## Towards a Comprehensive View of Web Services QoS Models

Jokha Al-Kalbani<sup>1</sup>, Naoufel Kraiem<sup>1,2</sup>, Zuhoor Al-Khanjari<sup>1</sup> and Yassine Jamoussi<sup>1,2</sup>

<sup>1</sup>Department of Computer Science, College of Science, Sultan Qaboos University, PO Box 36, Al-Khoudh 123, Muscat, Sultanate of Oman <sup>2</sup>RIADI Lab, ENSI, Campus of Manouba, Tunisia p890717@Student.squ.edu.om {naoufel,zuhoor,yessine@squ.edu.om}

#### Abstract

In a growing web services environment, the usage, design and composition methods of web services require proper and reliable information about web services quality. Qualityof-Service (QoS) is usually referred to the non-functional characteristics of web services which can be used as an important role to differentiate between different web services. In order to select the best web services among those with similar functionality, it is important for the web requester to have a mechanism to rank them and also from the service providers side to ensure that their web services are according to the agreed QoS. Currently there is no standard manner to define the QoS characteristics of web services. Many studies have shown different categorization of QoS which led to the issue of semantic interoperability of QoS. One way to understand the QoS, is to identify all its possible requirements for web services. This paper is an attempt to explore some of the OoS issues which are essential for web services. It first proposes a OoS framework which suggests considering QoS for web services along four different views. Each view is representing a particular relevant facet of QoS models. The main contribution of this paper is to show that there is a big quantity and variety of interpretations of various service quality concepts and models and gives the reasons why there is no holistic QoS modeling approach.

Keywords: QoS Model, SOA, Comparison QoS Framework, Web Services, WSDL, BPEL

#### **1. Introduction**

Web services technology has been one of the dominant technologies for software development since it enables rapid flexible development and integration of software systems. Web services are considered as the basic building entities. They can be defined as software units offering certain functionalities over the web and using a set of interface and protocol standards, *e.g.* Web Services Definition Language (WSDL) as a service contract, Simple Object Access Protocol (SOAP) as a messaging protocol, and Business Process Execution Language (BPEL) as a flow-based language for services composition [1]. As more and more web services are available on the Internet, this increased the need to have a mechanism to test and evaluate these services. In this context, different service quality concepts for describing QoS information have been introduced. This leads to a semantic interoperability issue for QoS [2]. The QoS information has been used for calculating overall quality degrees of web services with respect to a requester's QoS needs [3]. With this growing demands and to meet the need of different user groups, many service providers have started to offer different QoS levels [4].

According to [5], a variety of factors, like market conditions, resource constraints and inadequacy of available developers, may play an important role in the issue of

discrepancy between the actual Service Level Agreement (SLA) and the real needs of customers. As a result, the gap often appears between the requestor's point of view and the service owner's consideration of those needs. The need analysis or requirement analysis is a very important aspect to be clarified when developing any new software. Al-Khanjari (2014) [6] proposed a tool as a cross check technique to help in reducing the conflicts between software developers and system stakeholders. In order to compromise requirements stated on the basis of different viewpoints, Quality Models have been introduced as the engineering artifact to define the quality factors that apply to web service usage. These are used to produce a common framework of understanding for QoS. In general, the proposed approaches for evaluation of QoS capabilities for web services are quite different from each other. Each one focuses on a different set of QoS metrics and can be applied at runtime or system design time. Also, the existing approaches are built upon different QoS models and techniques. In this scenario, it can be difficult to choose the right approach which can be applied to a specific QoS evaluation study.

This paper addresses the issue of different QoS models and techniques to evaluate the QoS for web services. It proposes a framework that considers the web services QoS models from four different viewpoints. Each view has a set of attributes to describe a particular and specific aspect of service. This framework is built on top of the framework developed by [7]. It makes changes to: (i) the attributes that characterize every aspect, and (ii) the values these attributes can take with respect to the aspect analysis. The remainder of this paper is formulated as follow: Section 2 is an overview of the proposed framework. Section 3 provides a detailed description of the framework. Section 4 gives a review of four different existing QoS models. Finally, Section 5 concludes the paper and draws some future directions.

