Journal of King Saud University – Computer and Information Sciences (2014) **26**, 143–152

King Saud University

**Journal of King Saud University – Computer and Information Sciences**

www.ksu.edu.sa
www.sciencedirect.com

ORIGINAL ARTICLE

# Appraisal and analysis on various web service composition approaches based on QoS factors

**M. Rajeswari *, G. Sambasivam, N. Balaji, M.S. Saleem Basha, T. Vengattaraman, P. Dhavachelvan**

*Department of Computer Science, Pondicherry University, Puducherry, India*

Received 7 February 2013; revised 1 May 2013; accepted 22 August 2013
Available online 6 September 2013

**KEYWORDS**
Web services;
QoS modeling;
Analysis;
Service composition

**Abstract** Web services are the internet enabled applications for performing business needs, considered as the platform-independent and loosely coupled. Web service compositions build new services by organizing a set of existing services by providing reusability and interoperability. The research problem in web service composition is to obtain best effective services with the composition of services based on maximum quality of services (QoS) and satisfy the user's requirements. This study reveals various challenges in the QoS parameter for Web service composition because it is difficult to recognize. We have illustrated the related technology by analyzing QoS parameters based on existing algorithms with composition patterns and compared the results.


## 1. Introduction

Web services can be published and can be accessed with internet and business intranets for developing scattered applications. It can be defined as the model for software systems (Dhavachelvan et al., 2006) and designed to maintain interoperable communication and it can be published and can be invoked across the web (Papazoglou et al., 2004). The capacity to select and compose the inter-organizational and mixed services at runtime on the web is the important issue to be considered in web service applications. With the help of a single web service, we cannot satisfy both the functional and non-functional requirements of the user, so we need to mix a set of composite already existing web services to satisfy the user needs. The current research challenge is on web service composition of related web services based on its functional behavior and non-functional behavior. To supply languages and platforms which will allocate web services and making them to compose among dissimilar mixed systems. The research related to such languages are simple object access protocol (SOAP) (Box et al., 2001) when a web server calls the procedure which is running in another separate machine, they exchange their information with the help of XML formats. The use of SOAP provides interoperability and it allows the service to pass through HTTP.

* Corresponding author. Tel.: +91 9597975941.
E-mail addresses: raji.rajeswari18@gmail.com (M. Rajeswari), gsambu@gmail.com (G. Sambasivam), nbalajime1983@gmail.com (N. Balaji), m.s.saleembasha@gmail.com (M.S. Saleem Basha), vengattaraman.t@gmail.com (T. Vengattaraman), dhavachelvan@gmail.com (P. Dhavachelvan).
Peer review under responsibility of King Saud University.

ELSEVIER Production and hosting by Elsevier

1319-1578 © 2013 Production and hosting by Elsevier B.V. on behalf of King Saud University.
http://dx.doi.org/10.1016/j.jksuci.2013.08.003

Web services description language (WSDL) Chinnici et al., 2003 and universal description, discovery, and integration (UDDI) (T. Bellwood et al., 2005) are based on XML standards to identify the web service with the help of messages and operations as long as conceptual data are transmitted. Various added principles for service discovery, description and integration are the elements of ontology DAML-S (Martin et al., 2003) and for business process execution language for web service (BPEL4WS) provides web service composition based on WSDL (Andrews et al., 2003) were discussed. Web service composition is greatly difficult in terms of the number of available services which get increased, update at runtime based on up-to-date information and interoperability. Therefore building composite web service is critical. Web service composition provides flexibility for automatic integration and service reusability. It reuses the existing web services which are published by the service providers. Abstract web services are put forward in a language like BPEL, the service discovery uses UDDI to locate available services for the user request by functional matching.

