Quality of Service (Qos) Issues in Web Services

Kuyoro Shade O., Awodele O. Akinde Ronke O. and Okolie Samuel O.

Babcock University, Ilishan-Remo, Ogun State, Nigeria

Summary
The World Wide Web has evolved from being a pure information repository to a more functional and service oriented platform using technologies such as Web Services. Web service has been widely employed in e-business, e-government, automotive systems, multimedia services, process control, finance, and a lot of other domains. Quality-of-Service (QoS) is usually employed for describing the non-functional characteristics of Web services and employed as an important differentiating point of different Web services. With the prevalence of Web services on the Internet, a thorough awareness of the Web service QoS is becoming more and more important. In this article, an overall description of Web Services QoS requirements is presented.

Key words:
Service oriented platform, QoS, e-business, Web Services

1. Introduction

Web services have been emerging in recent years and are by now one of the most popular techniques for building versatile distributed systems. They are software components that communicate using pervasive, standards-based Web technologies including HTTP and XML-based messaging. Web services are designed to be accessed by other applications and vary in complexity from simple operations, such as checking a banking account balance online, to complex processes running CRM (customer relationship management) or enterprise resource planning (ERP) systems. Since they are based on open standards such as HTTP and XML-based protocols including SOAP and WSDL, Web services are hardware, programming language, and operating system independent. This means that applications written in different programming languages and running on different platforms can seamlessly exchange data over intranets or the Internet using Web services. Basically, four technologies form the basis of Web services: eXtensible Markup Language (XML); Simple Object Access Protocol (SOAP); Web Services Description Language (WSDL); and Universal Description, Discovery, and Integration (UDDI).

eXtensible Markup Language (XML) was created as a structured self-describing way to represent data that is totally independent of application, protocol, vocabulary, operating system, or even programming language.

Manuscript received January 5, 2012
Manuscript revised January 20, 2012

Simple Object Access Protocol (SOAP) is used for communication among different Web Services. SOAP was created as a way to transport XML from one computer to another via a number of standard transport protocols such as HTTP. A SOAP message consists of Soap Envelope which contains Soap Body element and an optional Soap Header element. The Soap Header element may contain a set of child elements that describe message processing that the sender expects a recipient to perform. Below is a typical SOAP listing.

```
<Soap: Envelope—-
<Soap: Header (optional)>
<Soap: Body> (mandatory)
<get Quote symbol = "——"/>
</Soap: Body>
</Soap: Envelope>
```

Listing 1: A Simple SOAP message

Web Service Description Language (WSDL) is used to describe the functionalities of the services. It is how one service tells another which way to interact with it, where the service resides, what the service can do, and how to invoke it.

Universal Description Discovery and Integration (UDDI) is used as a registry of information for Web Services, that is, to publish and discover information. It is an industry-wide effort to bring a common standard for business-to-business (B2B) integration, which defines set of standard interfaces for accessing a database of Web services. The purpose of UDDI is to allow users to discover available Web services and interact with them dynamically.

With the tremendous success of the Web service technology in both business and research area, quality of service (QoS) issues play an important role for Web service providers. The need for QoS in Web services is driven by two demands. Clients aim to experience a good service performance, e.g. low waiting time, high reliability, and availability to successfully use services whenever the need arises. On the other hand, especially in e-business, service providers need to formulate QoS-aware offers in order to gain the highest possible profit from their business. Examples are high throughput guarantees and low response time through dynamic capacity allocation, resource allocation, and load balancing in order to serve a high number of clients with assured QoS. Moreover, crucial transactions such as payment should experience prioritized execution realized by transaction differentiation. Service providers will strive to find an optimal relation between user satisfaction and system utilization.

Web services QoS can be described as a set of non-functional attributes that may impact the quality of the service offered by a Web service. Quality-of-Service (QoS), which is usually employed for describing these non-functional characteristics, has become an important differentiating point of different Web services [1].

In the field of service computing, Web service QoS have been discussed in a number of research investigations for presenting the non-functional characteristics of the Web services ([1][2][3][4][5][6]. [7] employ five generic QoS properties (i.e. execution price, execution duration, reliability, availability, and reputation) for dynamic Web service composition. [8] use five QoS properties (i.e., execution time, availability, price, reputation, and data quality) when making adaptive service composition in flexible processes. [9] propose an efficient service composition approach by considering both generic QoS properties and domain-specific QoS properties. A wide variety of QoS parameters for web services have been presented in previous work ([1][10][11][12][13]). This work presents an overall description of requirements for measuring the quality of service (QoS) of Web services. For the purpose of illustration in this work, QoS requirements are grouped into different categories, i.e. security, configuration management and cost, transaction support, and QoS related to runtime to present an overall description of web services quality of service that can serve as yardsticks for measuring quality in web services..[14] The remaining sections are arranged as follows. Section 2 presents security related web services QoS, section 3 describes configuration management and cost related web services QoS, section 4 presents transaction support related web services QoS, section 5 highlights the runtime related web services QoS and section 6 presents the conclusion.

