

QOS OF WEB SERVICE: SURVEY ON PERFORMANCE AND SCALABILITY

Ch Ram Mohan Reddy¹, R. V Raghavendra Rao¹

D Evangelin Geetha², T V Suresh Kumar², K Rajani Kanth²

¹B M S College of Engineering, Bangalore, India

²M S Ramaiah Institute of Technology, Bangalore, India

ABSTRACT

In today's scenario, most of the organizations provide the services through the web. This makes the web service an important research area. In addition, early design and building web services, it is necessary to concentrate on the quality of web services. Performance is an important quality attributes that to be considered during the designing of web services. The expected performance can be achieved by proper scheduling of resources and scalability of the system. Scalability is a desirable attribute of a process computer system or network. Poor scalability can result in lacking system performance. Hence, in this paper, we have reviewed the literature available for the quality attributes of performance and scalability and identified the issues that affect the quality attributes related to Web Services.

KEYWORDS

Web Services, Quality of Service, Performance, Scalability, Software Performance Engineering.

1. INTRODUCTION

Distributed computing is a technique which can be invoked simultaneously over a network or internet since the introduction of distributed object technologies such as CORBA which has been introduced by the Object Management Group (OMG) [19]. Distributed Component Object Model (DCOM) was developed by Microsoft and it is the distributed version of Component Object model (COM) and later Remote Method Invocation (RMI) was introduced by Sun Microsystems. However, these architectures have limitations and their performance was not satisfactory on the internet. Services in these environments may serve clients with different requirements. It refers to splitting the application into modules, which are hosted on different network nodes. Nowadays e-commerce applications are distributed and depend on various enterprise services for execution. Developing, programming and integrating these services on different platforms become a complex task. Distributed architectures such as two-tier and three-tier architectures were designed, but they failed to provide hundred percent interoperability. This problem was solved by web services by using web service enhancements. A web service is a reusable component providing services on the web and it functions irrespective of the parties' software and hardware platforms. Web services provide the service interface instead of a user interface. The service interface is an XML document that is used to call the web service. Therefore, this interface plays a major role in the lifecycle of a web service. The lifecycle of a web service defines a scope of the web service and the activities involved to use it. The tasks involved in the life cycle of a Web Service are shown in figure 1:

1. Design a Web Service.
2. Define the service interface and Web Service methods.
3. Create a Web Service and register on some central network node to help the Web Clients in finding and using the service.
4. The web clients have to locate the Web Services on the Web to use them.
5. The Web Service shall be called, by the clients.

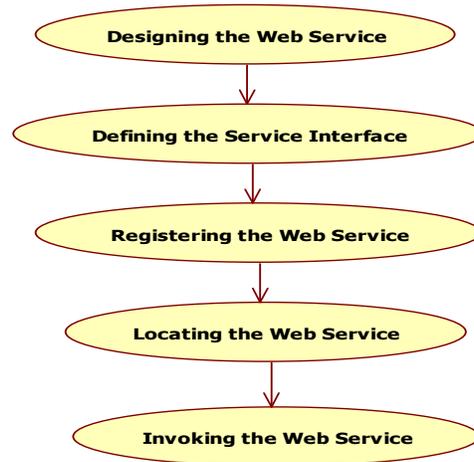


Figure 1. Life Cycle of a Web Service

Web Services are Services or applications that are made available in the Internet. The technologies that constitute Web Service are: extensible Mark-up Language (XML); Simple Object Access Protocol (SOAP)[16], Web Service Description Language (WSDL)[17], Universal Discovery Integration (UDDI)[18]. XML is the de facto standard and has been widely accepted as the universal language for exchanging information between applications across the internet. SOAP enables an exchange of information between two or more peers and enables them to communicate with each other in a decentralized, distributed application environment. WSDL defines a binding mechanism used to attach a protocol, data format, an abstract message or set of endpoints defining the location of services. UDDI defines the standard interfaces and mechanisms for registries intended for publishing and storing interfaces and mechanisms for registries intended for publishing and storing descriptions of network services in terms of XML messages.