Web service composition is the process of combining and reusing the existing web service to create a new business process according to user requirements. Web service provides self relating, interoperability, self-restricted and can be published, located and accessed over the internet through WSDL, UDDI and SOAP. Web service composition is used to attain interoperability and business to business interconnection of services from multiple business partners. Aggregation is the process in which an already existing web service was combined together based on certain definite rules. It allows reusability of web services, the rule states how to compose the service, the way of ordering and conditions for composing (Hull and Su, 2005). The reuse method is used to build new web service applications. These composition rules will explain how to compose all existing services into global services. Orchestration is the method in which already available web services were combined by the central coordinator that is orchestrator, which is responsible in invoking the service and merge these services in a particular way. It tells how web services can cooperate with everyone in message stage with execution stage and business sense. With this orchestration, the business models can interact with the private viewpoint of one of the business parties acting in the service process. In choreography process each component does its own complex task, then they used composition to combine overall composite services among the component as collaborative services. This choreography may associate with public messages, rules and agreements to process a single task which can be executed by a single component in the service model.

Service level agreement (SLA) provides maintenance of association between service provider and a service requester. Service requester will specify the terms and conditions about the services to the user (Dhavachelvan and Uma, 2005). SLA will tackle QoS measures as a term to achieve objectives of both service provider and service requester. QoS composition selects each task with different QoS functions and aggregates those values by satisfying the user's requirement. Based on heuristic value we can group all these methods into basic two classes: heuristics based approach and not-heuristic based approach to find the web services with the availability of multiple QoS constraints in service. The paper is organized as follows: Section II discusses the related works on web service composition. Section III explains about the QoS requirements and addresses various approaches for QoS based Web service composition and Section IV briefly describes various approaches on QoS factors. Finally, Section V concludes the study and gives the future research direction toward QoS calculation on service composition Ye et al. (2005).

## 2. Related work

Zeng et al. (2003) proposed the service community which consists of distinct service classes they used this community to differentiate the web service from others with same functionality but also with dissimilar QoS parameters. They proposed various quality determining processes to select a service with multiple QoS parameters such as cost, time, scalability, reliability, performance and security. Their selection is based on two approaches one is using a local selection for process and global selection of services. Aggarwal et al., 2004 proposed the framework based on service composition with ontology. They grouped the semantics into data, functional, non-functional or QoS and execution. This framework selects the best and identical service with the help of semantics.

Feng et al. (2007) consider multiple QoS constraints and found the utility based function for each QoS parameter. They have considered both the functional and non-functional behavior of the QoS parameters. This composition is highly complex because it has to select the best and worthy web service from multiple QoS and to fulfill user's requirements and outperforms with maximum of utility function. They had created a service class which consists of many candidate services with different non-functional parameters but with common functionality. They proposed an architecture which has composer to decompose user request and QoS analyzer to get QoS requirements from database and find the corresponding service from UDDI. They calculate service with multiple QoS constraints and compute the utility function. Karunamurthy et al. (2012) proposed an architecture which extends the standard web service models to explicitly support web service composition. The architecture is made up of three components namely: business process, a composition process and a description process. They validated by realizing the new interactions and developing prototypes of the composition framework and its techniques.

Yu and Lin (2005) proposed an approach for selecting the service as a problem. They have used many models like sequential, parallel and loop, their algorithm is based on sequential form by considering a single constraint. They proposed their problem as Multiple-Choice Knapsack Problem (MCKP), and they have used Dynamic programming to solve this problem and Pisinger algorithm to obtain a required solution. Then they used graph theory, services are modeled to a shortest path problem. They organized each service class as node and QoS parameter from nodes to their edges and constructed DAG using algorithms like Constrained Bellman–Ford (CBF), Constrained Shortest Path (CSP).

Yu and Lin (2005) proposed an approach is based on multiple QoS constraints. They have used dissimilar heuristic algorithm to tackle multiple constraint problems. They have designed a framework for web service composition. In this framework the QoS is evaluated and selects the web services which met user request. They considered time duration, execution rate, reputation, execution cost and availability as

QoS attributes. They proposed two approaches for service selection in the web composition, the first algorithm is local optimization and second algorithm is based on global planning.

