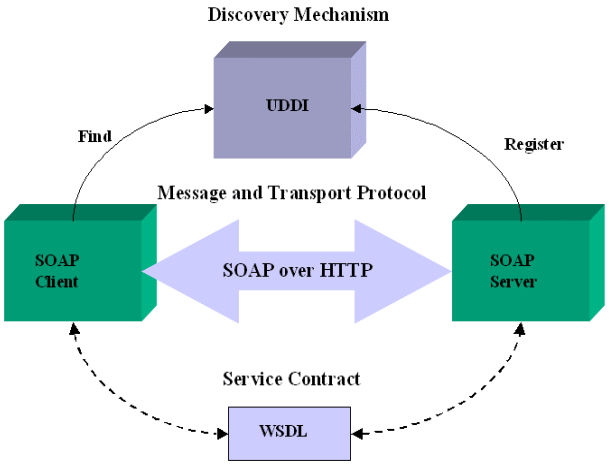

Fig: 1 Service Oriented Architecture using Web Services

II. BOTTLENECKS IN WEB SERVICES

WS face bottlenecks in their performance due to their underlying protocols. The commonly used protocols are HTTP and SOAP. HTTP is a best-effort delivery service in which there is no assurance of packets delivery. If bandwidth is insufficient, then the packets are dropped. Moreover, as the web services communicate over the internet, delay will be in the order of milliseconds. SOAP is Simple Object Access Protocol. SOAP uses XML as its payload. Hence extracting the packets is time consuming. Moreover, if more number of SOAP messages are transmitted over the internet, the bandwidth requirement becomes more. There are several other factors that affect the performance for web services. Some of them are availability, response time, run time, delay, etc. [3]. To improve the efficiency of web service in spite of these bottlenecks, quality of service (QoS) is needed.

III. QUALITY of SERVICE (QoS)

QoS is defined in ISO 8402 as: "The totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs". It is a set of non-functional attributes that may influence the quality of the service provided by a resource and consequently represent key components of a Web Service Agreement.[5].

In today's complex web service environment, QoS maintenance is a key factor for the sustainability of any service providers as the consumers today expect good quality of service. Based on the QoS characteristics, the web services are selected. Thus for each specific requirements, QoS characteristics varies and thus different web service must be selected. It is understood from many recent research works that Quality of Web Services is emerging as a main aspect.

These works state great importance of QoS properties for using specific resources in a Service Oriented Architecture (SOA). [1].

IV. A WEB SERVICE MODEL INCLUDING QoS CONSTRAINS

This is a web service model where the quality of service QoS is taken as constrain when searching for web services. This framework is a regulated model that can co-exist with the current de-regulated UDDI registries. The current de-regulated registries can offer services to people to whom the quality of service is not important whereas the regulated registries based on the new model can serve to the applications needing quality of service assurance. [4].

In this model, Web service provider needs to supply information about the company, the functional aspects of the provided service as requested by the current UDDI registry, as well as to supply quality of service information related to the proposed Web service. The claimed quality of service needs to be certified and registered in the repository.

The Web Service Consumer needs the web service offered by the provider. The consumer searches the UDDI registry for a web service with the required functionality as usual; plus he can add constraints such as required quality of service to the search operation. The search would or would not return a desired web service. Once a web service is found, the WSDL and certified QoS information is retrieved by the consumer. The consumer can verify the QoS claims with the certifier using the certification Id.

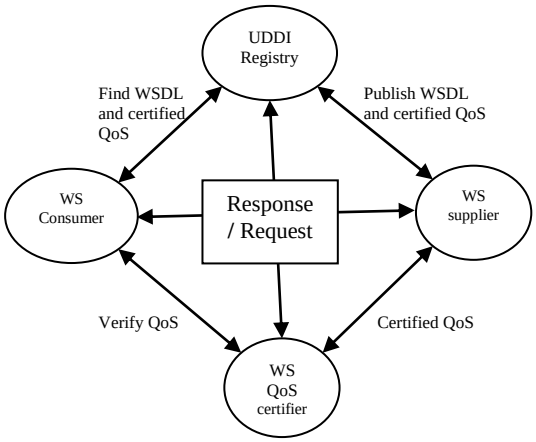

Fig2. A Web Service Model with QoS Constrain

Web Service QoS Certifier is responsible to verify the claims of quality of service for a web service before its registration. The new UDDI registry is a repository of registered Web Services with lookup facilities. It is different with the current model of UDDI by having information about the functional description of the web services as well as its associated quality of service registered in the repository.

