Kuwait J. Sci. **42 (3)** pp. 134-157, 2015

Modelling and measuring the quality of online services

ZAINAB MOHAMMD ALJAZZAF

Kuwait University, Department of Information Science,
College of Computing Sciences and Engineering, Kuwait
Email: eng.zainab4az@gmail.com

ABSTRACT

A distributed application may be composed of global services provided by different organisations and have different properties. Providing provisioned services is of primary importance due to the multi-tenant and potentially multi-provider nature of service computing paradigms. Quality of Services (QoS) has been used as a distinguishing factor between similar services and as a criterion for service selection. To date, the majority of research on QoS is not comprehensive in identifying QoS suitable for online open environments with diverse services.In addition, a lot of studies that is dedicated to build QoS models are omitting the evaluation methods for many QoS. Therefore, there is a need for a QoS model that is comprehensive. Accordingly, the contribution in this paper compromises a generic QoS model that identifies a comprehensive QoS metrics and their evaluation approaches. As a consequence, this study should be a useful reference to academic and industry researchers in the web-based service and online service discovery, selection, composition, and management. Experiments are conducted to test various QoS metrics from the proposed QoS model.

Keywords: Quality of services; quality of service model; service-oriented computing; services.

INTRODUCTION

The development of a distributed software system requires the interaction of services and the use of resources from diverse organisations throughout the Web. A service is "a discrete unit of business functionality that is made available through a service contract" (Rosen *et al.*, 2008), which includes a service interface, service documents, service policies, Quality of Service(QoS), and performance.

The international quality standard ISO 8402 (part of the ISO 9000) describes quality as "the totality of features and characteristics of a productor service that bear on its ability to satisfy stated or implied needs". Hoyle (2005) defines quality as "the degree to which a set of inherent characteristics fulfils a need or expectation that is stated, generally implied orobligatory". The World Wide Web Consortium (W3C) describes the QoS requirements for Web Services asthe quality aspects of Web Services (Lee *et al.*, 2003).

Considering QoS is important for the developing of distributed computing paradigms that make use of services, such as Service-Oriented Computing (SOC) and cloud computing. SOC is “a computing paradigm that utilises services as fundamental elements to support rapid, low-cost development of distributed application in heterogeneous environments” (Papazoglou & Georgakopoulos, 2008). To realise the potential of SOC, Service-Oriented Architecture (SOA) is developed to overcome many enterprise challenges, including designing complex distributed services, managing business processes, ensuring transaction QoS, complying with agreements, and leveraging different computing devices such as personal computers and cell phones (Papazoglou & Georgakopoulos, 2008). SOA is “an architectural style forbuilding enterprise solutions based on services” (Rosen *et al.*, 2008).

There are three role interactions in SOA, as shown in Figure 1 (Papazoglou, 2012):1) The service provider is an organisation or platform that owns, implements, and controls access to the service; 2) the service requestor is an application, services, or the client who is looking for and invoking a service; and 3) the service registry or service broker is a searchable directory where the description of the services is published by service providers and searched by requestors.

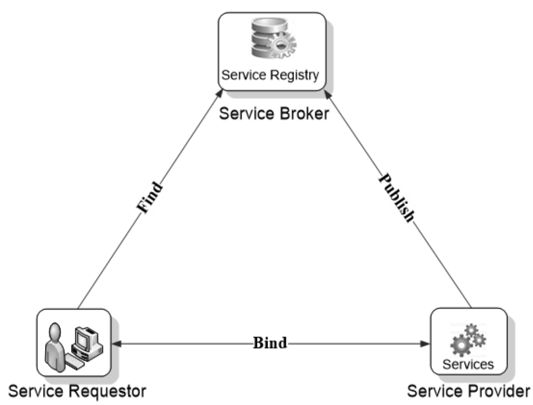

Fig. 1. Services roles and operations in SOA (Papazoglou, 2012).

Whereas, cloud computing is “a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction”(Mell & Grance, 2011). Hence, the word cloud describes the Web as a space where computing has been preinstalledand exists as a service (Sadiku *et al.*, 2014).

Figure 2 shows the cloud computing architecture (Sadiku *et al.*, 2014). There are many services provided by different service providers. Services provided by the cloud can be divided into three categories: 1) Software as a Service (SaaS), where the consumer uses the provider's applications running on a cloud infrastructure; 2) Platform as a Service (PaaS), where the consumer deploys, onto the cloud infrastructure, his created or acquired applications created using programming languages, libraries, services, and tools supported by the provider; and 3) Infrastructure as a Service (IaaS), where the consumer canprovision processing, storage, networks, and other fundamental computing resources (Mell & Grance, 2011).

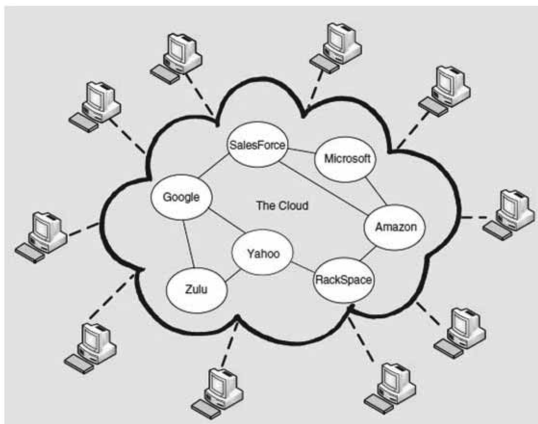

Fig.2. Cloud computing (Sadiku *et al.*, 2014).

To build a distributed application from services, the application developer, or service requestor, may need to select services from different service providers. Because there are many services with similar *functionalities*, service requestors need to be able to differentiate between them. The only differentiating factor between similar services may be their*non-functional* properties, which can be considered as criteria for service selection. QoS has been used as a non-functional property for selecting services (Papazoglou *et al.*, 2006; Maximilien & Singh, 2004; Dragoni, 2009; Ying Feng & Pei-Ji, 2006; Huhns & Singh, 2005; Zhang *et al.*, 2012; Kalepu *et al.*, 2003; Kim & Doh, 2007; Liu *et al.*, 2004; Zheng *et al.*, 2014).

For example, one approach in service selection (Ran, 2003) is where the service registry captures the QoS provided by the service provider andthe QoS required by the service requestor, and accordingly match between the two, when discovering the service to select the best match among services with similar functionality. In this scenario, a service requestor may need a service that is reliable, which is considered as a QoS property of the service. Among many similar services with different reliabilities, a service with a highest reliability will be selected by the requestor.

Many studies have been conducted to examine QoS compliance by monitoring services or by collecting quality ratings from the users (Papazoglou *et al.*, 2006; Kim

& Doh, 2007; Zhang *et al.*, 2012; Liu *et al.*, 2004; Huhns & Singh, 2005; Kalepu *et al.*, 2003; Zheng *et al.*, 2014).Thus, considering the issues related to the online services and distributing paradigms, QoS is an important contributing factor to the evolution of distributed paradigms, such as service-oriented and cloud computing paradigms.

Accordingly, identifying the QoS is crucial. To date, the majority of research on QoS is not comprehensive in identifying QoS. For example, researchers refer to other research works to extract multiple QoS, apply certain QoS appropriate for a certain domain, such as e-commerce, or categorise and define certain QoS from their study perspective with some overlap between the QoS used. Moreover, a lot of studies that is dedicated to build QoS models are omitting the evaluation methods for many QoS.