Triantaphyllou et al. (1981), for local optimization, used the SAW method to calculate QoS for each web service in order to select the optimal service while in global optimization they handled with the help of the IP (Integer Programming) method to solve the selection problem. Though it is better than existing searching running time in IP is considerable for service selection and composition in runtime. They proposed backward context based service selection (BCCbSS). This algorithm runs each process and selects the services step by step for each process. Following selecting services, the algorithm went reverse and checked the already selected services for composition. However this algorithm is suitable only for sequential model and it cannot produce any optimal composition. Wu et al. (2007) specify the entire possible grouping of web services for each task with optimal QoS value. Their model satisfies all global constraints and they achieved better QoS with higher execution duration. But the time complexity of this approach is high about O (pq) where p and q are the maximum number of services and number of tasks. Zeng et al. (2004) proposed IP for solving MCDM. They proposed Ag Flow, QoS aware middleware for web service composition. They evaluated the overall quality of a web service and achieved better QoS, lower execution price and duration. But there is no guarantee to satisfy global QoS constraints.

## 3. QOS requirements for web service composition

Quality of service (QoS) is a non-functional aspect. It represents the various non-functional parameters like total time taken for service to execute, the cost, availability and security features etc. Web service Composition is a compilation of several services aggregated to execute in a sequence form. A complex composition problem that is parallel implementation, branching, and loops is usually an NP-hard problem. For selecting a service we have a set of candidate services $S_p$, where $p \in [1\ldots n]$ and the task $t_q$ where $q \in [1\ldots m]$. Depending on the user need cost, availability, security, response time, throughput is selected.

i. *Cost*: The cost quality is the amount that a service requester needs to pay to execute a service using task (Cardoso et al., 2004; Liu et al., 2004). The cost factor $C_{pq}$ can be calculated by the amount required to pay for a service $p$ using task $q$ which is:

$$C_{pq}, p \in [1..n], q \in [1..m] \tag{1}$$

The value of $C_{pq}$ is not determined when the service $p$ cannot execute the task.

ii. *Time*: The time quality measures the execution time between the requests sent and results received. It is the length of time for a service to provide a response to various types of requests from composite users (Chandrasekaran et al., 2003). The time quality $T_{pq}$ can be calculated by,

$$T_{pq}, p \in [1..n], q \in [1..m] \tag{2}$$

iii. *Reliability*: Reliability is the capability of upholding the service and service quality. Reliability represents the ability of a service to function correctly and consistently. It is commonly measured in terms of transaction failures per year or month (Chandrasekaran et al., 2003; Cardoso et al., 2004; Jin et al., 2006).

$$\text{Reliability}_{pq} = e^{-\int_0^{time_{pq}}}\gamma^{(t)dt} * e^{-\gamma * time_{pq}} \tag{3}$$

where $\gamma(t)$ is the failure rate for the service.

iv. *Availability*: Availability is the presence of a web service to be connected to for a client. It is the absence of service downtime and represents the probability that service is available. Availability is associated with time-to-repair (TTR). TTR is the time it takes to repair a service which has failed. Preferably smaller values of TTR are desirable to achieve high availability Youngdahl and Kellogg (1997). Availability factor $A_{pq}$ can be calculated by the services which are responded to the total number of services as

$$A_{pq} = \frac{\text{respond}_{pq}}{\text{total request}_{pq}} \text{ where total request}_{pq} \neq 0 \tag{4}$$

v. *Accessibility*: Accessibility is the capability to satisfy a web service request. It represents the scale in which a service request is served. It is a measure denoting the achievable speed of a service in time. The service is available for a large number of clients at a greater extent of accessibility (Mani and Nagarajan, 2005).

$$\text{Accessibility} = \frac{\text{No of Acknowledged messages}}{\text{No of Requested messages}} \tag{5}$$

vi. *Performance*: Performance can be measured in terms of throughput and latency. Throughput is the number of user service requests served in a given period of time. Latency is the length of time between getting a reply and transferring a request. A good performance would achieve better throughput and lesser latency (Mani and Nagarajan, 2005; Papazoglou et al., 2004).