## 2. QoS Framework Overview

Many proposals for general purpose quality models of web services exist. These quality models differ from each other *e.g.* the usage of terminologies, definition of a set of quality attributes and the representation of quality model structure. Also, QoS is a broad concept that contains a large number of domain-specific and context-dependent non-functional properties, it is not sufficient or practical to develop a standard QoS model that can be used for all web services in all domains. The proposed framework is developed to compare different QoS models in order to characterize and classify them based on multiview and multi-faceted approaches. The multi-view approach consists of four different views. Each view is associated with a set of facets considered as viewpoints or dimensions and provides a specific knowledge and analysis of QoS. A fundamental perspectives of QoS will be presented by answering four questions: What (Nature), Why (Objectives), Which (Process) and How (Form). The facets are defined using a set of relevant attributes which is described by a set of values that are defined within a domain to measure the observed aspect. A domain can be one of a predefined type such as an enumerated type ( $\{x,y,z\}$ ), a structured type (Set  $\{x,y\}$ ), an Integer or Boolean.

As illustrated in Figure1, the proposed framework is composed of Nature view, Objective view, Process view and Form view. Detailed descriptions are provided below.

International Journal of Multimedia and Ubiquitous Engineering Vol.10, No.10 (2015)



Figure 1. QoS View Comparison Model

#### 2.1. QoS Nature View

QoS nature view describes the subject of QoS model, in terms of properties and classifications. It consists of four facets: Classification level, QoS modeling approach, Actor and Granularity. They are described further as follows.

#### 2.1.1. Classification Level Facet

Many classifications of QoS models are proposed in the literature and analyzed by the research community. The classification level facet specifies different usage methods so that a requester can choose the most suitable quality level for his demands.

OASIS [8] model classified the quality levels into three levels: i) Business level which represents the quality of business value realized by the user while using web services. It includes quality factors like Price, Penalty & Incentive, Service Recognition, Business Performance, Service Reputation and Service Provider Reputation. ii) Service level which represents the measurable performance of QoS is realized by the user while using web services. It includes mainly performance issues such as Maximum Throughput, Response Time, Availability, Accessibility and Success-ability. iii) System level is divided into three factors, the Interoperability factor that identifies whether different web services which are developed by different developers in different system environments can properly interoperate. The Security quality factor is to indicate the reaction of web services to the attack from outside or unauthorized access. The Manageability factor indicates the manageability from outside or within the system. Thus the classification level attribute of nature view can be defined as:

Classification level: SET (ENUM {Business, Service, System, Technical})

**2.1.2. QoS Modeling Approach Facet:** Most of the proposed QoS models fall into Taxonomy-based, Ontology-based and Activity-based classifications [9]. Where Taxonomy-based quality models setup a quality along with the characteristics (*e.g.* reliability, interoperability, efficiency, usability, portability, maintainability and security). Ontology-based quality models are usually taxonomy-based ones which define QoS characteristics in a formal way. Activity-based quality models described the quality along with the activities performed within or on a SOA system. Thus the QoS modeling approach of nature view can be defined as:

QoS modeling approach facet: SET(ENUM {Taxonomy-based, Ontology-based, Activity-based })

**2.1.3.** Actor Facet: Different actors are involved in different activities during the web service development life cycle *i.e.* the development, maintenance and service consuming activities. They have different, sometimes conflicting understanding of what is a quality of web service and how it should be described, specified and evaluated. To achieve better overall QoS, the different viewpoints and perspectives of quality should be considered. An overview of basic responsibilities of actors explained as following. Developers play a key role in helping and offering clients to achieve their business goals through different advanced technology solutions. Providers are the owners of the service. Consumers are the requesters who need certain functions to be satisfied. Managers are the systems administrators and monitors for the QoS of web services. The Actors facet attributes can be defined as:

Actors facet: SET (ENUM {Developers, Providers, Consumers, Managers})

**2.1.4. Granularity Facet:** The service size and the scope of functionality a service exposes refers to a service granularity or it may refer to the amount of detail associated with service design and it has an impact on the ability to reuse or compose the service in different contexts [10-11]. Along with the reusability and compos-ability, the performance of the service will also be affected from the inappropriate granularity design. In our proposed framework we will analyze two attributes on the QoS models, the atomic service and the aggregate one. Atomic service is a single interaction for a business process or an individual service. An aggregate service is formulated from many atomic and composite services. The Granularity facet attributes can be defined as:

Granularity facet: SET (ENUM {Atomic Service, Aggregate Service})

## 2.2. QoS Objective View

The objective view explains the why of the approach. This view captures why we should construct QoS models for web services and what are the benefits those will be achieved from its practical usage on the Internet. Service consumers can formalize their real intentions if they can express their QoS requests more effectively. Also, the design of a specific QoS query language could be useful as pointed out in [12-14] in improving the quality of service matching with focus on query and rule layers in the context of Semantic Web. Web services can be considered as third-party software made available by service providers. Therefore, it is their responsibility to ensure that their services according to the agreed QoS. Using QoS models to analyze and evaluate the QoS related issues becomes crucial for many web service activities. Such models were designed form different perspectives and since the user requirements and technologies are changing over time, these models should support these evolutions in a consistent manner. Also, it is important to identify and use some criteria to evaluate these models in order to conclude which is the most consolidated body of knowledge among the evaluated services. With the growing interest to implement a reusable design components, where the reuse is an important aspect which can occur at any stage of life cycle development. There is a need to introduce new methods to manage and capture any new aspect related to QoS for different activities of web services.