Therefore, there is a need for a generic QoS model that can be considered as a reference for the researchers. Accordingly, the contribution in this paper compromises a generic QoS model that identifies a comprehensive QoS metrics and their evaluation approaches. As a consequence, this study should be useful to academic and industry researchers in the web-based service and online service discovery, selection, composition, and management. The rest of the paper presents the related work, proposed QoS model, QoS evaluation approaches, experiment, and finally the conclusion.

RELATED WORK

There are different research efforts to define and categorise QoS as well as attempts to express, quantify, and model QoS (Lee *et al.*, 2003; Kim & Doh, 2007; Ran, 2003; Yu *et al.*, 2007; Rosenberg *et al.*, 2006; Zeithaml *et al.*, 2002). In (Lee *et al.*, 2003; Kalepu *et al.*, 2003; Wang & Vassileva, 2007; Ran, 2003; O'Brien *et al.*, 2007; Moorsel, 2001; Garcia & de Toledo, 2006; Rahman & Meziane, 2008; Menasce, 2002; Hoyle, 2005), the research efforts include generic and business QoS requirements for services. However, the researchers use an identified QoS from others' models, apply some QoS that are appropriate for a domain, orcategorise and define QoS from their study perspective. Even less research has been conducted to categorise and model QoS or to identify their evaluation techniques. To the best of our knowledge, there is no great consensus about a set of generic QoS vital to online services.

Some researchers identify the QoS but do not identify the evaluation techniques. O'Brien *et al.* (2007) define QoS requirements for SOA, which are modifiability, testability, and usability. In addition, the authors identify the required QoS for SOA and argue that QoS can be significantly affected by SOA. Ran (2003) identifies QoS and organises them into categories. The categories are grouped into different types, i.e., QoS related to runtime, transaction support, configuration management and cost, and security. Rahman & Meziane (2008) present five essential QoS requirements based on the most common QoS requirements in the literature, which are: readiness, transaction, reliability, speed, andsecurity. However, those models are not generic and cannot include other QoS. In addition, the researchers did not identify the QoS evaluation techniques.

Apart from some common QoS, Hoyle (2005) identifies other quality characteristics for services, such as courtesy, comfort, competence, honesty, and responsiveness. Larson (1998) identifies service ability and user satisfaction as performance measurements for service delivery, and Moorsel (2001) discusses quantitative metrics and develops a framework for evaluating Internet services. This author defines three metrics that should be used to evaluate Business to Consumer (B2C), Business to Business (B2B), and service providers. The metrics include QoS, Quality of Experience (QoE), and Quality of Business (QoBiz). While QoE quantifies the user experience, QoBiz measures the business return.

In QoE literature (Lalanne *et al.*, 2012; Bouch *et al.*, 2000; Fiedler *et al.*, 2010; Varela *et al.*, 2014), the QoS are considered as parameters for QoE evaluation. However, the researchers identified some QoS for evaluating QoE that are related to their field of study and environment, such as network environment, mobile, multimedia (ex. radio, video), smart phone, and web browsing (Yaacoub & Dawy, 2014; Kim & Choi, 2010; Kim *et al.*, 2010; Wei *et al.*, 2012). Some of the identified QoE parameters are: Throughput, Delay, Jitter (Hsu *et al.*, 2013), Mean-squared error distortion (Yaacoub & Dawy, 2014.), and delay, jitter, loss rate, error rate, bandwidth, and signal success rate (Kim *et al.*, 2008).

Moreover, Cabrera & Franch (2012) build a quality model for Web Services based on the ISO/IEC 9126 quality standard. The authors identify different technical and non-technical quality characteristics, such as efficiency, compliance, and changeability. Menasce (2002) examines QoS issues in Web Services. The author mentioned some QoS, which are: availability, response time, security, and throughput.

Yu *et al.* (2007) provide a list of QoS parameters and explain how to evaluate each. However, the authors worked on some major QoS for Web services as proposed by the W3C and did not build a QoS model. Zheng *et al.* (2014) mainly focused on investigating user-observed QoS properties, which are failure probability, response time, and throughput only. On the other hand, privacy is not identified as a QoS in the literature (Ran, 2003; Lee *et al.*, 2003; Kim & Doh, 2007; Yu *et al.*, 2007; Rosenberg *et al.*, 2006). However, Zeithaml *et al.* (2002) identified privacy as a quality metric for Web sites.

None of these works is complete in identifying a generic QoS, building a QoS model, and describing QoS evaluation approaches, which is essential as QoS is an important factor in evolving service computing paradigms. Therefore, there are many aspects of QoS and there is a need to identify the QoS and build a model that categorises the QoS. In addition, there is a need to represent each QoS and identify approaches forquantifying and evaluating them.

THE PROPOSED QOS MODEL

A QoS model is used for expressing QoS attributes for services. There is a need to build a generic QoS model suitable for online open environments with diverse services. There are many aspects of QoS important to online services. To identify a generic QoS, the QoS of services are aggregated from the literature (Lee *et al.*, 2003; Rahman & Meziane, 2008; Ran, 2003; Yu *et al.*, 2007; Kim & Doh, 2007; Garcia & de Toledo, 2006; O'Brien *et al.*, 2007; Papazoglou, 2012; Entrust, 2001; Hoyle, 2005; Menasce, 2002; Kim & Doh, 2007; Dingledine *et al.*, 2000, Swaid & Wigand, 2009; Zeithaml *et al.*, 2002; Li *et al.*, 2002) and a QoS model is built.

Subsequently, the aggregated QoS needs to be classified to build a QoS model. Since the QoS need to be quantified to be measured and different QoS metrics have different approaches to measure, the proposed QoS model is categorised based on the QoS evaluation approach. Accordingly, the QoS model divides QoS into two categories: objective QoS and subjective QoS, each with a different evaluation method, as presented in Table 1.

Table 1. The QoS model

QoS	**References**
Objective QoS	
Latency	Rosenberg *et al.*, 2006; Lee *et al.*, 2003; Ran 2003.
Execution Time	Lee *et al.*, 2003; Rosenberg *et al.,* 2006.
Response Time	Lee *et al.*, 2003; Rahman & Meziane, 2008; Garcia & de Toledo, 2006; O'Brien *et al.*, 2007; Menasce, 2002; Ran, 2003; Yu *et al.*, 2007; Kim & Doh, 2007.
Throughput	Lee *et al.*, 2003; Garcia & de Toledo, 2006; O'Brien *et al.*, 2007; Menasce, 2002; Ran, 2003.
Transaction Time	Lee *et al.*, 2003.
Availability	Lee *et al.*, 2003; Rahman & Meziane, 2008; O'Brien *et al.*, 2007; Menasce, 2002; Ran, 2003; Yu *et al.*, 2007; Kim & Doh, 2007; Zeithaml *et al.,* 2002.
Reliability	Lee *et al.*, 2003; Rahman & Meziane, 2008; Ran, 2003; Yu *et al.*, 2007; Kim & Doh, 2007; Garcia & de Toledo, 2006; Swaid & Wigand, 2009; Zeithaml *et al.,* 2002; Li *et al.,* 2002; Zeithaml *et al.,* 2000.

