International Journal of Computer Sciences and Engineering Open Access
Review Paper Vol.-6, Issue-7, July 2018 E-ISSN: 2347-2693

A Systematic Literature Review on QoS for SOA-based Web Services

A. **Negi[1*]**, **P. Kaur[2]**

[1]Department of Computer Science, Guru Nanak Dev University, Amritsar, Punjab, India
[2]Department of Computer Science, Guru Nanak Dev University, Amritsar, Punjab, India

[*]Corresponding Author: aradhana.csersh@gndu.ac.in, Tel.: +91-75089-34186

Available online at: www.ijcseonline.org

Accepted: 25/Jul/2018, Published: 31/Jul/2018

Abstract— As the espousal pace of Web Service technology has increased, so does the requirement for efficient Web Service development such as competent mechanism for discovery, monitoring, and composition etc. are obvious. In our opinion, the efficacy of Web Service development can be achieved only if the potency of two facets i.e. 1) *Quality Attributes along with the functional requirements of Web Services* and 2) *advantages of the foundational architecture of Web Services i.e. Service-Oriented Architecture* are recognized at their peak. For Web Services operated over the heterogeneous widespread network, the Quality Attributes are of principal importance, especially, when selecting one service out of many similar services. Till date, several Quality Attribute and their measuring methods have been published in the literature for SOA-based Web Services but none of them have discussed the Quality Attributes, their available different system of measurements, tradeoffs, tools, and standards altogether. Here, this paper presents a Systematic Literature Review on Quality-of-Service for Service-Oriented Architecture-based Web Services by addressing six significant research questions using a fine review protocol. This paper reviews the varied definitions, metrics, issues and challenges, standards and future directions of Quality-of-Service attributes for SOA based Web Services.

Keywords—Quality Attribute; Quality Metric; Quality-of-Service (QoS); Service-Oriented Architecture (SOA); Web Services

I. INTRODUCTION

Service-Oriented Architecture (SOA) is a mode for scheming, generating, deploying and controlling systems [1]. A system following SOA is composed of several services, being each one responsible for accomplishing a specific and finer-granular operation. The Web Service technology is considered as the most promising preference to implement SOA also these Web Services binds the distributed heterogeneous system with one another by means of the Internet. The complex business processes are aligned with information technology through SOA [2] and handover their application's functionality (to the Service Consumer/another service) in the form of service.

W3C defined Web Services as "[...] a software system de-signed to support interoperable machine-to-machine interaction over a network. It has an interface described in a ma-chine-process able format (specifically WSDL). Other systems interact with the Web Services in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards" [3]. In addition to the above-stated protocol Web Services uses UDDI (Universal Description, Discovery, and Integration), and WSDL (Web Service Description Language) open standards for integrating Web-based applications.

The Web Services are measured against functional metrics and non-functional metrics i.e. quality aspect of Web Service metrics. Quality Attributes are the source of differentiating alike Web Services. ITU (International Tele Communication Union) [4] define Quality as "the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs" and technology giant Microsoft [5] defines Quality Attributes as "factors that affect run-time behavior, system design, and user experience". The sum of all Quality Attributes of a service at a given time is known as Quality-of-Service (QoS). This paper makes an attempt to present a systematic and disciplined way to explore QoS for SOA based Web Services by following the Systematic Literature Review (SLR) guidelines given by Kitchenham et al. [6]. The rest of the writing is ordered as three sections. Section II discusses research methodology in parts i.e. planning, conducting, and reporting the review along with answers of six research questions. Section II.C.1 presents the answer of the first two research questions given in the subsection of planning the review. In Section II.C.2, the answers for research question number third and fourth have been discussed. The tradeoffs and research gaps of QoS for SOA-based Web Services in Section II.C.3, answers the last two research questions. Finally, the whole review is remarked in conclusion in Section III.

II. RESEARCH METHODOLOGY

The research methodology chosen for systematic review is significant for ensuring the correctness of the State-of-the-Art. This division discusses research methodology in three subsections i.e. Planning, Conducting, and Reporting the review.

A. Planning the Review

1) Identification of need

- To summarize the existing related work of QoS for SOA-based Web Services.
- To provide a background for positioning new research activities by identifying research gaps.

2) Research Question (RQ)

The research question drives the whole systematic review by means of controlling its scope. The answer to each RQ put light on a specific area and represents concepts in an organized manner. In Section II.C.1, answers to the following two RQs are presented with definitions, diagram, and tables.

RQ.1 What are the designated definitions of QoS for SOA-based Web Services?

RQ.2 What sort of responsibilities Web Service Model entities performs for QoS?

Based on literature, the section II.C.2 describes the answers to RQ.3 and RQ.4 respectively.

RQ.3 What are the available QoS Attributes and their respective metrics?

RQ.4 What are the available standards for QoS Attributes?

The answer to succeeding questions explains very noteworthy field of QoS for SOA-based Web Services. Answers to these questions conjointly illustrate the critical aspect in this area.

RQ.5 What are the tradeoffs between QoS, SOA and Web Services?

RQ.6 What are the main research challenges associated with QoS of SOA based Web Service?

3) Review Protocol

An extensive search has been done with the wide range of online academic references including journals, conference proceeding and working papers. Books, technical reports, master thesis, Ph.D. thesis and working draft are also included in this study. QoS is acquired as a constraint while searching for SOA and Web Services.