In line with the above, we proposed two important facets for the objective view in order to demonstrate the importance of these facets in the evaluation process of QoS models:

**2.2.1. Policy Facet:** There is a variety of participants in web services cooperation. Each participant can specify QoS descriptions and requirements which should be certified in order to get expected QoS from utilizing the specific web services. So different policies

are needed to capture the different QoS properties defined by different stakeholders and also because different services requesters have different levels of expectation there are important to them.

The security and management policies attributes must be taken into consideration when evaluating QoS policies. The security related quality will be used to provide integrated security service and to control any legal access for the use of an appropriate authority with reliability and stability. A manager or web services tool developer viewpoints will be presented in the management related quality where qualities can be managed by using object properties like the relationship between objects, status and any events to manage web services system. Miscellaneous attribute has been added to cover any other policies involved in the evaluated QoS model. The proposed policy facet attributes can be defined as:

Policy facet: SET (Boolean {Security policies, Management policies, Miscellaneous})

**2.2.2. Decision Making Facet:** Decision making is a very important task in business process for web services which would allow considering users demands and their capabilities in decision making in the business. The decision making facet works similar to a recommendation system which guides the process of web services selection toward the realization of its goal. These guidance processes have different methods to execute the most suitable service to the usage context. Some QoS models depend on reasoning for evaluating good web services from bad ones while other methods will vary from using simple rules to very advanced ones, also ad-hoc decision has been used in some models when there is no structured approach for decision making. The proposed decision making facet attributes can be defined as:

Decision making: SET (Boolean {Reasoning, Simple-Rules, Advanced-Rules, Ad-hoc Decision})

#### 2.3. QoS Process View

Process view explains the which of the approach. It considers the different ways of usage of QoS model and which tools and techniques are used to realize the QoS model. Managing of QoS properties occurs at different levels of abstraction during the development cycle of the web services. Therefore, our proposed framework concentrated on two main facets that cover the time of evaluation and the techniques used during this process. In order to demonstrate that, there is still no consensus on a general QoS evaluation techniques nor on the best time to run the evaluation process. Process view consists of three facets as follows:

**2.3.1. Quality Assessment Facet:** The validation of QoS can be done in any of the two phases: development time or runtime, where there remains a major gap in tools and techniques used for each phase. The validation in development time will enable design flaws to be rectified earlier in the development life cycle, while the runtime validation will proceed only after all web service components are available and deployed in the runtime infrastructure. The development time validation process will cover the design phase, implementation phase, testing phase and deployment phase. While the runtime validation will cover the find operation, bind operation and ad-hoc operation which have been added to cover any extra different validation activity during the web service runtime. The proposed quality assessment facet attributes can be defined as:

Quality assessment facet: SET (ENUM {Development time, Runtime})

**2.3.2. Evaluation Techniques Facet:** The execution environment nature of web services plays an important role on their QoS. The impact of the dynamic growth and unpredictability of this environment can be seen on increasing the demands for evaluating and testing the offered web services in order to make sure that the selected ones are with

high QoS. Thus, different techniques have been proposed to assure the agreed QoS requirements for a web service has been met. The aim of these techniques is to evaluate the QoS and uncover quality defects in the web services after development phase. The evaluation techniques that are covered on our proposed framework are as follows:

- Simulation technique: Is one of the dynamic analytical techniques that can be used to predict QoS for software applications in different modes and conditions of the execution environment [15]. Also, the simulation technique will provide an environment for tuning and evaluation process of web services without suffering from the cost of enacting them.
- Monitoring technique: This technique will handle the observation, collection, and reporting process of all QoS information about the execution and evaluation of web services. [16-17] expressed that the QoS monitoring is important to the extent that it can be used to analyze the problem that occurs in the services and may also become the determining factors for customers as to whether continue using the service or not.
- Matchmaking technique: The discovery process for web services requires a service requester to provide some information on the capability of a requested service. On the other hand, many service providers depending on advertising the service capabilities in order to publish their services. Therefore, matching technique between the user requirements and advertised capabilities of service providers will involve using a service discovery engine to perform the matchmaking of the non-functional and functional properties from many similar services that are resulted from the discovery process.
- Process mining technique: Managing the QoS for processes and workflows is one of the important requirements for Business Process Management Systems (BPMS) along with the Workflow Management Systems (WfMS) [18]. Any design or composition of a process should be undertaken with the emphasis of the importance of QoS management. Hence, the QoS specification must be included in the workflows design. The goal of the underlying management system is to generate data from the execution of process models and workflows that describe all activities being performed and stored in a log file. The process mining technique will use the data from the log file to discover and extract information about the execution and structure of processes.
- Data mining technique: Is used when trying to examine data resulted from search of systematic relationships and/or consistent manners between variables and apply the detected patterns into new subset of data in order to validate the findings [19-20]. Prediction is targeted by data mining where predictive data mining is considered as the most common type of data mining. Therefore, organizations started applying data mining algorithms to predict the QoS for their running workflow instances.

The proposed evaluation technique facet attributes can be defined as:

Evaluation techniques: SET (ENUM {Simulation, Monitoring, Matchmaking, Process Mining, Data Mining})

**2.3.3. Tool Support Facet:** This facet deals with the tools used for the construction and supporting of QoS models. According to [21], Service Oriented Systems (SOS) uses services which are loosely coupled and autonomous abstract modeling entities to provide the basis for dealing with software integration and composition. [22] support the addressing of QoS issues in design processes by using Attribute Driven Design (ADD) method to understand the relationship between software qualities and architectural mechanisms. There are several tools that support web QoS modeling and analysis. Hence the tool support facet attribute that can be defined as:

Tool support: TEXT

#### 2.4. QoS Form View

Form view explains the how of the approach. It describes how QoS model is represented and what are the methods used to achieve the goal of QoS evaluation. A number of requirements participating in comparison process for existing QoS models. These requirements can be categorized into two essential questions: first, how to model QoS properties and second, how to use QoS properties to evaluate and rank web services. In order to achieve this goal we should understand and highlight the different aspects that describe the web services. Indeed, we focus on the methods representation of QoS models. In this view we will concentrate on three facets: Notation, Formalism and Models.

**2.4.1. The Notation Facet:** There is no fixed defined standard for modeling web services, however there are several approaches in which UML is used in different ways to model them [23]. Model Driven development helps to focus on the important aspects of the system and allow to delay the decision about the implementation technology at a later stage. This facet captures the notation nature used in the proposed modeling method. The notation facet attributes can be defined as:

Notation: SET (ENUM {Standard, Owner, Mix})

**2.4.2. Formalism Facet:** According to [24], it is a very hard task to specify requirements using a mathematical notation, even a superior modeler in the mathematical field may have problems to address them properly. Therefore, to specify non-functional requirements they defined a requirement language to provide the required formalism. Also they pointed out the need for formal specification of non-functional requirements when they described that semi-formal models can be enriched by quantitative data. This can be extracted from the non-functional requirements of the system, and make sure that before system deployment it should be specified and validated against the system model. The formalism facet attributes can be defined as:

Formalism: SET (ENUM {Formal, Semi-Formal, In-Formal})

**2.4.3. Models Facet:** There are many different concepts, metrics, units and value types that may be used in different QoS models and languages to represent QoS information. It is hard for a web services system to execute QoS information collected from various web services in order to decide the best ones based on their quality. Therefore, various types of models can be applied to represent specific characteristics of classes of systems. A general QoS specification language called QoS Modeling Language (QML) has been proposed by [25], which can be used to capture QoS properties as part of Object-Oriented designs. In our proposal we categorized the QoS models into two aspects, knowledge-oriented and data-oriented models. Where the knowledge-oriented models can be treated as a repository of facts about web services related QoS parameters. Data-oriented models will use a classification data mining algorithm to classify web service candidates into different QoS levels according to the defined constraints from the user which defines a utility value for each of the service candidates [26]. The models facet attributes can be defined as:

Models: SET (ENUM {Knowledge-Oriented, Data-Oriented})

## 3. Review of Four QoS Models Using the Proposed Framework

In this section, the web services QoS model framework will be used to illustrate the use of QoS models for evaluation. Four different QoS models have been chosen: OASIS [8], SQuaRE [27], DAML-QoS [28] and onQoS-QL [29]. We aim to get better understanding of QoS models and their attributes and also to show that by analyzing various quality concepts and models there will be a big quantity and variety of interpretations of these concepts. Table 1, Table 2, Table 3 and Table 4 give a comparison of the four QoS

models. The proposed framework represents four views of QoS model namely: Nature, Objectives, Process and Form.