Scalability	Lee *et al.,* 2003; Garcia & de Toledo, 2006; Ran, 2003; Yu *et al.*, 2007.
Integrity	Lee *et al.,* 2003; Papazoglou, 2012; Entrust, 2001; Garcia & de Toledo, 2006; Ran, 2003; Yu *et al.*, 2007.
Capacity	Lee *et al.*, 2003; Rahman & Meziane, 2008; Garcia & de Toledo, 2006; Ran, 2003; Menasce & Almeida, 2001.
Robustness	Lee *et al.,* 2003; Garcia & de Toledo, 2006; Ran, 2003; Yu *et al.*, 2007.
Accuracy	Lee *et al.,* 2003; Garcia & de Toledo, 2006; Ran, 2003; Yu *et al.*, 2007; Hoyle, 2005; Zeithaml *et al.,*2002.
Accessibility	Lee *et al.*, 2003; Rahman & Meziane, 2008; Garcia & de Toledo, 2006; Yu *et al.*, 2007; Hoyle, 2005; Mathijssen, 2005; Zeithaml *et al.,* 2000;Zeithaml *et al.,* 2002.
Timeliness	O'Brien *et al.,* 2007.
Execution Price	Rahman & Meziane, 2008; Garcia & de Toledo, 2006; Ran, 2003; Kim &Doh, 2007.
Domain Specific and other Objective QoS	
Subjective QoS	
Security	Lee *et al.*, 2003; Ran, 2003; Hoyle, 2005; Dingledine *et al.*, 2000; Papazoglou, 2012; Garcia & De Toledo, 2006; O'Brien *et al.,* 2007; Rahman & Meziane, 2008; Menaasce, 2002; Yu *et al.,* 2007; IBM & Microsoft, 2002; Zeithaml *et al.,* 2002; Zeithaml *et al.,* 2000.
Transaction ACID	Rahman & Meziane, 2008.
Regulatory	Rahman & Meziane, 2008; Ran, 2003.
Exception Handling	Lee *et al.,*2003; Ran, 2003.
Interoperability	Lee *et al.*, 2003; Garcia & de Toledo, 2006; O'Brien *et al.*, 2007; Yu *et al.*, 2007.
Competence	Grandison & Sloman, 2000; Grandison & Sloman, 2000; Hoyle, 2005; Song *et al.*, 2007; Aljazzaf *et al.,* 2011.
Honesty	Hoyle, 2005; Malik & Bouguettaya, 2009; Jurca & Faltings, 2003; Miller *et al.*, 2002; Aljazzaf*et al.,* 2011; Malik & Bouguettaya, 2009
Usability	O'Brien *et al.,* 2007; Swaid & Wigand, 2009; Zeithaml *et al.,* 2002.
Testability	O'Brien *et al.,* 2007.
Stability	Ran, 2003.
Supported Standards	Ran, 2003.
Modifiability	O'Brien *et al.,* 2007.
Completeness	Ran, 2003.
Efficiency	Hoyle, 2005.

Changeability	Cabrera & Franch, 2012.
Compliance	Cabrera & Franch, 2012.
Comfort	Hoyle, 2005.
Credibility	Hoyle, 2005.
Dependability	Hoyle, 2005.
Effectiveness	Hoyle, 2005; Zeithaml *et al.,* 2000.
Flexibility	Hoyle, 2005, Zeithaml *et al.,*2000.
Promptness	Hoyle, 2005.
User Satisfaction	Larson, 1998.
Responsiveness	Hoyle, 2005; Swaid & Wigand, 2009; Zeithaml *et al.,* 2002; Li *et al.,* 2002; Zeithaml *et al.,* 2000.
Serviceability	Larson, 1998.
Domain Specific and Other Subjective QoS	

Objective QoS can be measured and contains formulas for measurement, while subjective QoS are difficult to measure and require other quantification approaches or techniques to evaluate them and quantify theiruse in the QoS evaluation. For example, although security is not a quantifiable element, there are different degrees of security that a system can provide based on the system's support level of security technologies. The following section discusses the objective and subjective QoS properties and their evaluation approaches.

THE QOS EVALUATION APPROACHES

This section presents the QoS and their evaluation approaches. The following two subsections discuss the objective QoS and subjective QoS. Different QoS are defined, followed by their evaluation techniques. However, extended research is required to study and evaluate the other QoS.

Objective QoS

This section presents the objective QoS and their evaluation approaches.The objective QoS are defined and represented as a set of metrics to quantify them for evaluation purposes, as follows:

- Latency ($QoS_{Latency}$):

The latency or network latency time of a service is the time the message needs to reach its destination (Rosenberg *et al.*, 2006). Lee *et al.* (2003) defines latency as "the Round-Trip Delay (RTD) between sending a request and receiving a response". Ran (2003) defineslatency as "time taken between the service request arrives and the request is being serviced". Normally there is a variation in the measured network latency (Jitter), which is caused by network congestion and route changes. Latency ($QoS_{Latency}$) can be measured as the time between sending the request and receiving the response excluding the time that the server spend processing the request.

- Execution Time ($QoS_{Execution}$):

The execution time of a service is the time taken by the service to execute and process its sequence of activities (Lee *et al.*, 2003). Rosenberg *et al.* (2006) represents execution time as the time that the provider needs to finish processing the request. Execution time includes the processing time, the time needed to carry out the operation for a request; and wrapping time, the time needed to wrap and unwrap the Extensible Markup Language (XML) structure of the request. Execution Time ($QoS_{Execution}$) can be measured as the time during which a service is executing its operation.

- Response Time ($QoS_{Response}$):

The response time of a service is the time required to process and complete a service request; the response time includes the execution time and the latency (Lee *et al.*, 2003; Rahman & Meziane, 2008; Garcia & de Toledo, 2006; O'Brien *et al.*, 2007; Menasce, 2002; Ran, 2003; Yu *et al.*, 2007; Kim & Doh, 2007). Response time is a measure of the total elapsed time between the start and completion of a task. The following is the formula to evaluate the response time:

$$QoS_{Response} = QoS_{Execution} + QoS_{Latency}$$

Figure 3 shows a graphical representation for the three time frames: latency time, execution time, and response time; and identifies the relationships between them. Moreover, the figure demonstrates the evaluation technique for each QoS. The transmission time and the message wrapping time are included in the response time.

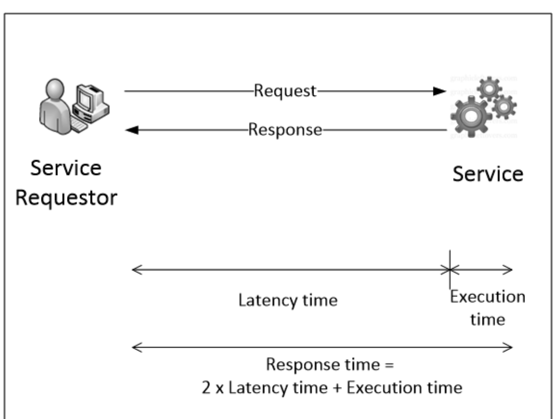

Fig. 3. Latency, execution, and response time frames.

- Throughput ($QoS_{Throughput}$):

The throughput of a service refers to the number of requests a service can process per unit of time (Lee *et al.*, 2003; Garcia & de Toledo, 2006; O'Brien *et al.*, 2007; Menasce, 2002; Ran, 2003). Throughput depends on the power of the service machines and it is measured by sending many requests over a period of time and counting the number of responses. Throughput is related to latency or capacity (Ran, 2003). The following is the formula to evaluate the throughput:

$$QoS_{Throughput} = \frac{Number \quad of \quad requests}{Time \quad period}$$

- Transaction Time ($QoS_{TransactionTime}$):

The service process may be divided into indivisible operations, called transactions. Each transaction must succeed or fail as a complete unit. Lee *et al.* (2003) defines transaction time as "the time that passes while the web service is completing one complete transaction". For example, transaction time may encompasses the time taken for the request made to the web server, there after being processed, sent to the application server, which may make a request to the database server, then repeated again backward from the database server, application server, web server and back to the user; including the transmission time.