4) Searched Keywords

Keywords are searched using AND and OR operator. The literature with exact matched keywords like (QoS OR 'Quality of Service') AND SOA AND Web Service, QoS AND Web Service AND Review OR (SLR OR Systematic Literature Review), QoS Attributes AND Web Service, QoS Attributes AND SOA, QoS metrics AND SOA AND Web Service is selected. Some other keywords based on individual RQ are also searched e.g. Define QoS for Web Services, "Issues and Challenges OR Confronts" AND SOA AND Web Services.

5) Search Engine database

Databases such as Science Direct (Elsevier), SCOPUS, Springer, Web of Science, ACM, and IEEE have been searched for results. These databases cover almost all the top level qualitative journal and conference papers. Because every found article was not relevant to aforementioned RQs so to go through all research articles from all the above-stated sources was a very deadening job. To handle the above deadening problem, papers are selected on the basis of some filters. The relevant paper selection and quality assessment criterion is covered in next sub-sections.

B. Conducting the Review

1) Identification of the Research

The searched keywords are explored using automatic and manual search. The AND, OR operators are found sufficient to get good quality results. RQ has been broken down in terms of a list of synonyms, and abbreviations. Combinations of these terms are also taken into account for obtaining genuine answers.

2) Study Selection and Quality Assessment

Three filters have been used (from journals, conferences, technical reports etc.) for inclusion and exclusion of research material. The filters are 1) Title, 2) Abstract, and 3) Introduction along with year constraint (2001-2015).

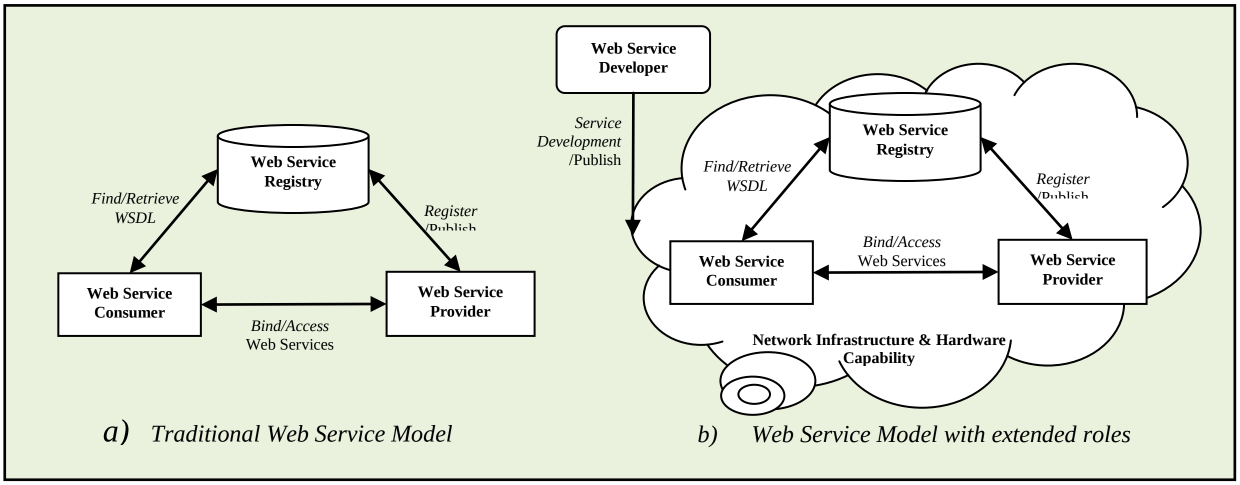

Fig. 1: Web Service Model

Under this selection criterion, the papers which have direct evidence of answers are considered in the first filter while for other papers, the reading of the abstract and introduction reveals their worth. Initially, RQs search results in 485 papers whereas after applying the filters only 48 papers are found pertinent. The final 48 papers are studied thoroughly and the quality of answers is checked simultaneously to confirm its validity. Numeric data, meta-analysis, figures, and tables from the selected material are also kept in mind for supporting the soundness of review. The reference section of the final selected papers is also considered to improve the quality of the paper.

C. Reporting the Review

The forthcoming three sub-sections and their subsections report the review by presenting answers to all RQs.

1) QoS for Web Services

Literature introduces the definition of QoS for Web Service in an ad-hoc manner and these definitions varies extensively in stated-concepts. The term 'Quality-of-Service' basically adopted from network/telecommunication research community [7] and defined as the ability to provide superior service over various technologies. Later on, in the year 2000, some research efforts brought this concept in distributed systems [8] to express its quality conception. After usage as QoS in distributed systems, QoS became relevant research field for Web Services. Table 1 point towards the popular definitions of QoS for SOA-based Web Service. The quality related notion is very well described by Oriol [9]. Nowaday, Web Services are being served by several Service Providers and many cases occur, where various Web Services can gratify operational needs of a Service Requester, but what resolves the matter of selecting the finest Web Service for Service Requester is the QoS level of Web Services. With affirmed QoS, customers get "what they really expect for their paid electronic solutions" [10] while Service Providers "[…] can prepare effective resources planning and avoid from over-committing the resources to certain services" [11].