The comparison activity is intended to obtain the strength and limitations of evaluated models with a special highlight on their inability to model QoS adequately in Service-Oriented enterprise systems context.

QoS	Nature View											
Model	Classification Level			Actors				QoS Modeling Approach			Granularity	
	Business	Services	System	Dev	Con	Prv	Mng	Tax- based	Actv- based	Ont- based	Atc	Agg
OASIS	Price, Penalty & Incentive, Business Performance, Service Recognition, Service Reputation, Service Provider Reputation	Response Time, Maximum Throughput, Availability, Accessibility, Successability	Interoperability, Security, Manageability	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes
SQuaRE	Efficiency, Reliability,	Functionality, Usability, Portability	Interoperability, Security, Maintainability	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes
DAML- QoS	Not Considered	Not Considered	Not Considered	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes
onQoS- QL	Not Considered	Not Considered	Not Considered	Yes	Yes	Yes	No	No	No	Yes	Yes	Yes

## Table 1. Instantiation of Four QoS Models According to Nature View of the Proposed QoS Framework

Table Ab	breviation:					
• • •	Dev: Developers Con: Consumers Prv: Providers Mng: Managers	• •	Tax: Taxonomy Actv: Activity Ont: Ontology	:	Atc: Atomic Agg: Aggregate	

From the nature view in Table 1 we conclude the following:

- Different QoS models focus on different QoS properties depending on different stakeholders' point of view.
- Actors play major roles and participate in different service life cycle phases.
- It is important to categorize services by their relations with business tasks or processes. This categorization will lead to analyze the QoS models at least from two perspectives, technological perspective (*i.e.* service quality group and system quality groups) and business perspective.
- The proposed classification for QoS models have their own limitations. [30] pointed out that, i) the classification approach used in taxonomy-based models usually has no clear definition for the relationship between super and sub

characteristics. ii) the activity-based models are product oriented with too much technology and encompass only three viewpoints.

• Whether atomic or composite, a Web service is described by an identifier (*e.g.* URL), attributes and a set of operations. The information provided by attributes of a service is very useful for the service's potential consumers.

#### Table 2. Instantiation of Four QoS Models According to Objective View of the Proposed QoS Framework

QoS Model	Objective View									
		Policy Facet	Decision Making Facet							
	Security Policies Management		Miscellaneous	Reasoning	asoning Simple-		Ad-hoc			
		Policies			Rules	Rules	Decision			
OASIS	Yes	Yes	Not defined	Not defined	Not defined	yes	Not			
							defined			
SQuaRE	Yes	Not defined	Not defined	Not defined	Not defined	Not defined	yes			
DAML-QoS	Not defined	Not defined	Yes	Yes	Not defined	Not defined	Not			
							defined			
onQoS-QL	Not defined	Not defined	Yes	Yes	Not defined	Not defined	Not			
							defined			

From the objective view in Table 2 we conclude the following:

- There is no clear common policies for the QoS models, where there should be Service Level Agreement (SLA) contracts and policies for each actor involved in the web service life cycle with an effective and a clear policy for each interaction.
- Management policies should contain information for each interaction via messaging activities in each process instance on the real QoS properties.
- Confidentiality and integrity requirements should be applied for all interaction activity by using a WS-Security based middleware service where not all QoS models consider this aspect as an important milestone when they construct their QoS model.
- One of the primary roles of QoS models should be to ensure obtaining a manageable and secure service at the same time in order to satisfy the users with clear and negotiable QoS requirements and these manageability obligations may be expressed in a policy document.
- The guidance processes available for users to realize the most appropriate service to their context vary from one model to another. Different decision making methods normally create different outcomes and enhance the accuracy of the final decision. Therefore, we need to have a more efficient tool.