Transaction time ($QoS_{TransactionTime}$) measure begins, when the defined transaction makes a request to the application, till the transaction completes before proceeding with the next subsequent request, and will stop when the transaction completes.The performance of a web service, which represents how fast a service request can be completed, can be measured in terms of throughput, response time, latency, execution time, and transaction time.

- Availability ($QoS_{Availability}$):

Services should be available for direct invocation. The availability of a service is the probability that a service is up, present, and accessible to use (Lee *et al.*, 2003; Rahman & Meziane, 2008; O'Brien *et al.*, 2007; Menasce, 2002; Ran, 2003; Yu *et al.*, 2007; Kim & Doh, 2007). Time-to-Repair (TTR) represents the time it takes to repair the service that has failed and is associated with availability. Availability is related to reliability (Lee *et al.*, 2003). The following is the formula to evaluate the availability:

$$QoS_{Availability} = \frac{Uptime}{Total\ time}$$

- Reliability ($QoS_{Reliability}$):

The reliability or success rate of a service means the ability of a serviceto perform its function under stated conditions correctly with either "nofail" or "response failure to the user" for a specific interval of time; itis related to availability (Lee *et al.*, 2003; Rahman & Meziane, 2008; Ran, 2003; Yu *et al.*, 2007; Kim & Doh, 2007). Other researchers define reliability in a similar way (Garcia & de Toledo, 2006; Swaid & Wigand, 2009). Important aspects of reliability are reliability of the message and the reliability of the services themselves. The SOA platform, not the service developer, is responsible for providing message reliability (assure that the message does not fail to get delivered) (O'Brien *et al.*, 2007). Reliability can be evaluated as follows (Yu *et al.*, 2007):

$$QoS_{Reliability} = 1 - \frac{n}{N*t}$$

where t denotes the total time a service is monitored for recording the number of failures, n is the number of failures encountered during that period, and N is the total number of events (number of successful events plus number of failures), deriving the the reliability or the success rate.

- Scalability ($QoS_{Scalability}$):

Lee *et al.* (2003) defines scalability as "the capability of increasing the computing capacity of a service provider's computer system and the system's ability to process more users' requests, operations or transactions in a given time interval". Other researchers define scalability in a similar way (Garcia & de Toledo, 2006; Ran, 2003; Yu *et al.*, 2007). Yu *et al.* (2007) define scalability as "how expandable a Web service can be". The authors identify the Performance Non-Scalability Likelihood (PNL) metric as a relatively newer technique to predict whether the system is going to be able to endure the higher loads of traffic without affecting the performance levels.

- Integrity ($QoS_{Integrity}$):

There are two types of integrity: data and transaction. Lee *et al.* (2003) defines data integrity as "whether the transferred data is modified in transit" and transactional integrity as "a procedure or set of procedures, which is guaranteed to preserve database integrity in a transaction". Integrity means that the data received is the same as the data sent and proof that at any time different copies of the message are the same (Papazoglou, 2012). Signature technique is used to ensure message integrity (i.e., the message does not change) (Entrust, 2001). Other researchers define integrity in a similar way (Garcia & de Toledo, 2006; Ran, 2003; Yu *et al.*, 2007). Yu *et al.* (2007) discussed that data integrity can be calculated as the ratio of successful transactions to the total number of transactions, as follows:

$$QoS_{Integrity} = \frac{Number\ of\ successful\ transactions}{Total\ number\ of\ transactions}$$

- Capacity ($QoS_{Capacity}$):

Capacity is the maximum number of simultaneous requests that the service can process with guaranteed performance (Lee *et al.*, 2003; Rahman & Meziane, 2008; Garcia & de Toledo, 2006; Ran, 2003). Menasce & Almeida (2001) provide a quantitative framework for planning the capacity of Web services and understanding their behavior.

- Robustness ($QoS_{Robustness}$):

Lee *et al.* (2003) defines robustness as "the degree to which a web service can function correctly even in the presence of invalid, incomplete or conflicting inputs". Other researchers define robustness in a similar way (Garcia & de Toledo, 2006; Ran, 2003; Yu *et al.*, 2007). Yu *et al.* (2007) include an approach for a robustness measurement and identify a robust system as a system that always has control overcritical situations and has the capability to recover from a failure ormay detect a failure.

The authors define Mean Time to Failure (MTTF) as the average time it takes for the system to fail once more and the Mean Time to Recover (MTTR) as the average amount of time it takes for the system to recover from a failure. Thus, Mean Time between Failures (MTBF) can be defined as the amount of time it takes for the failure to occur and then recover. An MTBF with a small value shows that the failures are very infrequent. Therefore, robustness can be measured in the followingway:

- Accuracy ($QoS_{Accuracy}$):

Lee *et al.*(2003) defines accuracy as "the error rate generated by the web service". Other researchers define accuracy in a similar way (Garcia & de Toledo, 2006; Ran, 2003; Yu *et al.*, 2007; Hoyle, 2005). Yu *et al.* (2007) measure the accuracy of the Web service as the standard deviation of the reliability. If the average value of the standard deviation is equal to zero, then the measurement is said to be accurate. However, the measurement is considered to be inaccurate if the average value of the standard deviation is very high.

- Accessibility ($QoS_{Accessibility}$):

Accessibility refers to the service capability to serve the client's requests (Lee *et al.*, 2003; Rahman & Meziane, 2008; Garcia & de Toledo, 2006; Yu *et al.*, 2007; Hoyle, 2005). Mathijssen (2005) denoted accessibility as follows:

$$QoS_{Accessibility} = \frac{The\ number\ of\ succeeded\ user's\ requests}{Total\ requests\ done\ by\ the\ user}$$

Similarly, Yu *et al.* (2007) calculate accessibility as "a ratio of the number of successful acknowledgements received to the total number of requests sent", as follows:

$$QoS_{Accessibility} = \frac{Number\ of\ acknowledgements\ received}{Total\ number\ of\ requests\ sent}$$

- Timeliness ($QoS_{Timeliness}$):

O'Brien *et al.* (2007) defines timeliness as "the ability of the web service to meet a deadline, i.e., to process a request in a deterministic and acceptable amount of time".

- Execution price ($QoS_{ExecutionPrice}$):

The execution price refers to the amount of money the request or should pay for operation execution (Rahman & Meziane, 2008; Garcia & de Toledo, 2006; Ran, 2003; Kim & Doh, 2007).

Subjective QoS

This section presents the subjective QoS and their evaluation approach.The subjective QoS are defined and represented as a set of metrics; metrics are used to quantify and thus to evaluate the QoS. However, the subjective QoS require extensive evaluation approaches.