Table 1: Definitions of QoS for SOA based Web Services

Def.	Definition + Source
1	"Quality-of-Service as a set of non-functional attributes that may impact the quality of the service offered by a Web Services" + [12]
2	"A set of non-functional attributes of the entities used in the path from a Web Services repository to the Service Consumer, who relies on the Web Services' ability to satisfy its stated or implied needs in an end-to-end fashion" + [13]
3	"A whole range of techniques that match the needs of service requestors with those of the Service Provider's based on the network resources available, it refer to non-functional properties of Web Service such as performance, reliability, availability, and security" + [14]
4	"The non-functional QoS facet describes how good the service performs the tasks, this leads towards the aspect of quality" + [15]
5	"QoS information is used for computing the quality degree of candidate Web Service" + [16]
6	QoS is a measure for how well a service serves the customer + [17]
7	"It is an agreement between the Service Consumer and the Service Provider, which is used to express the level of quality that is accepted by both Service Provider and Service Consumer" + [18]

a) Definition Analysis

In Table 1, Def.1 shows uncertainty for the impact of QoS on quality of a Web Service, this definition criticizes the basic nature and purpose of Quality Attribute, so it cannot be considered as a good definition of QoS for Web Services. In Def No.2, QoS is defined as dependent attributes on Web Serviceability and used only on the route between Service Repository and Service Consumer. Here, in this definition (Def. 2) the actual entity which provides service is missing and by adopting this definition, the QoS monitoring cannot happen because stated needs can only be verified when delivered qualities are captured and monitored. QoS is described as the range of techniques in Def No. 3, where the need of Service Requester and Service Provider matches with one another. It can be considered as a good definition but not a complete one.

The next three definitions i.e. Def. 4, 5, and 6, define QoS in general terms without specifying the approach of ensuring Web Service quality. The Def. 7 talks about QoS as an agreement to show the level of quality for a service which is acceptable by both Service Consumer and Service Provider of the service. The range of definitions presented in Table 1, implies that the lack of standard definition of QoS for Web Services subsist but the core idea of QoS is persisted i.e. the optimal level of QoS satisfies Service Consumer and maintenance of ranking of Service Provider. By combining these definitions one can say that QoS is an agreed-upon strategy that ensures the quality of a Web Service by confirming the quality delivered through Service Provider with the quality of Web Service in Service Repository by means of Quality Attribute metrics of like security, efficiency, reliability, and other similar feature of a service.

For the better understanding of QoS for SOA-based Web Services, the responsibilities of entities in Web Service Model must be discussed from QoS perspective. Hence, the responsibilities of three main entities/roles in Web Service model is discussed along with two implicit entities. The traditional Web Services model given in Fig. 1: (a)), describes only three main roles- 1) Service Consumer 2) Service Provider, and 3) Service Registry whereas behind the scene more than three entities work together to decide QoS of Web Services i.e. Web Service Developer and Network Infrastructure as given in Fig. 1: (b). Although the role, Web Service Developer is unseen, but most important tasks like structure designing and optimal implementation strategies are defined by this facet only. Therefore, one dedicated column is presented for this role and its respective responsibility in Table 2. For network infrastructure, an operating system and middleware are responsible but the researchers cannot ignore the impact of network conditions and hardware capability of host device over the QoS of Web Services. This is the reason, why the network infrastructure and hardware capabilities have included as an essential entity in Table 2 (see column no. 6). There are some others entities proposed in literature for Web Services model e.g. involvement of third-party Service Provider (who offer various services to support the utility of Service Provider), QoS Certifier, Ranker, QoS Broker etc. but this paper does not consider these additional entities because of their variations and low acceptance rate in literature.

Table 2: Responsibilities of Web Service Model entities for QoS

Roles	Responsibilities
Web Service Provider	• Service Description, Service Publication, Service Registration, Service advertisement [16] • Service Handling, Regulation [14] • Load Balancing [19] • Service-Level Agreement [20][21] • QoS policy and Decision making [20] • Mapping between Global & Local SLAs [22] • Service availability [23] • Capacity detection or Scaling [24] • Priority-based admission control mechanism • Request-Traffic Management &Caching
Web Service Developer	• Service Development, Service Maintenance, Code-Function Reusability [13] • Designing a structure with QoS [25] • Exception handling mechanism [12][24] • Static service composition [26] • Stability, Testability, Analyzability, Changeability [27] • Service interface stability, Method signature stability [28], Interoperability [25][30]
Web Service Consumer	• Service Discovery, Locate Web Service, Service invocation [16] • Usability Identification [31][32] • SLA understanding and acceptance [33]
Web Service Registry	• WSDL Management [20] and Registry search function [16] • Updating service profile with time
Network Infrastructure & Hardware Capability	• Service Infrastructure , Network Infrastructure and basic network-level QoS parameters [13] • Network delay, delay variation, and packet loss [24] • Transactional service Management [34] • Transmission control and Hardware Capabilities of hosting devices

2) QoS for Web service: Attributes, Metrics, and Standards

Academicians and industry experts have proposed several Web Services Quality Attributes and their system of measurements. To answer the question "What are the available QoS Attributes and their respective metrics?", the first half of this segment tabularized the available Quality Attributes and their respective metrics. Metrics basically quantifies the features of a component/subject and in our case, components are Web Services. The QoS is calculated on the behalf of associated value of an attribute of a service at a given time. In literature, a range of Quality Attributes classifications based on different criterion like ISO/IEC 9126, ISO/IEC 25010:2011, Kim & Lee [25] have been presented. Some of these classifications are directly applicable to Web Services whereas some are indirectly or not applicable to Web Services e.g. installability attribute is not applicable for Web Services. A systematic mapping of 47 different proposals from 65 papers (2001 to 2012) of Web Services Quality models has been done by in 2014 by Oriol et al.[23]. This study has identified three factors i.e. the most consolidated quality factors, the most influencing quality factors, and the most influenced quality factor. Instead of considering a particular quality model, the current paper has considered all established Quality Attributes with their metrics. Table 3 shows the Quality Attributes and their various systems of measurements. Table 3 also depicts that some of the Quality Attribute have many measurement ways e.g. Reliability whereas some are having very few metrics for measurement e.g. Testability, Accountability etc. This depiction signifies that Quality Attribute having fewer metrics needs to be re-analyzed for the better evaluation system.