In the evolution of Web Services the issues like Performance, Scalability, Scheduling, Capacity, interoperability and usability are considered by some of the researchers. As a consequence of the rapid growth of web services and the abundance of service providers, quality of service is becomingly a key aspect in web services competition. Hence, selecting the right service provider has become a challenge for the consumers. In such scenarios, the quality of service becomes essential to differentiate services and providers. Quality of Service forms the keystone for a broad range of web service activities such as admission control, server selection, scheduling, pricing and specifying service level agreements. The QoS delivered to a customer is highly affected by various factors such as the performance of the web service itself, the hosting platform and the underlying network.

Performance is one of the most important QoS attributes. It is also important to consider the factors that contribute to the effective performance experienced by a various techniques during the analysis of various services provided at the client-side performance. According to Ana C [1]

the performance of Web Services has been analysed in different ways. Use of Web Services integration technology is not justified based on its performance capabilities it is chosen in terms of simplicity, interoperability, flexibility and reuse of services. To reduce its limitations in terms of performance issues in Web Services, many studies have compared the performance of different toolkits.

Scalability is more important characteristics need to finds for the web service to check for the end-to-end user requirements. The increasing popularity of Web Services has serve as an example of the need for scalable and robust discovery mechanisms. Other factors such as variety of communication mechanisms, QoS issues and additional cost make scalability analysis of Web Services more complex. Client/server applications share common resources and as the number of clients (users) increases, the shared resources quickly bottleneck and scalability is affected. For this computing model, the server side of the web service should be very responsive. When scalability is drastically affected by the introduction of web services, some mechanism is to be found for alleviating the problem.

In among the several quality attributes, performance and scalability have prominent impact on the web services. Hence, we have carried out a survey on the literature related quality attributes and presented the review in this paper.

2. SURVEY ON QUALITY ISSUES OF WEB SERVICES

2.1. QoS in Web Services

Web Services applications have the capability of additional features to their applications as part of implementing QOS aspect in their services. There are huge numbers of frameworks in each category to implement QOS in Web Services in the form of non-functional requirements. There are many aspects of QoS important to Web Services. QoS Requirements related to Web Services refers to the quality of both functional and non-functional characteristic of a web service are shown in figure 2. We are starting to organize them into QoS Categories. Each Category needs to have a set of quantifiable parameters or measurements. The QoS attributes of Web Services have the utmost priority for various providers of services because of the unpredictable and dynamic nature of the web services. They include Service time[29], Reliability[3,6,7], Execution price[8,12,22,40],Availability[50],Scalability[27],Accessibility[13,24,28],Performance[9,24,28,33],Capacity[35],Integrity[24,28],Robustness/Flexibility[36,38],Completeness[36,38],Regulatory[24,28,35],Reputation[51]Cost[24,25,30,36],Accuracy[25,29,36],Transaction[28,29,35],Stability [29,36], and Security-Related QoS [9,35,46]. The challenges in Web Services are to enhance the QOS requirements to justify the requirements for quality of Web Services, and that requirement must be relevant to selected Web Services. To know the QoS requirement distinct perspective Web Services gives possible results as per identified and implemented predictions and best practice to improve QOS of Web Services. The QoS attributes are taken from [29, 35, 36] and presented in Table 1.



Figure 2 QOS Categories

Table 1. QoS attributes related to Web Services.

QoS Attribute	Description
Scalability	The throughput improvement rate in a given time interval
Latency	The time taken to start servicing a requested service
Throughput	A service request processing rate
Response time	The time taken between the end of service or demand of a service and the beginning of a response.
Capacity	The number of parallel requests a service allows
Availability	A service is operating time percentage
Reliability	The time for continuity of expected service and for transition to accurate state
Accuracy	The service error rate in a specified time interval.
Robustness	The service flexibility level to inappropriate access and invocation services
Stability	The change rate of service
Cost	The total cost to utilize the service
Security	Specifies the methodologies to protect data
Reliable messaging	Determines if the service offers mechanisms to guarantee reliable message delivery
Integrity	Determines the transactional properties support to the services
Interoperability	Defines if the service is acquiescent with interoperability profiles

2.2. Performance issues in Web Services

Performance is a one of the major problem in Web Services. Web service is a flexible technology for QOS. Web Services are proven that this technology is more adaptable and superior than technologies available currently. Web Services attain success because of its framework, which supports conventional distributed heterogeneous architecture. [48] discusses the real world web services performance and presents various data sets for web service QOS. [32] introduces hibernate technology to improve the performance of java web service by implementing better mapping approach with corresponding server. [4] addresses the issue of web services related to QOS trust metric of service request. This approach assumes that every request is genuine and no anomaly request exists. This work is proposing suitable trust calculation for method for newly

published web services but not for existing ones and yet to define optimal way of configuring these web services. [49] the authors demonstrates the functionality of web services invocation from distributed and heterogeneous environment, but the work is not concentrated on issues like chance of service failure, response time etc...