- Security ($QoS_{Security}$):

Security should be considered when providing services (Lee *et al.*, 2003; Ran, 2003; Hoyle, 2005). Secure services provide the following aspects:

1. Authentication: Users should be authenticated to access the services and data (Lee *et al.*, 2003). Authentication in the online world is important to verify the parties involved (Dingledine *et al.*, 2000). Some authentication methods are user name and password, digital certificate, or tokens (Papazoglou, 2012).
2. Authorisation: This allows access to resources and limits the action performed in the system. An access control rule is used to restrict access to resources. Authentication needs to be established before authorisation.
3. Confidentiality: This means only authorised parties can view the message. Encryption is used to provide confidentiality of the message, i.e., no one other than the sender and the receiver can reador modify the message (Entrust, 2001).
4. Non-repudiation: To prove that the actions have taken place.
5. Accountability: The supplier can be held accountable for their services.
6. Traceability and Auditability: The possibility to trace the history of a service when a request was serviced.
7. Data encryption. A service should provide encryption techniques to secure the data.

Similarly, Garcia & De Toledo (2006) list authentication, confidentiality, and integrity as security requirements. O'Brien *et al.* (2007) describe the following security requirements for SOA: confidentiality and encryption, authentication, authorisation, and ensuring the published data in the directory is up to date and was added by a valid publisher. Rahman & Meziane (2008) and Menaasce (2002) added rigidity to the denial-of-service attack (DoS) as a security requirement.

Yu *et al.* (2007) specify that security is not a quantifiable QoS, but they presented that a more secure system is the system that is able to detect the malicious actions, code or attacks in a Web service system. Accordingly, they present a formula to test the security of Web Services based on the number of attack detections, as follows:

$$QoS_{Security} = \frac{Total\ number\ of\ attacks\ detected}{Total\ number\ of\ attack\ simulations}$$

The security value indicates the probability of successfully defending against the attacks, where a value closer to 1 indicates a more secure service. Moreover, there are different research efforts conducted to address security for services. For example, IBM & Microsoft (2002) addressed security standards for Web Services technology. The standards define aunified approach for managing security during message exchange in a Web Services environment. The foundational standard is WS-Security, which is built on XML Signature, XML Encryption, and other standards such as WS-Policy.

Moreover, some research defines levels of security, which can be identified as a QoS evaluation approach. For example, El Yamany (2009) builds a security framework for SOA. The author specifies four security levels: high, moderate, low, and guest. Each of these levels is related to the main aspects of SOA security, including authentication, authorisation, and privacy.

- Transaction ($QoS_{Transaction}$):

Rahman & Meziane (2008) apply the ACID property to the transaction: Atomicity, which executes the whole transaction or not at all; Consistency, which maintains data consistency in updating transactions; Isolation, which isolates a transaction as if no other transaction is present; and Durability, which results in persistence.

- Regulatory ($QoS_{Regulatory}$):

Regulatory refers to compliance to the rule, specifications, standards, and laws (Rahman & Meziane, 2008). Ran (2003) defines regulatory as "a measure of how well the service is aligned with regulations".

- Exception Handling ($QoS_{ExceptionHandling}$):

Lee *et al.* (2003) presents that "exception handling is related to how the service handles these exceptions". Ran (2003) defines exception handling as how the service handles the exceptions, which are the unspecified possible outcomes and alternatives.

- Interoperability ($QoS_{Interoperability}$):

Services should be interoperable between systems with different software, programming languages, and operating systems (Lee *et al.*, 2003; Garcia & de Toledo, 2006; O'Brien *et al.*, 2007; Yu *et al.*, 2007). Yu *et al.* (2007) discussed that interoperability can be calculated as "the ratio of the total number of environments the Web service runs to the total number of possible environments that can be used", as follows:

$$QoS_{Interoperability} = \frac{Total\ number\ of\ environments\ the\ Web\ Service\ runs}{Total\ number\ of\ possible\ environments\ that\ can\ be\ used}$$

- Competence ($QoS_{Competence}$):

Competence demonstrates a provider's ability to provide a service that performs the function expected from it (Grandison & Sloman, 2000). In fact, competence is a relevant term for the environment related toservices and computing systems and it applies to entities that performan action on behalf of others (Grandison & Sloman, 2000).

Hoyle (2005) identifies competence as a quality characteristic for services and argues that people are either competent or incompetent without any varying degrees of competency. Individuals are competent, if they have the ability to produce the desired results when requiredand demonstrate performance that meets all required standards. Competence, which can be assessed under close supervision is "the ability to demonstrate the skills, behaviours, attributes, and qualifications to the level required for the job" and "a quality of individuals, groups, and organisations" (Hoyle, 2005).

Moreover, "A competent entity is capable of performing the functions expected of it or services it is meant to provide correctly and within a reasonable time scale" (Song *et al.*, 2007). Aljazzaf *et al.* (2011) evaluate the competence of services based on other QoS through long-term interaction.

- Honesty ($QoS_{Honesty}$):

Hoyle (2005) identifies honesty as a quality characteristic for services. Some researchers (Malik & Bouguettaya, 2009; Jurca & Faltings, 2003; Miller *et al.*, 2002) evaluate honesty of raters through long term interaction. Specifically, Aljazzaf *et al.* (2011) evaluate the honesty of services based on other QoS through long-term interaction. Moreover, Malik & Bouguettaya (2009) evaluate the credibility of raters in a reputation-based framework based on the evaluation of their honesty over time. The framework aims to protect the reputation system from malicious raters and to fairly assess the providers' reputations.

In addition, there are other QoS in the literature that can be considered as subjective QoS (O'Brien *et al.*, 2007; Hoyle, 2005; Ran, 2003; Larson, 1998). For example, O'Brien *et al.* (2007) define other QoS requirements for SOA such as: modifiability, "The ability to make changesto a system quickly and cost-effectively"; testability, "The degree to which a system or service aids the establishment of test criteria and the performance of tests to determine whether those criteria have been met"; and usability, "a measure of the quality of a user experience in interacting with informationor services". The authors define QoS requirements for SOA and how different QoS can be effected using SOA.

Moreover, Ran (2003) identified other QoS, which are: supported standard, stability/change cycle, and completeness. In addition, there are a domain or application specific QoS. Hoyle (2005) identifies other quality characteristics for services such as courtesy, comfort, credibility, dependability, efficiency, effectiveness, flexibility, promptness, and responsiveness. Larson (1998) identified service ability and user satisfaction as a performance measurement of service delivery. Other researchers identified quality metrics for Web sites and services such as: usability, reliability, responsiveness, accessibility, flexibility, and efficiency (Swaid & Wigand, 2009; Zeithaml *et al.,* 2002; Li *et al.,* 2002; Zeithaml *et al.,* 2000).

There are different studies to define and evaluate QoE (Lalanne *et al.,* 2012; Bouch *et al.,* 2000; Fiedler *et al.,* 2010; Varela *et al.,* 2014). Comparing QoE and subjective QoS as defined in this paper, QoE is defined as "the overall acceptability of an application or service, as perceived subjectively by the end-user" (ITU-T Recommendation, 2007). Thus, QoE is the term that is used to describe how the users are satisfied to the provided service quality (Kim *et al.*, 2010). The measured QoS is closely related to the quality of experience QoE for the end-user. (Khirman & Henriksen, 2002). QoE can be considered as a "perceived QoS", and reflects better than QoS the quality of the service as seen by users. (Yaacoub & Dawy, 2014). Therefore, the QoE literature addresses the QoS as parameters for QoE evaluation. Thus, the Objective QoS or Subjective QoS can be use as QoE parameters for evaluating services.

EXPERIMENT

This section presents the experimental results of the proposed QoS model with respect to some objective QoS. The experiments are conducted andthe results are analysed to show the effectiveness of the proposed QoS evaluation approach and to show the importance of QoS in the distinction between similar services from different service providers. In this experiment, the QoS services are monitored to evaluate their objective QoS.