Table 3: QoS Attributes and respective Metrics

QoS Attribute: Metrics for QoS Attributes
Performance: • Throughput [13][14][21][24][25] • Latency time [35][36] • Connection latency, Request latency, Execution time, WIPS(Web interaction per second) [35] • Response time [12][13][21][25][35][36] • Jitter [36] • Delay time = (Transmission time + Transaction time) [37] • Bandwidth = Number of bytes per second supported by a network [38] • Waiting time experience by Service Consumer, constrained by network [39]
Stability: • Number of emerged adverse impacts in the system after modification/ Total number of modifications made) [27] • Measure of frequency of change related to the service in terms of its interface and/or implementation [40]
Accountability • Service support for transaction(SST) = NTAS*/(NTAS+TAS**) [41] [*NTAS= Non-Transaction Aware Services **TAS =Transaction Aware Services]
Reusability • Effective size calculation for reusability = (existing size × (0.4 × redesign% + 0.25 × reimplementation % + 0.35 × retest %)) [42] • Reusability = Readability value + Publicity value + Coverage of variability + Commonability features value [43]
Regulatory • Measure how well the service is aligned with regulations [40]
Accuracy • Number of error produced by the service [12][40] • IF Standard Deviation (SD) of the errors produced by the service is equal to 0 then accurate, otherwise not [16] • Number of Response messages/ Number of Request messages (Successability) [25]
Security • Number of successful authorized attempts, Number of successful unauthorized attempts, Standards followed for encryption, Access to Log, Number of non-denial requested service, Standard followed for access modify the data [40]
Interoperability • Total number of environments the Web Services run / Total number of environments that can be used [16] • Numbers of service deal/interact with different client/services using different language/platforms [40]
Scalability • Performance Non-Scalability Likelihood (PNL) metric [16] • Stateless service (SS) = SLS (Stateless)/ (SLS(Stateless) + SFS (Stateful)) [41] • Totality of higher number* of requests/ clients and operations/transactions in a given time interval [25] (*more than capacity) • Number of concurrent request for guaranteed performance(Capacity) [40]
Robustness • Degree to which a service can function correctly in the presence of invalid, incomplete or conflict input [12][24]
Exception handling • Number of exceptions handled by the service over a given period of time [40]
Reliability • MTBF*(Mean Time Between Failures)*, MTF*(Mean Time To Failure,* MTTT *(Mean Time To Transition)* [12] • MTBF*(Mean Time Between Failures)* = MTTF*(Mean Time To Failure)*+ MTTR *(Mean Time To* Repair*)*, FIT(Failure-in-time) [44] • Time-to-repair [40][45] • Number of ordered delivery of service, Number of assured delivery of service, Number of answered client request , Number of successful performed service [40]

- Error(s) rate [25][32]
- Number of Successfully used Web Services per month [46]
- E (i, j) = M-R-t (where E is an edge between two services i and j, M is the maximum expected recovery time, R is the actual recovery time and t is the expected execution time of service j. Here j Web Service is to be executing after completion if service i). If the value of E(i, j) is negative then a check point is inserted [47]

Traceability

- Traceability refers to the possibility that a developer or Service Provider can trace history of a service in the log collection system when a request was served [24][40]

Discoverability

- Number of appropriate Interface/Total number of interfaces to discover [48]
- Dynamic Vs. Static Service Selection(DSSS) = DS(Dynamically Selected)/(DS(Dynamically selected Services)+SS(Statistically selected Services)) [44]

Availability

- (MTTF(*Mean Time To Failure*)/(MTTF(*Mean Time To Failure*) + MTTR *(Mean Time To Repair*)))*100% [45]
- MTTR(*Mean Time To* Repair) [28][36], UpTime [28]
- Availability= <Uptime>/(<Uptime> + <Downtime>) [12][32]
- Down Time [16]
- 1-(Down Time/Up Time) [25]

Usability

- Usability determined by Users' experience [49] and User satisfaction [50]
- It is measured usability form the syntactic completeness of a Web Services interface and the well-described semantic elements in WSDL [49]
- Reputation = (The sum of end user ranking on service's reputation/the number of time it has been assessed) [51]

Testability

- Testability is an ability of developer to test the service system operation [51], testability is subjective to developer's test case and testing methodology

Cost

- Execution fee of a Service [32]

Accessibility

- Measure of the success rate of a service instantiation at a given time [16]
- Service Access Method (SAM)= VAS (Virtualized Access Service)/(VAS(Virtualized Access Service) + PPS (Point to Point Service)) [44]
- Accessibility = Number of Acknowledgements received / Total number of request sent [25][37]

Integrity

- Number of successful transaction/Total number of transaction [16]
- Number of new service being added, Number of existing extended services without changing the interface, Number of existing extended services with changing the interface, Extent for prevention unauthorized access [40]

In the second half of this segment, available tools and standards are described in a tabulated outline which answers to the question "What are the available standards for QoS Attributes?". Standard Bodies like W3C (World Wide Web Consortium), OASIS (Organization for the achievement of Structured Information Standards), WS-I (Web Services Interoperability Organization), IETF (Internet Engineering Task Force), Java Community Process and SOAPBuilders Community for Interoperability are currently actively working for SOA, Web Services, and QoS. These bodies deal with two generations of Web Services - 1) First-Generation Web Service Platform 2) Second-generation Web Service Platform (WS-* extensions). Presently, there are more than one standard for a particular QoS attribute e.g. for Security the standards such as trust, privacy, authentication, authorization are available. Table 4 shows the standards/ tools for QoS Attributes.