V. QoS REQUIREMENTS

Accuracy

It deals with the number of errors that the service generates over a time interval. It should be minimized.

Accessibility

It represents whether the web service is capable of serving the client's requests. It may be expressed as a probability of measure denoting the chance of a successful service instantiation at a point in time.

Availability

Any web service application must be ready for consumption. It must have very small mean time-to-repair (i.e.) the time it takes to repair the Web Service should be minimum.

Capacity

It is the maximum number of requests that can be provided with guaranteed performance simultaneous.

Exception handling

A web service must be able to handle special cases and unanticipated possibilities.

Interoperability

It is the key advantage of any web service application. It must reliably operate even between different platforms.

Integrity

Prevent unauthorized access to, or modification of, programs or data. It is the aspect of how the Web service maintains the correctness of the interaction in respect to the source. If any loss of data packets is there, corrective actions must be taken.

Performance: It is the speed at which a service request is completed. It can be measured in terms of throughout, response time, latency, execution time, transaction time

- *Throughput*: Number of web service requests served in a given time.
- *Response time*: Time required to complete a Web service request
- *Latency*: round-trip delay between sending a request and receiving the response
- *Execution time*: Time taken by a Web service to process its sequence of activities
- *Transaction time*: Time that passes while the Web Service is completing one complete transaction

Web Service should provide high throughput, fast response time, low latency, low execution time, fast transaction time.

Reliability

It is the ability to perform its required functions under the given circumstances over a given time interval. The overall measure is used to measure its service quality and is related to number of failures per day, week, month or year.

Robustness

It is the degree to which a web service can function correctly even in the presence of incomplete, invalid or conflicting inputs.

Security

Security is mandatory in any web service application. Security must always be given higher priority. It is the aspect that provides confidentiality and access control. The service providers can use different methods of authentication to secure the service provided to the requestor.

Scalability

The capability of increasing the computing capacity of service provider‘s system and the ability to process more users’ requests in a given time.

VI. EXISTING APPROACHES IN QoS BASED WEB SERVICES

OWL-Q [6] is an extensible ontology language is a complement of the WS functional description language OWL-S. This can be used for both request and response. OWL-Q has eleven facets. OWL-Q (main), Measurement Directive, Time, Goal, Function, Measurement, Metric, Scale, QoSSpec, Unit and ValueType.

The Main Facet connects OWL-S with OWL-Q and provides the high-level QoS concepts. For this connection, subclasses called as ServiceAttribute class is used. In *QoSSpec facet*, the classes that define QoS offers and requests are represented. It has two subclasses: QoSoffer and QoSDemand so as to enable both providers and requesters to define their specification in the same way. In *Goal Facet*, mathematical formulas and QoS goals/constraints were previously expressed. In *Measurement Facet*, measurements are modeled in OWL-Q in order to enable their storage and statistical processing by registries or other parties. Functions in OWL-Q are separated into functions applied to metrics and functions applied to scales in *Function Facet.*