In the current implementation, services are deployed on Windows machines (running Windows 8) with the following software tools, protocols, and technologies:

Web Service technology, Simple Object Access protocol (SOAP), Web Services Description Language (WSDL) protocol, Structured Query Language (SQL) database, and SoupUI tool (SoapUI, 2014) for testing and monitoring Web Services. In addition, different ping tools are used to test the network connections.

Services are implemented using Web Services technology. Specifically, WSDL is used to describe the Web Services and SOAP is used as a messaging standard. Using Java and NetBeans IDE 7.4, the Web Service providers are implemented as Enterprise Java Bean (EJB). The Web Services are deployed into GlassFish Server 4.

SoapUI is a functional testing tool for testing and monitoring Web Services. SoapUI parse the Web Service WSDL, invoke Web Services, and monitor Web Services. XML, XPath, Groovy, and Java Database Connectivity (JDBC) are used within the SoapUI to write different scripts, including collecting the QoS, monitoring Web Services, and connecting to the database.

The experiment methodology is as follows: A number of online Web Services are discovered from online registries. Some examples of Web Servicesare: WeatherSoap service, stockQuote service, IP2Geo service, shop service, and article service. The Web Services are provided by a number of service providers. Subsequently, a list of objective QoS are selected for the experiment which are $QoS_{Response}$, $QoS_{Latency}$, $QoS_{Execution}$, $QoS_{Througput}$, $QoS_{Availability}$, $QoS_{Reliability}$, and $QoS_{Accessibility}$. Then, the Web Services are invoked and monitored to evaluate their objective QoS.

Monitoring the QoS is conducted several times for duration of hours throughout days; and each time the average value is evaluated and the final value in the database is updated. Table 2 shows someof the monitored Web Services and their evaluated QoS. In the table, the Web Services are abbreviated and numbered for simplicity.

Table 2. The monitored Web Services and their evaluated QoS.

WS	QoS						
	$QoS_{Response}$	$QoS_{Latency}$	$QoS_{Execution}$	$QoS_{Througput}$	$QoS_{Availability}$	$QoS_{Reliability}$	$QoS_{Accessibility}$
WeatherSoap	311.82*	260*	51.82*	1.455	94.97	99.990	97.487
stockQuote	475.4	207	268.4	1.526	87.77	99.982	96.363
IP2Geo	241.07	214	27.07	1.422	97.13	99.993	98.65
SendService	345.3	288	57.3	1.352	57.53	99.994	98.323
Address	392	179	213	1.176	100	99.987	97.777
Globalweather	350.34	316	34.34	1.518	96.94	99.995	98.431
ShopService	777.45	317	460.45	1.692	96.81	99.994	98.263
BLZService	205	174	31.05	1.372	100	99.993	98.901
WS: Web Services, * Time in ms							

Figure 4 shows the $QoS_{Availability}$, $QoS_{Reliability}$, and $QoS_{Accessibility}$, for the monitored Web Services; and Figure 5 shows the and for the monitored Web Services. The figure shows a variation on the value of QoSfor Web Services provided by different providers. This variation can assist requestors in their selection decision.

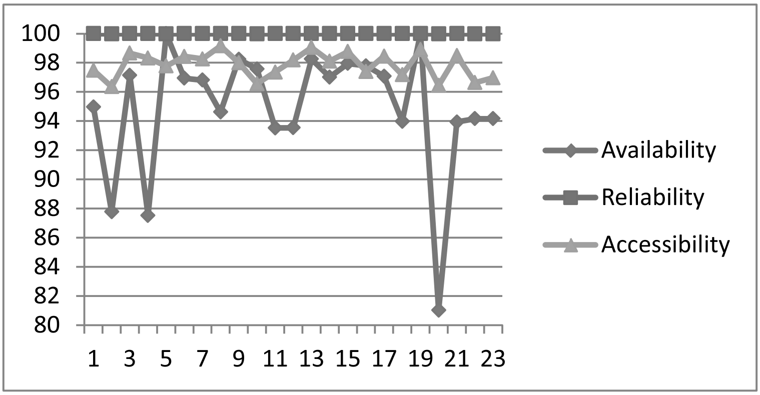

Fig. 4. The $QoS_{Availability}$, $QoS_{Reliability}$, and $QoS_{Accessibility}$ for the monitored Web Services.

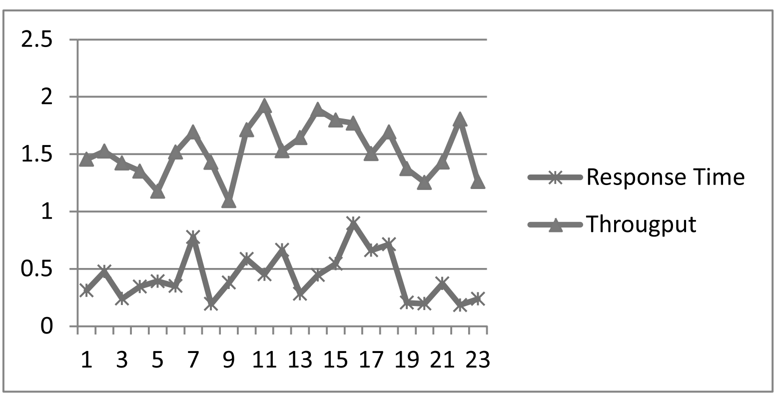

Fig. 5. The $QoS_{Response}$ and $QoS_{Througput}$ for the monitored Web Services.

A Scenario: Selecting a Web Service based on QoS

The services from different providers may have different QoS, which can assist requestors in selecting services. The following scenario represents the importance of QoS in distinguishing between similar services from different service providers.

An application developer wants to build an application that is composed of various Web Services. One of the needed Web Servicesis weather Web Service. Because there are many weather Web Services provided by different providers, the developer wants to select a Web Service with the lowest response time as his preference. Table 3 shows some QoS of the monitored weather Web Services.

Table 3. QoS of the monitored weather Web Services.

WS	QoS						
	$QoS_{Response}$	$QoS_{Latency}$	$QoS_{Execution}$	$QoS_{Througput}$	$QoS_{Availability}$	$QoS_{Reliability}$	$QoS_{Accessibility}$
WeatherSoap	311.82*	260*	51.82*	1.455	94.97	99.990	97.487
Globalweather	350.34	316	34.34	1.518	96.94	99.995	98.431
ndfdXML	379.11	37	342.11	1.318	100	99.990	96.65
WS: Web Services, * Time in ms							

Figure 6 shows the $QoS_{Response}$ for the three weather Web Services, each provided by different Web Service providers. Subsequently, the requestor can select a Web Service that has the lowest $QoS_{Response}$ value, which is the 'WeatherSoap' Web Service with $QoS_{Response} = 311.82\ ms$ from the first provider.

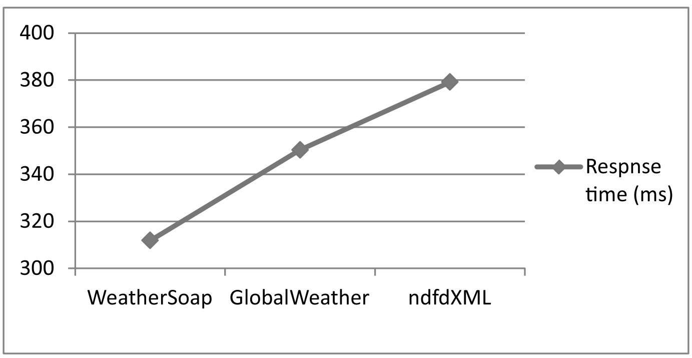

Fig. 6. $QoS_{Response}$ for the three Weather Web Services.