Table 4: Tools/Standards for QoS Attributes

QoS Attributes	Tools/Standards
Reliability	• WS-Reliability • WS-ReliableMessaging • BTP (OASIS) • BPEL
Reusability	• WS-BPEL(Orchestration) • WS-CDL(Choreography) • WS-CI (Choreography interface) • ebXML • OWL-S • WSMF(Web Services Modeling Framework) • BPML
Security	• WS-Security • WS-Trust • WS-SecureConversaition • WS-Federation • WS-SecurePolicy • WS-Authorization • WS-Privacy • XACML • SAML • XML-Encryption • XML-Signature • XML Key Management ○ XKISS ○ XKRSS
Interoperability	• OASIS WS-I (It has finalized the Basic Profile, Attachments Profile and Simple SOAP Binding Profile. Work on a Basic Security Profile is currently underway.) • WS-I Deliverables
Messaging	• WS-Event • WS-Notification
Transaction	• WS-CAF • WS-Context • WS-CoordinationFramework • WS-TransactionManagement • WS-TX(Microsoft, IBM, BE and others) • WS-Coordination • WS-Atomic Transaction • WS-Business activity specification

Metadata Management	• XML Schema • WSDL • WS-Addressing • WS-Policy • WS-MetadataExchange

3) QoS for Web service: Tradeoffs and Hotspots

a) Tradeoffs

Acclivity of one Quality Attribute can drop another Quality Attribute e.g. elevation of Modifiability increases Reliability and Scalability whereas another attribute like Performance may go down, so it is not realistic to recognize all QoS Attributes together at a particular time. A comparative analysis of such trade-off for QoS in SOA is presented by Bashir et al. in [52], where few mostly used QoS Attributes for e-government are identified and then treated as goals, and lastly with decomposition policy, the compromised relationships are recognized between selected QoS. This section tries to cover the positive and negative interrelationship between SOA-based Web Services on individual Quality Attribute one by one.

- The main aim of SOA-based Web Services is to provide application functionality of software as a service on the internet, where any system (also of different platform) can use that service by fulfilling its invocation requirements. Providing services in distributive environment brings Interoperability attribute automatically in the frame of SOA-based Web Services. Interoperability using Web Service is purely syntactic WSDL [19] and Interoperability lacks when a UDDI registry uses different policies on an acceptable announcement of Web Services [51]. Cross-platform Interoperability goal of Web Services also begins to fall when services start to use features beyond WSDL.

- SOA Location Transparency feature and deployment of Web Services to multiple locations help the load balancing i.e. Caching and Replication strategy improves the performance of a service by improving throughput, response time [2] and layering of resources. XML encryption at message level [25] also increases the security performance aspect while at the same time the utilization of loose coupling, integration of heterogeneous technologies, XML parser and mapping between data model used by Service Consumer and Service Provider [53], Two-level naming system, idempotent operations, use of SOAP envelop [54], augmentation of Flexibility, Reliability, Integrity, Security, and Reusability minifies Performance attribute [52].

- At network infrastructure level, use of Secure Sockets Layer (SSL), digital signatures, encrypted data transmission, and authentication of communicating parties [19] enhances security but this deals with point-to-point security only [55]. Increased security level makes the system complex and put a negative impact on Performance, Modifiability and Interoperability of Web Services [40]. Single sign-on may affect another attribute due to sharing of session information and also encryption can increases message size [19]. Here, message layer security provides end-to-end security but it has the organizational challenge like maintaining, signing certificates distribution etc. [55].

- Reliability attribute is the one which lifts up many other Quality Attributes e.g. Availability, Flexibility, Maintainability, Testability, Robustness, Usability [52] but the issues of management of transactional context to preserve data during failure and concurrent access decrease message reliability [19].

- Using more than one server is beneficial for Flexibility, Performance, Reliability, and introduces Efficiencies [56] which make possible to accommodate more users [57], reduce denial of service, and workload reduction with caching and load balancing [58] but inherently Web Service do not offer any scalability feature, therefore many performance issues are induced like complexity of both architecture and Web Services amplification (when multiple servers and multiple service intermediaries get involved) [56].

- Run-time discovery reduces Testability because it is impossible to predict which service is actually used by a system until the system start execution. Also the XML document error (WSDL) makes the testability hard.

- Legacy application utilization and Web Service Composition (either using Orchestration or Choreography strategy) increases the Reusability attribute of a service, which further lifts up Flexibility, Interoperability, Portability, Maintainability and Testability of the system. On the other hand service interface modification (legacy system) creates a problem when interfaces are being used by the Service Consumer because in such case it is difficult to identify who is using a service and what impact they will have? [19].