[23] is proposing a paradigm to create a new web service based on existing services called as Hybrid approach to enhance the performance of a web service. [47] briefs the role of Grid agents and Semantic web technologies to improve the availability, performance and composition of web services. [20] targets the technical phenomena of web services implementation and is found as standard methodology for developing and deploying web services in Java and .Net. [10] explains the major issues in Web services development related to publish, discovery and selection. These events implementing mechanism varies depending on the type of web service. [5] work talks about agent based web service model. It maintains the information regarding the popular was available based on some reputed parameters. This model can be improved further by analysing services at different time slots.

[26] paper submits a semantic – based message interaction model of web services for get connected to various types of web services based on this developer can integrate additional services to generate new value added web services. [37] Web services are proved that this technology is more flexible and superior than technologies available currently. Web services are huge success because of its framework which supports conventional distributed heterogeneous environment. Requests Automated Binding Mechanism [RABM] is proposed to reduce authentication overhead for user requests in [44]. This model packs all similar type of requests and applies authentication process on a whole by invoking corresponding web services. The requests all together will be processed and as a result no service overhead. An analysis based on generic model that is implemented using four different combinations of available technologies on web services is discussed in [21]. It traces out the performance metrics based on time spent for response, network traffic. The work in [2] speaks about QoS in composite web services. This has been achieved by optimal scheduling of service by monitoring the status of the service. Always there is gap between development and requirement of a service. Developer concentrate more on issues expected to occur at the time of development and implementation, but not at the time of user request. [45] describes the meta model, which fills this gap and makes it more reliable. Blue Pencil framework is suggested to improve the performance of web service by automating the process of checking the availability and accessibility properties of bindings at deployment of service in [42].

To improve the performance of Composite Web Service scheduling policies are suggested in [11] that can able to fix the execution of complex business process. This paper does not address the issues like implementing optimal policy patterns for Composite Web Services (CWS). [14] provides information about various technologies available to develop web services and the purpose of using those technologies based on their requirement. Web services latency based on simple performance model is observed and found accurate according to the requirement in [39]. [34] illustrates a solution called Chain of Adapter to resolve the issue related to backward compatibility of web Services. In this model author identified a trade-off of response time. [31], concentrates on Web Services security with single window model. This work suggests Kerberos protocol, which extends support by delegation access privileges. Delegation can avail services temporary basis without multiple authentication & authorization process. [15] talks about performance overhead of web Services while invoking a Web Services on large business process. This work proposes optimization of proxies to avoid performance overhead expected to occur in conventional invocation of Web Services. In [52] the work redefines the role of standard activities like publish, discover and delivery services while handling non-functional requirements.

2.3. Scalability of Web Services

Web services provide application-to-application integration and interoperability across different platforms. However, users usually point to an inefficient processing model and bandwidth requirements for developing Web services. This is often considered as a reason why Web services cannot perform and scale well in development environments. While designing the web service, the following things need to be considered. It needs to meet the initial requirements and should grow rapidly to meet the user and organization requirements. Many people have done research on achieving performance and scalability of web service and come up with different solutions we will have a look at the different solutions.

[34] describes a scalable method for the QoS-based service selection. The proposed solution allows to reducing the efforts compared to existing solutions available to a greater extent. By dividing the problem into number of sub-problems that can easily be solved by using local search within each service class. Downsizing the problem causes to better utilize the available processing power while increasing the availability of service information within a service infrastructure. This approach is for a limited set of architectural requirements so it can be easily integrated in a wide range of service oriented infrastructures.

Although different solutions for discovering Web Services promise to fulfil these needs, most of the solutions make assumptions with limitations concerning the number of nodes and the topology of the network and rely on having information on the data categorizations or popularity distributions. In addition, most systems are verified via simulations using artificial datasets. A lightweight, accessible and tough WSDL discovery mechanism based on real-time calculation of term popularity in [18]. Through the experiment firstly he advanced slowly, parsed, extracted and analysed the largest WebService description of collected information, to the best of our knowledge, with the benefit of having rich descriptions for WSDL files without the need for additional annotation. Secondly, he developed popularity as a counter-indication for selectivity. Thirdly, he implemented a system using the aforementioned algorithm and has performed extensive empirical evaluation using real data on an implementation running on a large number of peers. Finally, this approach showed important presentation gain, compared to two situation methods.