## Table 3. Instantiation of Four QoS Models According to Process view of theProposed QoS Framework

QoS Model	Process View									
	Quality Assessmer	nt		Tool						
				Support						
	Development Runtime		Simulation	Monitoring	Matchmaking	Process	Data	Tool		
	time					Mining	Mining	Support		
OASIS	Design phase,	Find	Not defined	Yes	Not defined	Not	Not	OASIS		
	Implementation	operation,				defined	defined	Open		
	Phase, Testing	Bind						Composite		
	phase,	operation						Services		
	Deployment							Architecture		
	phase							(CSA)		
SQuaRE	Design phase,	Find	Not defined	Yes	Not defined	Not	Yes	Not defined		
	Implementation	operation,				defined				

	phase, Testing	Bind						
	phase,	operation						
	Deployment							
	phase							
DAML-QoS	Not defined	Find	Not defined	Not defined	Yes	Not	Not	Not defined
		Operation,				defined	defined	
		Bind						
		Operation						
onQoS-QL	Not defined	Find	Not defined	Yes	Yes	Not	Not	Not defined
		operation,				defined	defined	
		Bind						
		operation						

From the process view in Table 3 we conclude the following:

- Evaluation methodology depends on the coverage area of web service life cycle. Not all QoS models cover all tasks of development and execution of web services. Also the evaluation during the development time has its own tools and techniques which differ from the used one during the evaluation at runtime. It is important that, each evaluation phase has sufficient details to carry out the evaluation process in an effective manner.
- Many different analytical quality assurance techniques have been used by QoS models to evaluate QoS and discover quality defects in the web service. Therefore, more systematic and logical scientific procedure need to be developed in order for Web service designers to achieve the optimum web design.
- Each QoS model is inherited from different approaches and stakeholders point of views and focuses on essential properties that play an important role for the effective measurement of web services. These properties are measured using different techniques. This leads to a difficulty to choose the suitable evaluation technique depending on the QoS model approach.
- Different supporting tools are used to obtain QoS models depending on the way web services are designed and built.

# Table 4. Instantiation of Four QoS Models According to Form View of theProposed QoS Framework

QoS	Form View									
Model		Notation			Formalism	l	Model			
	Standard	Owner	Mix	Formal Semi- Inform		Informal	Knowledge-	Data-		
					Formal		Oriented	Oriented		
OASIS	Yes	No	No	Yes	Not	Not	Not defined	Not		
					defined	defined		defined		
SQuaRE	Yes	No	No	yes	Not	Not	Not defined	Yes		
					defined	defined				
DAML-	Yes	Not	Not	Yes	Not	Not	Yes	Not		
QoS		defined	defined		defined	defined		defined		
onQoS-	Yes	Not	Not	Yes	Not	Not	Yes	Not		
QL		defined	defined		defined	defined		defined		

From the form View in Table 4 we conclude the following:

- There are no standard notations that can be followed or suitable for all QoS models. Therefore, notations with different levels of abstraction and formalities are used to represent the QoS models.
- Different types of models can be applied to represent specific characteristics of QoS models like knowledge-oriented or data-oriented.

## 4. Conclusion and Future Work

The complexity and the quality of a system are determined by functional and nonfunctional properties. Non-functional properties of QoS models address critical aspects which determine the usability of a system. They can affect not only the development process of a system but also the software product. Modeling approaches for QoS of web services in Service Oriented Architecture (SOA) range from informal to UML-based solutions. In order to apply non-functional properties properly, it is important to understand the benefits they offer and their limitations in order to enhance them. The suggested framework allows a comparison structured in four views. The different viewpoints of the framework allow identifying and clarifying a particular aspect of QoS for web services. This study shows that we cannot use the same evaluation techniques to evaluate the QoS models, also the analysis of four different QoS models revealed the lack of well-established and standard OoS model for services. In a similar manner, it has been found that, it is difficult to standardize service quality because results from these services varies from day-to-day and from customer-to-customer. Also the dynamic and unpredictable nature of the execution environment has an important impact on QoS of web services.

To overcome these limitations a new approach equipped with tools and techniques should be introduced to support self-healing and adapting for the dynamic service composition in order to provide continues response with desired QoS under execution environment. For future work, a new approach for managing QoS properties need to be formulated. This approach will address the management issues of non-functional properties of QoS models. It will also explore the effect of these properties in the architectural and design decisions of a system in order to improve services management in terms of monitoring and controlling.

## Acknowledgements

This work was supported by The Research Council (TRC) of Oman (ORG/ICT/14/004).