- SOA increases usability attribute because of the separation of implementation logic and interface logic [2] also statelessness feature advantages user satisfaction towards Web Service [49][31]. Pragmatically, there is no way to get user feedback, to control over communication, to get progress notifications and additional information to avoid same service call repetition or cancellation of an active call [19].

b) Hotspots

The hotspots in any research field lead towards improvement. So to answer the question "What are the main Research Challenges with QoS Attributes for Web Services and SOA?", this part briefly present research gaps of QoS for SOA-based Web Service. Management of Web Services QoS is Critical in SOA because of distributed nature of this architecture. Many QoS Attributes are addressed but still, some of them are not under the umbrella of measurement. Ahead of this issue, some other highlighted areas are Semantic Web-based Service discovery, selection, and composition with QoS Attributes for effective and efficient performance. Service monitoring is also an important research area for ensuring the exact level of delivered quality. O'Brien et al. [19]discusses open issues related with nine Quality Attributes of SOA i.e. semantic Interoperability, performance model for highly complex run-time environment, need of efficient security mechanism, distributed transactional model, service monitoring, escalation and compensation framework for service-based system, processes and techniques to deal with identification of service up gradation and incorporation of new versions of service, automated testing for Web Services along with management of Service Level Agreement (SLA) that guarantees the required level of service for an entire system.

Distributed UDDI should focus on unveiling the unexplored alternatives to recognize the stated promise of SOA and flexible discovery of Web Services [51]. According to Olston et al. [58] "The right metrics for QoS are fuzzy. Future research on QoS should address this issue." [59]. Semantic modeling of QoS category and need of matching algorithm between desired and supplied QoS are considered as future work by [12]. As the number of QoS Attributes varies in literature so validation and customization of the quality model are required for the Service-based system. In [60], the monitoring of Web Services and need for an adaptive metrics has been emphasized while the balanced security mechanism is focused by[2][30][56].

The study of a data-center, virtualization, and cloud computing, industrial communication has stimulated a fresh direction for research awareness in Quality of Experience (QoE) and Quality of Protection (QoP). Development of QoS-aware middleware, QoS negotiation protocols, QoS monitoring, mapping global to local SLAs, automatic QoS controlling algorithms, the self-managing system for QoS (at traffic intensity and service demand at various resource system), workload forecasting are some issues addressed by [22]. Transactional Web Services management with QoS is found as research challenge for Web Services developers by Oracle [34]. The authors of this paper experience the need for standardized non-functional metrics so by considering these gaps and trends QoS can lead to its potential realization.

III. Conclusion

In this study, a systematic review on Quality-of-Service (QoS) for Service-oriented Architecture (SOA)-based Web Services has been presented by exploring the six significant research questions. The QoS is about satisfying the requirements of Service Consumers and fetching business values to the Service Providers with an agreed upon contract by describing the quality-level of services. This review work presents the analysis of existing definitions, metrics, standards, and research hotspots of QoS for SOA-based Web Services. The observed facts of this review paper are: 1) lack of the standard definition of QoS for Web Services, 2) the extended roles i.e. role of Web Service developer and network and hardware capability should be considered as an integral part in Web Service model from Quality Attributes view point, 3) some QoS attributes have many metrics for quantification e.g. for coverage of Reliability attribute more than nine authors have proposed different metrics while some important attributes such as Testability, Robustness and Accountability have not explored from metric development perspective, 4) lack of Quality metric standardization, and 5) efficient approaches for the management of trade-offs between attributes are highly required. In order to enhance this research effort, more research questions can be added to collect in-depth knowledge of QoS for SOA-based Web Services.

References

[1] Lewis, G. (2010). Getting Started with Service-Oriented Architecture (SOA) Terminology. Retrieved from http://www.w3c.or.kr/kr-office/TR/2003/NOTE-ws-qos-20031125/.

[2] Newcomer, E., & Lomow, G. (2004). Understanding SOA with web services (independent technology guides). Addison-Wesley Professional.

[3] Haas, H., & Brown, A. (2004, Feburary). Web services glossary. Retrieved from https://www.w3.org/TR/ws-arch/.

[4] International Tele Communication Union (2008). E.800: Definitions of terms related with quality of service. Retrieved from http://www.itu.int/rec/T-REC-E.800-200809-I.

[5] Microsoft (2009). Quality Attributes: Microsoft Application Architecture Guide. Retrieved from https://msdn.microsoft.com/en-us/library/ee658094.aspx.

[6] Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic Literature Reviews in Software Engineering–A Systematic Literature Review. Information and software technology, 51(1), 7-15.

[7] Cruz, R. L. (1995). Quality of Service guarantees in Virtual Circuit Switched Networks. IEEE Journal on Selected areas in Communications, 13(6), 1048-1056.

[8] Le Tien, D., Villin, O., & Bac, C. (2000). CORBA application tailored manager for quality of service support. In Proceedings of the Object-Oriented Real-Time Distributed Computing. Newport, CA:IEEE.

[9] Oriol, M. (2015). Monitoring the Quality of Service to support the Service Based System lifecycle. (Doctoral dissertation). Retrieved from http://upcommons.upc.edu/handle/2117/95669. (Accession Number B 13211-2015)

[10] Angelov, S., & Grefen, P. (2004). The business case for B2B e-contracting. In Proceedings of the Sixth International Conference on Electronic Commerce. Delft, The Netherlands:ACM.

[11] Allenotor, D., & Thulasiram, R. K. (2008). A fuzzy grid-QoS framework for obtaining higher grid resources availability. In Proceedings of the International Conference on Grid and Pervasive Computing. Kunming, China: Springer Berlin Heidelberg.