Mashups, while providing improved Web personalization, pose new scalability and performance challenges. Traditional Web service performance schemes are not effective for mashup domain due to its unique characteristics. [46] Proposed common component detection scheme which is used to reduce delay resulting from executing repeated mashup components.

3. CONCLUSION & FUTURE WORK

From the survey, we have understood that performance and scalability are the two attributes that play important role in the quality of web services. Hence the following activities are to be carried out to build a responsive web service.

1. The performance measurement on the client side where the performance dynamically changes overtime.
2. Understands the behaviour of the activities in a composed process through the performance evaluation of web services.
3. Compare the performance of different web services at early stages.
4. Develop algorithms to predict the performance of the best web services.
5. Estimation of performance parameters early in the web service life cycle.

6. Simulating and predicting the performance of web services considering different service time, deployment environment, etc.

Software Performance Engineering (SPE) is a methodology to assess the performance of software systems early (analysis phase) in the life cycle [43]. SPE continues through the thorough strategy, coding and testing stages to calculate and succeed the performance of the developing software and to monitor, report actual presentation against specifications and calculations. SPE is important for software engineering and in particular for software quality. The Software Performance Engineering Practice uses several presentation assessment tools depending on the state of the software and the amount of performance data available. In future, we propose model the performance characteristics of web services and develop algorithms to evaluate the performance of web services using SPE approach.

REFERENCES

- [1] Ana C. C. Machado, Carlos A. G. Ferraz (2005), "Guidelines for Performance Evaluation of Web Services", WebMedia '05, December 5-7, 2005, Pocos de Caldas, MG, Brazil. ACM.
- [2] An Liu, Qing Li, Liusheng Huang, Mingjun Xiao, and Hai Liu (2008)," QoS-Aware Scheduling of Web Services", the Ninth International Conference on Web-Age Information Management, DOI 10.1109/WAIM.2008.37, IEEE.
- [3] Burstein M., Bussler C., Finin T., Huhns, M.N., Paolucci M., Sheth A.P., Williams S. and Singh M.P (2005). "A Semantic Web Services Architecture", IEEE, Internet Computing, pp. 72-81.
- [4] Bixin Li, Rui Song, Li Liao, Cuicui Liu (2013), "A User-Oriented Trust Model for Web Services",Seventh International Symposium on Service-Oriented System Engineering, DOI 10.1109/SOSE.2013.17, IEEE.
- [5] Babak Khosravifar, Jamal Bentahar, Ahmad Moazin (2010), "Analyzing the Relationships betweenSome Parameters of Web Services Reputation". IEEE International Conference on Web Services,DOI 10.1109/ICWS.2010.88, IEEE.
- [6] Chen Z., Chia L.T., Silverajan B. and Lee B.S (2003), "UX- An Architecture Providing QoS –Aware and Federated Support for UDDI", Proceeding of the 1st International Conference on Web Services, pp.1-6.
- [7] Cordoso J. (2004) "Modeling Quality of services for Workflows and Web Services Processes", Technical Report, LSDIS Lab, Computer Sciences Department, The University of Georgia.
- [8] Chen M., Chen A.N.K. and Shao B.B.M (2003) "The Implications and Impacts of Web Services to Electronic Commerce Research and Practices", Journal of Electronic Commerce Research, 4(4):128-139.
- [9] D'Ambrogio A (2006) "A Model-driven WSDL Extension for Describing the QoS of WebServices", IEEE International Conference on Web Services.
- [10] Demian Antony D'Mello, Ananthanarayana V. S., V. Lakshmi Narasimhan (2010)," Challenges (Research Issues) in Web Services, International Conference on Advances in Computer Engineering", DOI 10.1109/ACE.2010.75, IEEE.
- [11] Dmytro Dyachuk, Ralph Deters (2007), "Improving Performance of Composite Web Services",International Conference on Service-Oriented Computing and Applications (SOCA'07, IEEE.
- [12] Fung C.K., Hung P.C.K., Linger R.C., Wang G. and Walton G.H (2006). "A Service-Oriented Composition Framework with QoS Management", International Journal of Web Services Research, 3(3), pp. 108 -132.
- [13] Gu X., Nahrstedt K., Yuan W. and Wichadakul D (2002). "An XML-based Quality of Service Enabling Language for the Web", Journal of Visual Languages and Computing, pp 1-39.
- [14] Hai Huang Rick (2006), "A. Mason, Model Checking Technologies for Web Services", Proceedings of the Fourth IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems and Second International Workshop on Collaborative Computing, Integration, and Assurance (SEUS-WCCIA'06), IEEE.
- [15] Hans Albrecht Schmid (2004), "Performance Problems of Large Operational Systems Based on Web Services and a Solution", Proceedings of the International Conference on Services Computing (SCC'04), IEEE.