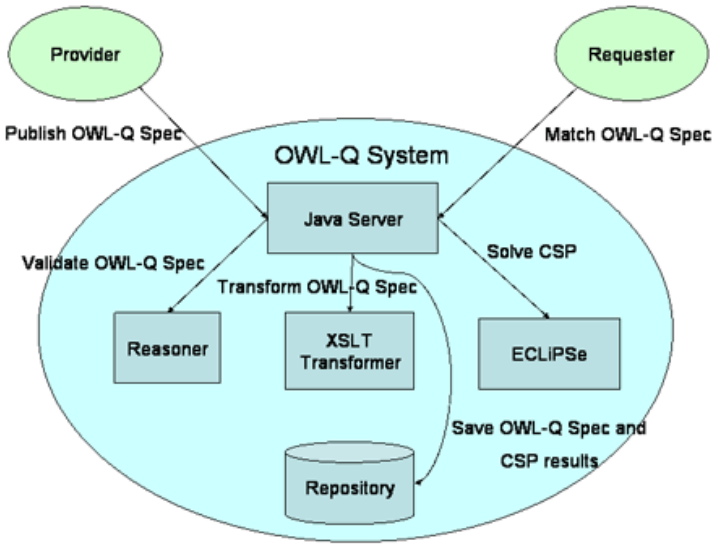

Fig:2 The Architecture of the QoS-based Web Service Discovery Process

In *Measurement Directive Facet*, the MeasurementDirective class specifies the way in which simple metrics are measured. *Time facet* specifies the Schedule and Trigger classes. The *Metric Facet* describes all the appropriate classes and properties used for a proper formal definition of a QoS metric. By using *Scale Facet, a* measurement scale controls the value type and the type of operations allowed for a metric and belongs to a specific Attribute. The *Unit Facet* formally describes the unit of a ratio scale of a QoS metric. A Unit has one name, several abbreviations and synonyms. The *ValueType facets* describe the types of values such as numbers, alphabets, etc that a QoS metric can take.

OnQoS [6] is used for specifying QoS advertisements by providers and QoS requirements by consumers. It is a quantitative approach. This ontology is composed of three extensible layers: upper, middle and lower ontology.

Upper ontology explains the words in the language that is needed for handling and answering the QoS queries. The *middle ontology* just refines the operation of upper ontology. It defines the standard vocabulary of the ontology.[1]. The QoS vocabulary is a set of parameters which is independent of domain. QoS parameters can be evaluated from the perspective of providers and also from the users. For service providers, a QoS parameter describes

the provided QoS and the service consumers expect the required QoS. The lower ontology is domain specific and it describes the concepts and properties of each domain and these are used in real applications.

QoS-MO [8] is an upper level ontology that describes QoS characteristics and web service constrains. The QoS-MO has the following classes. For QoS characteristics; *QoS characteristics*, which is a main class for defining characteristics of services; *QoSCategory* is used to group related QoS characteristics; *QoSDimension* is for modeling the measurement of QoS characteristics; *QoSContext* allows the definition of quality expression that combine multiple QoS characteristics; *QoSValue* represents the value of a *QoSDimension*. For QoS constrains, there are three types of QoS constrains: *QoSOffered*, *QoSRequired*, and *QoSConstract*. *QoSOffered* can be defined by providers as well as consumers to express the QoS that the consumer will get from the providers and the expected constrain by the consumer when accepting the service. *QoSRequired* is similar to *QoSOffered*. *QoSConstract* connects both *QoSRequired* and *QoSOffered*. Finally for QoSLevels describes the various modes of QoS that a service can support and can be changed to other levels by *QoSTransitions* which specify essential actions for the transition.

VII. CONCLUSION

QoS plays a key role in the efficient performance of web services and because of this the world today have modernized so much. It has become a mandatory requirement of business transactions. In this paper, the various QoS properties that are to be considered while selecting web services was discussed. For effective servicing, these requirements must be satisfied. These requirements are becoming more and more complex due to the complicated architecture of the web service today. There are several bottlenecks that hinder the operation of web services. But these can be overcome by the new approaches and new models of web services that are developed today.

REFERENCES

[1] More Semantics in QoS Matching by Ester Giallonardo and Eugenio Zimeo, IEEE International Conference on Service-Oriented Computing and Applications(SOCA'07)

[2] Requirements for QoS-based Web Service Description and Discovery by Kyriakos Kritikos and Dimitris Plexousakis, Foundation for Research and Technology-Hellas (FORTH) and Institute of Computer Science (ICS).

[3] Understanding quality of service for Web services by Arun Nagarajan and Anbazhagan Mani, IBM Developers works.

[4] Shuping Ran, “A Model for Web Services Discovery With QoS”, CSIRO Mathematical and Information Sciences, Australia

[5] S.Bleul, T.Weise, Kurt Geihs, “An Ontology for Quality-Aware Service Discovery”, special issue on "Engineering Design and Composition of Service-Oriented Applications", Computer Systems Science Engineering, 2006.

[6] OWL-Q for Semantic QoS-based Web Service Description and Discovery Kyriakos Kritikos and Dimitris Plexousakis, Foundation of Research and Technology, Heraklion, Greece.

[7] The QoS-MO Ontology for Semantic QoS Modeling by Gustavo Fortes Tondello and Frank Siqueira, Federal University of Santa Catarina

[8] Semantics in QoS for Web Services: A Survey by Vuong Xuan TRAN and Hidekazu TSUJI, SIG-SWO-A801-02