[12] Ran, S. (2003). A model for web services discovery with QoS. ACM Sigecom exchanges, 4(1), 1-10.

[13] Kaewbanjong, K., & Intakosum, S. (2015). QoS Attributes of Web Services: A Systematic Review and Classification. Journal of Advanced Management Science, 3(3). 194-202.

[14] Mani, A., & Nagarajan, A. (2002). Understanding quality of service for Web services. Retrieved from http://www.ibm.com/developerworks/library/ws-quality/ws-quality-pdf.pdf

[15] Chung, L., & do Prado Leite, J. C. S. (2009). On non-functional requirements in software engineering. In Borgida, A. T., Chaudhri, V. K., Giorgini, P., & Yu, E. S. (Ed.), Conceptual modeling: Foundations and Applications (pp. 363-379). Springer Berlin Heidelberg.

[16] Rajendran, T., & Balasubramanie, P. (2009). Analysis on the study of QoS-aware Web Services discovery. Journal of Computing, 1(1), 119-130.

[17] Makhlughian, M., Hashemi, S. M., Rastegari, Y., & Pejman, E. (2012). Web service Selection based on Ranking of QoS using Associative Classification. International Journal on Web Service Computing, 3(1), 1-14.

[18] Rathore, M., & Suman, U. (2011). Quality of Service Broker Based Process Model for Dynamic Web Service Composition. Journal of Computer Science. 7(8), 1267-1274.

[19] O'Brien, L., Merson, P., & Bass, L. (2007). Quality attributes for service-oriented architectures. In Proceedings of the International Workshop on Systems Development in SOA Environments. Minneapolis, MN:IEEE Computer Society.

[20] Priya, N. H., Priya, A. S., & Chandramathi, S. (2014). QoS based Selection and Composition of Web Services-A Fuzzy Approach. Journal of Computer Science, 10(5), 861-868.

[21] Menascé, D. A. (2002). QoS issues in Web Services. IEEE internet computing, 6(6), 72-75.

[22] Menasce, D. A. (2004). QoS Challenges and Directions for large Distributed Systems [PDF Document]. Retrieved from http://cs.gmu.edu/~menasce/papers/menasce-rome-2004.pdf.

[23] Oriol, M., Marco, J., & Franch, X. (2014). Quality Models for Web Services: A Systematic Mapping. Information and Software Technology, 56(10), 1167-1182.

[24] W3C Working Group. (2003). QoS for Web Services: Requirements and Possible Approaches. Retrieved from http://www. w3c. or. kr/kr-offce/TR/2003/ws-qos.

[25] Kim, E. & Lee, Y. (2005). Quality Model for Web Service. Retrieved from http://cliplab.org/Projects/S-CUBE/papers/oasis05:WSQM-2.0.pdf.

[26] Sheng, Q. Z., Qiao, X., Vasilakos, A. V., Szabo, C., Bourne, S., & Xu, X. (2014). Web Services Composition: A Decade's Overview. Information Sciences, 280, 218-238.

[27] Perepletchikov, M. (2009). Software Design Metrics for Predicting Maintainability of Service-Oriented Software (Unpublished Doctoral dissertation), RMIT University Melbourne, Australia.

[28] Maximilien, E. M., & Singh, M. P. (2004). A Framework and Ontology for Dynamic Web Services Selection. IEEE Internet Computing, 8(5), 84-93.

[29] Tran, V. X. (2008). WS-QoS Onto: A QoS Ontology for Web Services. In Proceeding of the IEEE International Symposium on Service-Oriented System Engineering. Jhongli:IEEE.

[30] Erl, T. (2007). SOA: Principles of Service Design. Prentice Hall Press.

[31] Nikov, A., Zaim, S., & Oztekin, A. (2006). Usability evaluation of web services by structural equation modeling. In Proceedings of the Conference of Mathematics and Its Application. St. Augustine, Trinidad:Publisher.

[32] Upadhyaya, B., Zou, Y., Keivanloo, I., & Ng, J. (2015). Quality of Experience: User's Perception about Web Services. IEEE Transactions on Services Computing, 8(3), 410-421.

[33] Oberortner, E., Sobernig, S., Zdun, U., & Dustdar, S. (2012). Monitoring Performance-related QoS properties in Service-Oriented Systems: A Pattern-based Architectural Decision Model. In Proceedings of the Sixteenth European Conference on Pattern Languages of Programs. New York, NY: ACM.

[34] Durand, J. (2011). Interoperability Guidelines. Retrieved from https://docs.oracle.com/cd/E23943_01/web.1111/e15184/interop.htm#WSCPT168.

[35] Ladan, M. I. (2012). Web Services Metrics: A Survey and A Classification. Journal of Communication and Computer, 9(7), 824-829.

[36] Jiang, S., & Aagesen, F. A. (2006). An Approach to Integrated Semantic Service Discovery. In Proceedings of the Autonomic Networking. Paris, France: Springer Berlin Heidelberg.

[37] Zhou, W., Wen, J., Gao, M., & Liu, J. (2013). A QoS preference-based algorithm for Service Composition in Service-Oriented Network. Optik-International Journal for Light and Electron Optics, 124(20), 4439-4444.

[38] Recio, R. J. (2003). Server I/O networks past, present, and future. In Proceedings of the ACM SIGCOMM workshop on Network-I/O convergence: experience, lessons, implications. NewYork, USA: ACM.