Similarly, Figure 7 shows the $QoS_{Availability}$, $QoS_{Reliability}$, and $QoS_{Accessibility}$ for the three weather Web Services, each provided by different Web Service providers. Requestors can select a Web Service that, for example, has the highest availability, which is the 'ndfdXML' Web Service with $QoS_{Availability} = 100$ from the third provider.

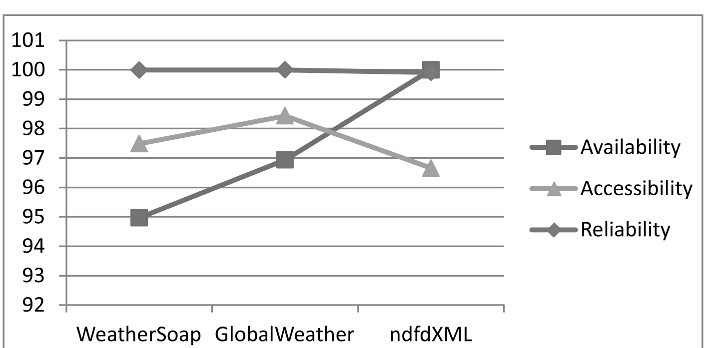

Fig. 7. $QoS_{Availability}$, $QoS_{Reliability}$, and $QoS_{Accessibility}$ for the three Weather Web Services.

CONCLUSION

QoS has been used as a distinguishing factor between similar services and as a criterion for service selection. This paper builds a generic QoS model and presents the evaluation approaches for some of the QoS. The implementation and evaluation results prove usability of the QoS model and the important of QoS for online service consumers and service providers.

Further research is needed in this area to complete the QoS model and make it more comprehensive by covering the evaluation approaches for the QoS. Specifically, future work will seek to conduct extensive research to investigate and study the evaluation approaches for other QoS from the proposed QoS model, especially the subjective QoS, such as security, transaction ACID, and regulatory.

ACKNOWLEDGMENT

This work was supported by Kuwait University, Research Grant No. [QI4/13]

REFERENCES

Aljazzaf, Z. M., Capretz, M. A. & Perry, M. 2011. Trust bootstrapping services and service providers. The Ninth Annual International Conference on Privacy, Security and Trust. Montreal, Canada.

Bouch, A., Kuchinsky, A. & Bhatti, N. 2000. Quality Is in the Eye of the Beholder: Meeting User's Requirements for Internet Quality of Service," tech. rep. HPL-2000-4, HP Labs, Palo Alto, CA.

Cabrera, O. & Franch, X. 2012. A quality model for analysing web service monitoring tools. Sixth International Conferenceon Research Challenges in Information Science (RCIS), pp. 1

Dingledine, R., Freedman, M. & Molnar, D. 2000. Peer-to-Peer: Harnessing the Power of Disruptive Technologies. O'Reilly. Chapter 16: accountability.

Dragoni, N. 2009. Toward trustworthy web services - approaches, weaknesses and trust-by-contract framework. IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology 3, 599-606.

El Yamany, H. F. 2009. A fine-grained intelligent security framework for service-oriented architecture. Ph.D. thesis. The University of Western Ontario, Ontario, Canada.

Entrust 2001. Web services trust and XML security standards.

Fiedler, M., Hossfeld, T. & Tran-Gia, P. 2010. A generic quantitative relationship between quality of experience and quality of service. Network, IEEE , vol.24, no.2, pp.36,41.

Garcia, D. Z. G. & de Toledo, M. B. F. 2006. Semantics-enriched QoS policies for web service interactions. Proceedings of the 12th Brazilian Symposium on Multimedia and the web. pp. 35-44. ACM.

Grandison, T. & Sloman, S. 2000. A survey of trust in Internet applications. IEEE Communications Surveys and Tutorials 3.

Hoyle, D. 2005. Automotive Quality Systems Handbook. 2nd edition, Elsevier Ltd, pg. 419.

Hsu, W., Yeh, S., Shieh, Y. & Hsieh, C. 2013. Web-Based QoE Measurement Framework. International Conference on Signal-Image Technology & Internet-Based Systems (SITIS),pp.265,272. IEEE.

Huhns, M. N. & Singh, M. P. 2005. Service-oriented computing: Key concepts and principles. IEEE Internet Computing 9, 75-81.

IBM Corporation & Microsoft Corporation 2002. Security in a web services world: a proposed architecture and roadmap. Technical Report.

ITU-T Recommendation 2007. Definition of Quality of Experience (QoE).P.10/G.100 Amendment 1, New Appendix I.

Jurca, R. & Faltings, B. 2003. An incentive compatible reputation mechanism. AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems (AAMAS). 1026-1027.

Kalepu, S., Krishnaswamy, S. & Loke, S. 2003. Verity: a QoS metric for selecting web services and providers. Proceedings Fourth International Conference on Web Information Systems Engineering Workshops (WISEW), 131-139.

Khirman, S. & Henriksen, P. 2002. Relationship between quality-of-service and quality-of-experience for public internet service. Proceedings of the 3rd Workshop on Passive and Active Measurement.

Kim, H., Kim, B. K., Choi, H. H. & Choi, S. 2010. Implementation of QoS control system with QoE parameters on multimedia services. 12th International Conference on Advanced Communication Technology (ICACT), vol.2, pp.1035,1040.

Kim, H. J., Lee, D. H., Lee, J. M., Lee, K. H., Lyu, W. & Choi, S. G. 2008. The QoE Evaluation Method through the QoS-QoE Correlation Model. Fourth International Conference on Networked Computing and Advanced Information Management (NCM), vol.2, pp.719,725.

Kim, H. & Choi S. 2010. Design of QCS with QoE parameters on multimedia service and its performance analysis. Digest of the 9th International Conference on Optical Internet (COIN).

Kim, Y. & Doh, D. 2007. A trust type based model for managing QoS in web services composition. International Conference on Convergence Information Technology 0, 438-443.

Lalanne, F., Cavalli, A. & Maag, S. 2012. Quality of Experience as a Selection Criterion for Web Services. Eighth International Conference on Signal Image Technology and Internet Based Systems (SITIS), pp.519,526.

Larson, K. 1998. The role of service level agreements in IT service delivery. Information Management and amp; Computer Security 6, 128-32.

Lee, K., Jeon, J., Lee, W., Jeong, S. & Park, S. 2003. QoS for Web Services: Requirements and Possible Approaches. Technical Report. pg. 1, W3C, Web Services Architecture Working Group. URL:http://www.w3c.or.kr/kr-office/TR/2003/ws-qos/, last accessed on May, 2014.

Li, Y. N., Tan, K. C. & Xie, M. 2002. Measuring web-based service quality. Journal of Total Quality Management, Vol. 13, Iss. 5, pp. 685-700(16).

Liu, Y., Ngu, A. & Zeng, L. 2004. QoS computation and policing in dynamic web service selection. Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters (WWW), ACM, New York, NY, USA. pp. 66-73.

Malik, Z. & Bouguettaya, A. 2009. Rater credibility assessment in web services interactions. World Wide Web journal 12, 3-25.

Mathijssen, S. 2005. A fair model for quality of web services. 3rd Twente Student Conference on IT, Enschede .