2. Security Related QoS

With the increase in the use of web services which are delivered over the public Internet, there is a growing concern about security. The web service provider may apply different approaches and levels of providing security policy depending on the service requestor. Security for web services means providing authentication, authorization, confidentiality, and non-repudiation. Each of these aspects is described below [14][15].

A. Authentication verifies that principals (human users, registered system entities, and components) are who they claim to be. The result of authentication is a set of credentials, which describes the attributes (identity, role, group, and clearance) that may be associated with the authenticated principal.

B. Authorization grants permission for principals to access resources, providing the basis for access control, which enforces restrictions of access to prevent unauthorized use. Access controls ensure that only authorized principals may modify resources and that resource contents are disclosed only to authorize principals.

C. Confidentiality keeps the message secret. This process requires encryption, which scrambles the message in such a way that only authorized identities can decrypt and see the data. To do this, a shared secret and an algorithm for encrypting and decrypting the message is exchanged. In the real world, these algorithms are very challenging mathematical functions with keys that are very large numbers, and the time to do the analysis is technically infeasible even with modern computers.

D. Cryptography provides cryptographic algorithms and protocols for protecting data and messages from disclosure or modification. Encryption provides confidentiality by encoding data into an unintelligible form with a reversible algorithm, which allows the holder of the decryption key(s) to decode the encrypted data. A digital signature provides integrity by applying cryptography to ensure that data is authentic and has not been modified during storage or transmission.

E. Accountability ensures that principals are accountable for their actions. Security auditing provides a record of security-relevant events and permits the monitoring of a principal's actions in a system. Non-repudiation provides irrefutable proof of data origin or receipt.

F. Security administration defines the security policy maintenance life cycle embodied in user profiles, authentication, authorization, and accountability mechanisms as well as other data relevant to the security framework.

G. Non-repudiation proves that one identity sent the data only to another identity. This then proves that the specific transaction was entered into by the recipient, and neither party can refute or deny that it occurred later. If the transaction is challenged legally, a contract that was supposedly executed must be shown to have been entered into by both parties. Each party must have seen the contract signed, and their identities confirmed traditionally by validating wet signatures on paper and notary witnesses- must have been confirmed at the time of signing. These are difficult, and as yet legally unchallenged, tenants to uphold in a digital and anonymous world, but that day is coming. Non-repudiation depends on public key cryptography technology.

3. Configuration Management and Cost Related QoS

A. Regulatory is a measure of how well the service is aligned with regulations.
B. Supported Standard is a measure of whether or not the service complies with standards (e.g. industry specific standards). This can affect the portability of the service and interoperability of the service with others.
C. Stability/change cycle is a measure of the frequency of change related to the service in terms of its interface and/or implementation.
D. Guaranteed messaging requirements ensures the order and persistence of the messages
E. Completeness is a measure of the difference between the specified set of features and the implemented set of features.
F. Service charge is the cost involved in requesting the service. The web service cost can be estimated by operation or by volume of data.
G. Web services should be interoperable between the different developments environments used to implement services so that developers using those services do not have to think about which programming language or operating system the services are hosted on. [14]

4. Transaction Support Related QoS

Integrity asserts that no one has tampered with a message since it was initially created. This assures the sender and the receiver that every bit produced by the sender is received by the recipient in precisely unaltered form. This is described by the ACID properties: Atomicity (executes entirely or not at all), consistency (maintains the integrity of the data), isolation (individual transactions run as if no other transactions are present) and durability (the results are persistent). [14]

5. Runtime Related QoS

A. Scalability is the capability of increasing the computing capacity of service provider's computer system and system's ability to process more operations or transactions in a given period.[14] It is closely related to throughput and performance. Every Web service should be high scalability in terms of the number operations or transactions supported.

B. Capacity is the limit of the number of simultaneous or concurrent requests which should be provided with guaranteed performance.[14] Web services should support the required number of simultaneous connections.

C. The performance of a web service represents how fast a service request can be completed. It can be measured in terms of throughput, response time, latency, execution time, and transaction time, and so on.[14][15] Response time is the guaranteed max (or average or min) time required to complete a service request. It is related to capacity. Latency is the time taken between the service requests arrives and the request is being serviced. Throughput is the number of completed service requests over a time period. Throughput is related to latency and capacity. Execution time is the time taken by a web service to process its sequence of activities. Finally, transaction time represents the time that passes while the web service is completing one complete transaction. This transaction time may depend on the definition of web service transaction. Higher throughput, faster response time, lower latency, lower execution time, and faster transaction time is expected of high quality web services.[17]

D. Reliability is the ability of a service to perform its required functions under stated conditions for a specified period of time. [16] The reliability is the overall measure of a web service to maintain its service quality. The overall measure of a web service is related to the number of failures per day, week, month, or year. Reliability is also related to the assured and ordered delivery for messages being transmitted and received by service requestors and service providers.[15] It is closely related to availability.

E. Availability is the probability that resources, services is available to authorized parties at all times for use or the percentage of time that the service is operating.