## References

- [1] T. Erl, "Soa: principles of service design", Upper Saddle River: Prentice Hall, vol. 1, (2008).
- [2] Z. Zheng and M. R. Lyu, "Personalized reliability prediction of web services", ACM Transactions on Software Engineering and Methodology (TOSEM), vol. 22, no. 2, (2013), pp. 12.
- [3] J. Zhou and E. Niemela, "Toward semantic qos aware web services: Issues, related studies and experience", Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, (2006), pp. 553-557.
- [4] T. Yu and K. J. Lin, "Service selection algorithms for Web services with end-to-end QoS constraints", Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service, Hong Kong, China, March (2005), pp. 129–136.
- [5] A. Parasuraman, V. A. Zeithaml and L. L. Berry, "A conceptual model of service quality and its implications for future research", the Journal of Marketing, (1985), pp. 41-50.
- [6] Z. A. S. Al-Khanjari, "Proposing a systematic approach to verify software requirements", Journal of Software Engineering and Applications, (2014).
- [7] C. Rolland, C. B. Achour, C. Cauvet, J. Ralyté, A. Sutcliffe, N. Maiden, M. Jarke, P. Haumer, K. Pohl, E. Dubois and P. Heymans, "A proposal for a scenario classification framework", Requirements Engineering, vol. 3, no. 1, (1998), pp. 23-47.
- [8] OASIS Group. Quality model for web services. Tech. rep., The Oasis Group. http://www. clip. dia. fi. upm. es/Projects/S-CUBE/papers/oasis05: WSQM-2.0. pdf, (2005).
- [9] A. Lupeikiene, J. Miliauskaite and A. Caplinskas, "A Model of View-Based Enterprise Business Service Quality Evaluation Framework", Informatica, Lith. Acad. Sci., vol. 24, no. 4, (2013), pp. 543-560.
- [10] T. Senivongse, N. Phacharintanakul, C. Ngamnitiporn and M. Tangtrongchit, "A capability granularity analysis on web service invocations", Proceedings of the world congress on Engineering and computer science, vol. 1, (2010), pp. 20-22.
- [11] X. Zhengyu, D. Baotian and W. Li, "Research of Service Granularity Base on SOA in Railway Information Sharing Platform", Proceedings of the 2009 International Symposium on Information Processing (ISIP" 09) Huangshan, PR China, (2009), pp. 391-395.

International Journal of Multimedia and Ubiquitous Engineering Vol.10, No.10 (2015)

- [12] E. Prud and A. Seaborne, "Sparql query language for rdf", (2008).
- [13] R. Fikes, P. Hayes and I. Horrocks, "OWL-QL—a language for deductive query answering on the Semantic Web", Web semantics: Science, services and agents on the World Wide Web, vol. 2, no. 1, (2004), pp. 19-29.
- [14] G. Antoniou and A. Bikakis, "Dr-prolog: A system for defeasible reasoning with rules and ontologies on the semantic web", IEEE Transactions on Knowledge and Data Engineering, vol. 19, no. 2, (2007), pp. 233-245.
- [15] Y. Jamoussi, M. Driss, J. M. Jézéquel and H. H. B. Ghézala, "QoS assurance for service-based applications using discrete-event simulation", International Journal of Computer Science, vol. 7, no. 6, (2010).
- [16] A. Leff, J. T. Rayfield and D. M. Dias, "Service-level agreements and commercial grids", IEEE Internet Computing, vol. 7, no. 4, (2003), pp. 44-50.
- [17] M. H. Zadeh and M. A. Seyyedi, "Qos monitoring for web services by time series forecasting", Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), vol. 5, (2010), pp. 659-663.
- [18] J. Cardoso, A. Sheth, J. Miller, J. Arnold and K. Kochutm, "Quality of service for workflows and web service processes", Web Semantics: Science, Services and Agents on the World Wide Web, vol. 1, no. 3, (2004), pp. 281-308.
- [19] H. A. Edelstein, "Introduction to data mining and knowledge discovery", (1998).
- [20] J. Friedman, T. Hastie and R. Tibshirani, "The elements of statistical learning", Springer, Berlin: Springer series in statistics, vol. 1, (2001).
- [21] M. Broy, "Service-oriented systems engineering: Modeling services and layered architectures", In Formal Techniques for Networked and Distributed Systems-FORTE 2003, Springer Berlin Heidelberg, (2003), pp. 48-61.
- [22] F. Bachmann and L. Bass, "Introduction to the attribute driven design method", Proceedings of the 23rd international conference on Software engineering, IEEE Computer Society, (2001), pp. 745-746.
- [23] J. Lee, "Editor, Service Life Cycle Tools and Technologies: Methods, Trends and Advances", IGI Global, (2011).
- [24] G. Huszerl and K. Kosmidis, "Object Oriented Notation for Modeling Quantitative Aspects", In Proc. Workshop of the International Conference on Architecture of Computing Systems (ARCS 2002), Karlsruhe, Germany, (2002), pp. 91-100.
- [25] S. Frølund and J. Koistinen, "Quality of Service Specification in Distributed Object Systems Design", In COOTS, vol. 98, (1998), pp. 1-1.
- [26] M. Makhlughian, S. M. Hashemi, Y. Rastegari and E. Pejman, "Web service selection based on ranking of qos using associative classification", International Journal On Web Service Computing (IJWSC), vol. 3, no. 1, (2012), pp. 1-14.
- [27] W. Abramowicz, R. Hofman, W. Suryn and D. Zyskowski, "SQuaRE based web services quality model", Proceedings of The International Multi Conference of Engineers and Computer Scientists, (2008), pp. 827-835.
- [28] C. Zhou, L.-T. Chia and B.-S. Lee. DAML-QoS ontology for web services. Proceedings of the IEEE International Conference on Web Services, pp. 472-479, (2004)
- [29] G. Damiano, E. Giallonardo and E. Zimeo, "On QoS-QL: a query language for QoS-based service selection and ranking", In Service-Oriented Computing-ICSOC 2007 Workshops, Springer Berlin Heidelberg, (2009), pp. 115-127.
- [30] S. Döweling, B. Schmidt and A. Göb, "A model for the design of interactive systems based on activity theory", Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work, (2012), pp. 539-548.
- [31] S. S. Selmi, N. Kraiem and H. B. Ghezala, "Toward a comprehension view of web engineering", Proceedings of the 5th international conference on Web engineering, Springer Berlin Heidelberg, (2005), pp. 19-29.
- [32] Z. Zheng, Y. Zhang and M. R. Lyu, "Distributed qos evaluation for real-world web services", Proceedings of the IEEE International Conference on Web Services (ICWS), (2010), pp. 83-90.
- [33] Z. Al-Khanjari, N. Al-Hosni, N. Kraiem and Y. Jamoussi, "Developing E-Government Interoperability Driven Methodology", Journal of Emerging Technologies in Web Intelligence, vol. 6, no. 3, (2014), pp. 318-323.