- [16] <http://www.w3.org/TR/soap12-part1/>, Last visited on 2013.
- [17] <http://www.w3.org/TR/wsdl12/>, Last visited on 2013.
- [18] http://uddi.org/pubs/uddi_v3.htm, Last visited on 2013.
- [19] <http://www.omg.org/cgi-bin/doc?ccm/2002-04-01>, Last visited on 2013.
- [20] J. Matkovi, K. Fertalj (2010) "Comparative analysis of Web services and Web service development technologies", MIPRO 2010, May 24-28, Opatija, Croatia.
- [21] K.Velmurugan, M.A.Maluk Mohamed (2009), "An Empirical Performance Metrics Measurement and Analysis of Software Platforms for Implementation of Web Services", DOI 10.1109 /ICIS. 194, IEEE.
- [22] Liu Y., Ngu A.H.H. and Zeng L., (2006), QoS Computation and Policing in Dynamic Web Service Selection, Retrieved September 3, 2006, from: http://www.ra.ethz.ch/CDstore/www_2004/docs/2p66.pdf.
- [23] Leslie Cheung, Leana Golubchik, Fei Sha (2011), "A Study of Web Services Performance Prediction: A Client's Perspective", 19th Annual IEEE International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems, DOI 10.1109 /MASCOTS.2011.66,IEEE.
- [24] Looker N., Munro M. and Xu J (2004), "Simulating Errors in Web Services", I.J. of Simulation, 5(5), pp 29-37.
- [25] Ludwig H (2004), "Web services QoS: external SLAs and internal policies or: how do we deliver what we Promise?", In Proc. of the Int'l Conf. on Web Information Systems Engineering Workshops, Pages 115–120, Springer.
- [26] Laomo Zhangl Ying Mal Guodong Wang (2010), "A Semantic-based Message Interaction Model and Application of Web Services", International Conference on Computer and Communication Technologies in Agriculture Engineering, IEEE.
- [27] Marin Litoiu (2002), "Migrating to Web Services - Latency and Scalability", Proceedings of the Fourth International Workshop on Web Site Evolution (WSE'02), IEEE.
- [28] Mani A and Nagarajan A (2002.)," Understanding quality of services for web services", [www-106.Ibm.com /developerworks /webservices/library/ws-quality.html](http://www.ibm.com/developerworks/webservices/library/ws-quality.html) IBM Developer Works.
- [29] Menasce D.A (2002), "QoS Issues in Web Services", IEEE Internet Computing, 6(6), ISSN: 1089-7801. pp. 72-75.
- [30] Maximilien E. M. and Singh M. P (2004), "A framework and ontology for dynamic Web services Selection". IEEE Internet Computing, 8(5):84–93.
- [31] Markus Hillenbrand, Joachim Götze, Jochen Müller, Paul Müller (2005)," A Single Sign-On Framework for Web-Services-based Distributed Applications", 8th International Conference on Telecommunications – ConTEL, Zagreb, Croatia, , ISBN: 953-184-081-4, June 15-17.
- [32] Nitin Kumar Verma, Dr. Neeta Singh, Rishi Gautam (2013)," performance analysis of web services by using j meter performance tool", International Journal of Engineering And Computer Science ISSN:2319-7242 Volume 2 Issue, Page No. 1449-1452.
- [33] Papazoglou and Georgakopoulos (2003):" Service-oriented computing: Introduction", Communication of the ACM, pp 25-28.
- [34] Piotr Kaminski, Marin Litoiu, Hausi Müller (2006), "A Design Technique for Evolving Web Services", Piotr Kaminski, Hausi Müller, and IBM Canada Ltd.
- [35] Ran S (2003), "A Model for Web Services Discovery with QoS", ACM Inc., 4(1), pp. 1-10, ISSN 15519031.
- [36] Ran S (2003), "A framework for discovering Web services with desired quality of services Attributes". In Proc. of the Int'l Conf. on Web Services, pages 208–213. CSREA Press.
- [37] Rohit Dhand (2009), "Web Services: A Trend Shift from conventional Distributed Computing Model", Second International Conference on Computer and Electrical Engineering, DOI 10.1109/ ICCEE, 2009.26, IEEE.
- [38] Shuping Ran (2003), "A Model for Web Services Discovery with QoS", ACM Inc., ISSN 15519031, 4(1), pp. 1-10.
- [39] Shiping Chen, Bo Yan, John Zic, Ren Liu, Alex Ng (2006), Evaluation and Modeling of Web Services Performance, IEEE International Conference on Web Services (ICWS'06), IEEE.
- [40] Sivashanmugam K., Miller J.A., Sheth A.P and Verma K. (2005), "Framework for Semantic Web Process Composition", International Journal of Electronic Commerce/winter 9(2), pp. 71-106.
- [41] Senthilanand Chandrasekaran, John A. Miller, Gregory S. Silver, Budak Arpinar, Amit P. Sheth, (2006) "Composition, Performance Analysis and Simulation of Web Services".