F. Robustness/Flexibility is the degree to which a service can function correctly in the presence of invalid, incomplete or conflicting inputs.[14] Web services should be provided with high robustness that can still work even if incomplete parameters are provided to the service request invocation.[17]

G. Accuracy defines the error rate produced by the web service.[14] Web services should be provided with high accuracy and the number of errors that the service generates over a time interval should be minimized.

H. Exception handling is how the service handles these exceptions. Since it is not possible for the service designer to specify all the possible outcomes and alternatives (especially with various special cases and unanticipated possibilities), exceptions can be expected.[14] Exception handling is related to how the service handles these exceptions. It can be in a brutal or a graceful way. Web services should be provided with the functionality of exception handling.

6. Conclusion

With the integration of Web Services as a business solution in many enterprise applications, the QoS of Web Services is becoming the main concern of both service providers and clients. Providers need to specify and guarantee the QoS in their web services to remain competitive and achieve the highest possible revenue from their business while the clients desire to have a good service performance (e.g. high availability, short response time, etc.). The Web services QoS requirements described here do not exhaust the list of yardsticks necessary to measure the quality of service in Web service.

References

[1] D. A. Menascé, "QoS Issues in Web Services", IEEE Internet Computing, pp. 72-75, Dec. 2002, http://csdl.computer.org/comp/mags/ic/2002/06/w6072abs.htm

[2] M. C. Jaeger, G. Rojec-Goldmann, and G. Muhl. Qos aggregation for web service composition using workflow patterns. In *Proc. 8th IEEE Int'l Conf. Enterprise Computing*, pp 149–159, 2004.

[3] J. O'Sullivan, D. Edmond, and A. H. M. ter Hofstede. What's in a service? *Distributed and Parallel Databases*, 12(2/3):117–133, 2002.

[4] M. Ouzzani and A. Bouguettaya. Efficient access to web services. *IEEE Internet Computing*, 8(2):34–44, 2004.

[5] S. Rosario, A. Benveniste, S. Haar, and C. Jard. Probabilistic qos and soft contracts for transaction-based web services orchestrations. *IEEE Trans. Services Computing*, 1(4):187–200, 2008.

[6] N. Thio and S. Karunasekera. Automatic measurement of a qos metric for web service recommendation. In *Proc.Australian Software Engineering Conference*, pp 202-211, 2005.

[7] L. Zeng, B. Benatallah, A. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang. Qos-aware middleware for web services composition. *IEEE Trans. Software Engineering*, 30(5):311–327, 2004.

[8] D. Ardagna and B. Pernici. Adaptive service composition in flexible processes. *IEEE Trans. Software Engineering*, 33(6):369–384, 2007.

[9] M. Alrifai and T. Risse. Combining global optimization with local selection for efficient qos-aware service composition. In *Proc. 18th Int'l Conf. World Wide Web(WWW'09)*, pages 881–890, 2009.

[10] Liangzhao, "Quality Driven Web Services Composition", *The Twelfth International World Wide Web Conference* , Budapest, Hungary, May 2003.

[11] A. Mani and A. Nagarajan, "Understanding quality of service for web services", *IBM paper*, January 2002.

[12] Ying Li, Xiaochen DING, Ying CHEN, Dong LIU, Thomas LI, "The Framework Supporting QoS-enabled Web Services", *IEEE International Conference on Web Services*, Las Vegas, Nevada, USA, June 2003.

[13] Shuping Ran, "A Framework for discovering web services with Desired Quality of Services Attributes", *IEEE International Conference on Web Services*, Las Vegas, Nevada, USA, June 2003.

[14] Shuping Ran, "A Model for Web Services Discovery With QoS", ACM SIGecom Exchanges, 4(1), March 2003

[15] Rajesh Sumra, Arulazi D., "Quality of Service for Web Services-Demystification, Limitations, and Best Practices" March 2003. Available at http://www.developer.com/services/article.php/2027911

[16] IEEE, "IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries", New York, NY, 1990.

[17] W3C QoS for Web Services: Requirements and Possible Approaches. Working Group Note 25 November 2003

[18] Zheng Zhibi (2011) QoS Management of Web Services, An unpublished Phd Thesis submitted to The Chinese University of Hong Kong January, 2011

Kuyoro Shade O. received the B.Sc. and M.Sc. degrees in Computer Science from Olabisi Onabanjo University (2004) and University of Ibadan (2010) respectively. She is an assistant lecturer at Computer Science Department, Babcock University, Ogun State, Nigeria Her research interests are in the area of computer networking, machine learning and artificial intelligence.

Awodele Oludele is a Senior Lecturer at Computer Science Department, Babcock University Ilishan-Remo, Ogun State, Nigeria His research interest is in the area of artificial intelligence.

Akinde Ronke O. is an assistant lecturer in the Department of Computer Science, Babcock University, Ogun State, Nigeria. Her research interest is in the area of software engineering.

Okolie Samuel O. is a Senior Lecturer at Computer Science Department, Babcock University Ilishan-Remo, Ogun State, Nigeria. His research interest is in the area of numerical analysis.