[39] Egger, S., Reichl, P., Hoßfeld, T., & Schatz, R. (2012). "Time is bandwidth"? Narrowing the gap between subjective time perception and Quality of Experience. In Proceedings of the International Conference on Communications.Ottawa, Canada: IEEE.

[40] Kaur, P. & Singh, H. (2015). Web Services Quality Metrics: A Review. Paper presented at Second International Conference on Emerging trends in Computer Science, Communication and Information Technology, Nanded, Punjab.

[41] Hirzalla, M., Cleland-Huang, J., & Arsanjani, A. (2008). A Metrics Suite for Evaluating Flexibility and Complexity in Service Oriented Architectures. In Proceedings of the International Conference on Service-Oriented Computing. Sydney, Australia: Springer Berlin Heidelberg.

[42] Suresh Babu, G. N. K., & Srivatsa, S.K. (2009). Analysis and Measures Of Software Reusability. International Journal of Reviews in Computing, 1, 41-46.

[43] Nadanam, P., & Rajmohan, R. (2012). QoS evaluation for web services in cloud computing. In Proceedings of the third International Conference Computing Communication & Networking Technologies. Coimbatore, India: IEEE.

[44] Pressman, R. S. (2005). Software Engineering: A Practitioner's Approach. Palgrave Macmillan.

[45] Gunther, N. J. (2000). The practical performance analyst. Authors Choice Press.

[46] Jian, H., Xincai, W., Zhenming, H., & Bian, W. (2010, April). Research on a data exchange center of land resources based on SOA. In Information Management and Engineering (ICIME), 2010 The 2nd IEEE International Conference on (pp. 644-647). IEEE.

[47] Vathsala, A. V., & Mohanty, H. (2014). A survey on checkpointing web services. In Proceedings of the Sixth International Workshop on Principles of Engineering Service-Oriented and Cloud Systems. New York, NY: ACM.

[48] Choi, S. W., Her, J. S., & Kim, S. D. (2007). Modelling QoS attributes and metrics for evaluating services in SOA considering consumers' perspective as the first class requirement. In Proceedings of the Second IEEE Asia-Pacific Service Computing Conference. Tsukuba Science City, Japan: IEEE.

[49] Marzolla, M., & Mirandola, R. (2010). QoS Analysis for Web Service Applications: A Survey of Performance-Oriented Approaches from an Architectural viewpoint. Retrieved from http://www.informatica.unibo.it/it/ricerca/technical-report/2010/UBLCS-2010-05.

[50] Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J., & Sheng, Q. Z. (2003). Quality driven web services composition. In Proceedings of the Twelfth International Conference on World Wide Web. USA: ACM.

[51] Mcheick, H., & Qi, Y. (2012). Quality attributes and design decisions in Service-Oriented Computing. In Proceeding of the International Conference on Innovations in Information Technology. AbuDhabi, UAE: IEEE.

[52] Bashir, R. Azam, F. Iqbal, M. A. Khanum, A., & Malik, H. (2012). A Comparative Model for Trade-off Analysis of Attributes in SOA. Journal of Basic and Applied Scientific Research, 2(11), 11097-11105.

[53] Swientek, M., Bleimann, U., & Dowland, P. (2008). Service-Oriented Architecture: Performance Issues and Approaches. In Proceedings of the Seventh International Network Conference. Plymouth, UK: Publisher.

[54] Erenkrantz, J. R. (2004). Web Services: SOAP, UDDI, and Semantic Web. Retrieved from http://www.erenkrantz.com/Geeks/Research/ics-221/ICS221-Term-Paper.pdf

[55] Josuttis, N. M. (2007). SOA in Practice: The Art of Distributed System Design. O'Reilly Media.

[56] Nottingham, M. (2001, April). Web Service Scalability and Performance with Optimizing Intermediaries [PDF document]. Retrieved from http://www.w3.org/2001/04/wsws-proceedings/mnot/wsws-nottingham.pdf.

[57] Bravetti, M., Gilmore, S., Guidi, C., & Tribastone, M. (2007). Replicating web services for scalability. In Proceedings of the International Symposium on Trustworthy Global Computing. Sophia-Antipolis, France: Springer Berlin Heidelberg.

[58] Olston, C., Manjhi, A., Garrod, C., Ailamaki, A., Maggs, B. M., & Mowry, T. C. (2005). A Scalability Service for Dynamic Web Applications. Presented at Second Biennial Conference on Innovative Data Systems Research. Asilomar, CA.

[59] Juvva, K. (1998). Quality of Service. Retrieved from https://users.ece.cmu.edu/~koopman/des_s99/quality_of_service/.

[60] Hasan, M. H., Jaafar, J., & Hassan, M. F. (2014). Monitoring Web Services' Quality of Service: A Literature Review. Artificial Intelligence Review, 42(4), 835-850

Authors Profile

Aradhana Negi is a Ph.D. research scholar in the Department of Computer Science, Guru Nanak Dev University, India. She received her Master of Computer Applications from Guru Nanak Dev Engineering College, India. Her research interest includes Semantic Web, Automatic Semantic Web Service Discovery and Composition, and Cognitive Psychology in computing.

Parminder kaur is working as Assistant Professor in the Department of Computer Science, Guru Nanak Dev University, India. She has received her Ph.D. degree in 2011 from Department of Computer Science, Guru Nanak Dev University, India. Her research interests are Component-Based Software Engineering, Semantic Web Services, Version Controlling mechanism in Open Source Development System. She has published approximately 46 research papers in various international/national journals and conferences.