## Authors

**Jokha Al-Kalbani**, (MSc from Herriot-Watt University-UK) is currently a PhD student in the Department of Computer Science at Sultan Qaboos University (SQU) in Oman. She worked as Systems Analyst for more than 10 years at Hospital Information Systems Department in Sultan Qaboos University Hospital.



**tiem**, (HDR from University of Paris1 –Sorbonne ently Associate Professor of Computer Science at earch interests include IT adoption and usage delling, Software Engineering, Software product E tools. His research work has been supported by CNRS, INRIA, MRT (Ministry of Research and I Industry) and by the Commission of the European

Communities under the ESPRIT Programs (BUSINESS CLASS). He has had several articles published in many journals such as Management Science Information Systems Research Communications of the ACM and IEEE Transactions. He has been invited to present his research in many countries in North America, Europe, Africa and in the Middle East. Pr Naoufel KRAIEM has been member of over 20 program committees.



Zuhoor Al-Khanjari, (PhD from the University of Liverpool) is currently Associate Professor of Software Engineering in the Department of Computer Science at the Sultan Qaboos University (SQU). Her research interests include Software Engineering, Software Testing and Security, E-Learning. Her research also involves employing the Software Engineering strategies and methodologies to enhance the development of the e-Learning Content Management Systems. This research in e-Learning is more beneficial to Education community (SQU and other academic institutions across the country) particularly in encouraging the use of the efficient e-Learning strategies in future education. Currently, her research concentrates on developing a structured methodology for developing e-learning software components (e.g. Learning Objects) and e-Gov. Her research work has been supported by SQU funding. She is Chairing the SE Research Group in the department. She participates in the Evaluation Committees for IT international awards, the Editorial Boards of IT International Journals and Conferences, and other professional bodies and societies. She published 33 Journal and 40 conference papers in International journals and conference proceedings.



**Yassine Jamoussi**, (HDR between University of Manouba – University of Toulouse - France) is currently Assistant Professor of Computer Science at SQU. His research interests include Process-Centered Environment, Modeling and meta-modeling of flexible process, Process enactment, Modeling variability, Adaptation, Evaluation of process model, Service Oriented Architecture. His research work has been supported by several funding, such as CMCU (Europeen), INRIA (French), MRT (Ministry of Research and Technology and Industry of Tunisia), IG (SQU Oman). He has been invited to present his research in many countries in North America, Europe, Africa and in the Middle East. Dr Yassine Jamoussi has been conference chair of IEEE conferences and member of over 20 program committees. He published 15 Journal and 40 conference papers in International journals and conference proceedings.