Maximilien, E. & Singh, M. 2004. Toward autonomic web services trust and selection. in: Aiello, M., Aoyama, M., Curbera, F., Papazoglou, M.P.(Eds.). Proceedings of Second International Conference inService-Oriented Computing (ICSOC), ACM, New York, NY, USA. pp. 212-221.

Mell, P. & Grance, T. 2011. The NIST definition of cloud computing: Recommendations of the national institute of standards and technology. National Institute of Standards and Technology, Information Technology Laboratory, Computer Security Division, US Department of Commerce, NIST Special Publication, pp. 1-12.

Menasce, D. 2002. QoS issues in web services. IEEE Internet Computing 6, 72-75.

Menasce, D. A. & Almeida, V. 2001. Capacity Planning for Web Services: Metrics, Models, and Methods. 1st ed., Prentice Hall PTR, Upper SaddleRiver, NJ, USA.

Miller, N. H., Resnick, P. & Zeckhauser, R. J. 2002. Eliciting Honest Feedback in Electronic Markets. Working Paper Series rwp02-039. Harvard University, John F. Kennedy School of Government. URL:http://ideas.repec.org/p/ecl/harjfk/rwp02-039.html, last accessed on May 2014.

Moorsel, A. 2001. Metrics for the internet age: Quality of experience and quality of business. 5th Performability Workshop.

O'Brien, L., Merson, P. & Bass, L. 2007. Quality attributes for service-oriented architectures. Proceedings of the International Workshop on Systems Development in SOA Environments (SDSOA), IEEE Computer Society, Washington, DC, USA. p. 3.

Papazoglou, M. 2012. Web Services and SOA: Principles and Technology. Pearson Education, Essex, England ; New York.

Papazoglou, M. P. & Georgakopoulos, D. (Eds.) 2008. Service-Oriented Computing. The MIT Press, Cambridge, MA, pg. 1.

Papazoglou, M. P., Traverso, P., Dustdar, S., Leymann, F. & Krmer, B. J. 2006. Service-oriented computing research roadmap. Dagstuhl Seminar Proceedings 05462, pp. 1-29.

Rahman, W. & Meziane, F. 2008. Challenges to describe QoS requirements for web services quality prediction to support web services interoperability inelectronic commerce. Proceedings of the 10th IBIMA Conference on Innovation and Knowledge Management in Business Globalization, Kuala Lumpur, Malaysia, 30 June - 2 July 2008, 4 (6) , pp. 50-58.

Ran, S. 2003. A model for web services discovery with QoS. ACM SIGecom Exchanges 4, 1-10.

Rosen, M., Lublinsky, B., Smith, K. T. & Balcer, M. J. 2008. Applied SOA: Service-Oriented Architecture and Design Strategies. Wiley Publishing.Pp. 28, 37.

Rosenberg, F., Platzer, C. & Dustdar, S. 2006. Bootstrapping performance and dependability attributes of web services., IEEE Computer Society. Pp. 205-212.

Sadiku, M., Musa, S. & Momoh, O. 2014. Cloud computing: Opportunities and challenges. Potentials, IEEE 33, 34-36.

SoapUI, http://www.soapui.org/ - last accessed on May, 2014.

Song, R., Korba, L. & Yee, G. 2007. Trust in E-services: Technologies, Practices and Challenges. IGI Global.

Swaid, S. & Wigand, R. 2009. Measuring the Quality of E-Service: Scale Development and Initial Validation. Journal of Electronic Commerce Research, 10 (1), 13-28.

Varela, M., Skorin-Kapov, L. & Ebrahimi, T. 2014. Quality of Service Versus Quality of Experience Quality of Experience, T-Labs Series in Telecommunication Services, pp 85-96.

Wang, Y. & Vassileva, J. 2007. A review on trust and reputation for web service selection. (ICDCSW) IEEE Computer Society, Washington, DC, USA. p. 25.

Wei, F., Xu, J., Wen, T., Liu, X. & Yan, H. 2012. Smart phone based online QoE assessment for end-to-end multimedia services on 3G mobile Internet. In 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), pp. 2861-2865. IEEE.

Yaacoub, E. & Dawy, Z., 2014. Network QoE metrics for assessing system-level performance of radio resource management algorithms in LTE networks.International Conference on Wireless Communications and Mobile Computing (IWCMC). pp.411,416, 4-8.

Ying-Feng, Z. & Pei-Ji, S. 2006. The model for consumer trust in C2C online auction. International Conference on Management Science and Engineering (ICMSE).pp. 125-129.

Yu, W. D., Radhakrishna, R. B., Pingali, S. & Kolluri, V. 2007. Modeling the measurements of QoS requirements in web service systems. Simulation journal, 83, 75-91.

Zeithaml, V., Parasuraman, A. & Malhotra, A. 2002. Service quality delivery through the Web sites: A critical review of extant knowledge. Journal of the Academy of Marketing Science, 30 (4), 362-375.

Zeithaml, V., Parasuraman, A. & Malhorta A. 2000. A Conceptual Framework for Understanding E-Service Quality: Implications for Future Research and Managerial Practice.Marketing Science Institute (MSI), Report # 00-115.

Zhang, J., Zhong, F., Yang., Z. & Liu, H. 2012. A QoS-aware computation model for dynamic web service selection. IEEE 12th International Conference on Computer and Information Technology (CIT), pp. 230-235.

Zheng, Z., Zhang, Y. & Lyu, M. 2014. Investigating QoS of real-world web services. IEEE Transactions on Services Computing, 7, 32-39.

Submitted **:** 12/06/2014

Revised **:** 12/10/2014

Accepted **:** 27/10/2014

بناء نموذج وقياس جودة خدمات الويب

زينب محمد الجزاف
قسم علوم المعلومات – كلية علوم وهندسة الحاسوب – جامعة الكويت

خلاصة

قد تتألف تطبيقات الويب من خدمات عالمية ذات خصائص مختلفة وتقدمها جهات مختلفة. يعد توفير خدمات مشروطة ذو أهمية كبرى ضمن أساليب حوسبة الخدمات نظراً لاختلاف خصائص هذه الخدمات من جهة وتوفيرها من قبل عدة مقدمين من جهة اخرى.لتحديد خدمة من بين العديد من الخدمات المماثلة فإنه من المهم التمييز بينها. تستخدم جودة الخدمة للتمييز بين الخدمات المماثلة و كمعيار لانتقاء الخدمات. معظم الابحاث في مجال جودة الخدمة ليومنا هذا لا تحدد بشكل قاطع جودة خدمة شاملة مناسبة لبيئات شبكية مفتوحة ذات خدمات متعددة. علاوة على ذلك فان كثير من الدراسات المعنية ببناء نماذج جودة الخدمة تتجاهل الكثير من مقاييس جودة الخدمة. لذلك كانت هناك حاجة لنموذج شامل لجودة الخدمة. وعليه يقدم هذا البحث نموذج عام لجودة الخدمة يحدد مقاييس شاملة لجودة الخدمة واساليب تقييمها. نتيجة لذلك تعد هذه الدراسة كمرجع مفيد للاكاديميين والباحثين والعاملين بمجال ادارة خدمات الويب. تم اجراء تجارب لاختبار عدة مقاييس لجودة الخدمة من النموذج المقترح.

الكلمات المفتاحية: جودة الخدمة ، نموذج جودة الخدمة ، حوسبة الخدمات ، الخدمات.