- [42] SangJeong Lee, Kyung Dong Ryu, Kang-Won Lee, Jong-Deok Choi (2007), "Improving the Performance of Web Services Using Deployment-Time Binding Selection", IEEE International Conference on Web Services (ICWS 2007), IEEE.
- [43] Smith C.u (1990). "Performance Engineering of Software Systems", Reading, MA, Addison Wesley.
- [44] Tong Yizhu, Wang Hao, Liu Taoying (2009), "Exploiting Request Characteristics to Improve Performance of Web Services Based Business Applications", International Conference on Advanced Information Networking and Applications, DOI 10.1109/AINA.2009.103, IEEE.
- [45] Wan Nurhayati WAN AB. RAHMAN and Dr Farid MEZIANE (2008), "Challenges to Describe QoS Requirements for Web Services Quality Prediction to Support Web Services Interoperability in Electronic Commerce", Communications of the IBIMA, Volume 4.
- [46] Wang G., Chen A., Wang C., Fung C. and Uczekaj S (2004). "Integrated Quality of Service (QoS)Management in Service-Oriented Enterprise Architectures", Proceedings of the 8th IEEE International Enterprise Distributed Object Computing Conference, pp. 21-32, ISSN 1541-7719, ISBN 0-7695-2214-9, INSPEC Accession Number 8314546.
- [47] Xia Zhao,Tao Wang, Enjie Liu, Gordon J. Clapworthy (2010), "Web Services in Distributed Information Systems: Availability, Performance and Composition", Web Services in Distributed Information Systems, IGI Global.
- [48] Zibin Zheng, Yilei Zhang, and Michael R. Lyu (2010), "Distributed QoS Evaluation for Real-World Web Services", IEEE Intl Conference on Web Services, DOI 10.1109 /ICWS .2010.10, IEEE.
- [49] Zibin Zheng, Yilei Zhang, Michael R. Lyu (2012),"Investigating QoS of Real-World Web Services", IEEE TRANSACTIONS ON SERVICE COMPUTING,1939-1374/12/\$31.00, IEEE.
- [50] Zeng L, Benatallah B, Dumas M, Kalagnanam J. and Sheng Q.Z (2005), "Quality Driven Web Services Composition", Retrieved July 1, 2005, from: [www2003.org/cdrom/papers/refereed /p358 /htm /p358- zeng.html](http://www2003.org/cdrom/papers/refereed/p358/htm/p358-zeng.html).
- [51] Zeng L., Benatallah B., Lei H., Ngu A., Flaxer D. and Chang H (2003). "Flexible Composition of Enterprise Web Services", Electronic Markets, 13(2), pp. 141-152.
- [52] Zafar U. Singhera (2006), "Extended Web Services Framework to Meet Non-Functional Requirements", ICWE'06 Workshops, July 10-14, 2006, Palo Alto, CA, ACM.