$$Performance = \text{Response Time} + \text{Latency} \tag{6}$$

vii. *Security*: Security involves authentication, authorization, confidentiality, and access control and message integrity. It is significantly characteristic because web service incantation happens over the internet. The security aspects that a service requires are described in the SLA and service providers should maintain it to obtain the level of security (Ran et al., 2003; Jin et al., 2006). Security can be calculated as

$$\text{Security} = \max\{\text{security}(p,q)\}, \text{ where } p \in [1..n], q \in [1..m] \tag{7}$$

viii. *Reputation*: The reputation of an execution plan is the average of the reputations of the services that take part in the plan. It measures the service trustworthiness based on the user experience on using the service. Reputation can be calculated by the ranking $r$ given for the service by the users. Here for $N$ times of service, $F_r$ is $r$th rank given as:

$$\text{Reputation} = \frac{\sum_{r=1}^{N} F_r}{N} \tag{8}$$

ix. *Transaction*: The transaction is related to the process performed on ACID property which contains the characteristics like atomicity, consistency, isolation and durability. A particular web service needs transactional behavior and it is accompanied in SLA (Mani and Nagarajan, 2005; Ran et al., 2003).

x. *Integrity*: Integrity is the measure in which a web service performs its process with respect to WSDL description and SLA. Providing superior level integrity is always nearer to its WSDL description. It refers to the maintaining of reliable and correct interaction for a service.

There are fundamental QoS requirements that service providers must think about whenever they develop their web service applications for their corresponding service requester. These are speedy, readiness, security and reliability as illustrated in Nurhayati et al. (2008). Readiness parameter includes certain parameters like availability and accessibility to provide guarantees for the requester. Reliability process is used to describe the wholeness and sturdiness. Speedy process tells how quick the providers will deliver their service. The security process is used to describe authentication of identity, authorization for access control, confidentiality for privacy and non-repudiation for confirmation of services Al Masri and Mahmoud (2007).

QoS management of services is used for the distributed services (Victer Paul et al., 2012; Wang et al., 2004) requiring both the QoS characteristics and parameter. In SOA architecture, we have different service providers and service requestors having distinct QoS requirements in terms of their service reliability, time, security and performance (Victer Paul et al., 2013). For example some service requestors may be more important than others and some providers will have greater priority than with correct ordering in faster response time (Fourar-Laidi, 2013).

The QoS calculation is always necessary and can be used for a requester to get distinct service; we should distinguish and recognize the QoS requirements to attain high quality of web services. Nurhayati et al. (2008), provided an assurance for service applications as follows,

- Route service providers in providing and publishing their web service applications in registry to increase availability and accessibility.
- Users trust their web service applications with the help of reliable transactions.
- Allow requesters to select their services from service providers.
- Service providers should reply to requesters with less time.
- Provide assurance for the data through security and complete transactions for providers and requesters.

QoS has many advantages over service composition: We can translate our vision into business processes. Allow for the best selection of service to satisfy the user's requirements. Monitor the web process based on QoS scale. It can evaluate strategies and methodology.

Web service composition architecture, shown in Fig. 1, consists of service requester, service composer, third party provider, QoS analyzer, QoS processing, public registry, private registry and database for storing parameters. Composer does matching of service request with provider based on QoS parameters. Third party providers are the authorized providers who fulfill the requester based on the SLA. All the services from third party providers are stored in private registry. Initially the third party provider will put the services in private registry, the composer will register about the service to its public registry so when the service provider finds the services automatically the composer will locate the service in private registry. The Bind operation takes place between composer and third party provider, then to service requester and composer. Now the service requester will provide the QoS request to QoS analyzer, the metric calculation is done by QoS processing and it is stored in the database (Senthilkumaran and Sankaranarayanan, 2013).

When the requester queries for the service if the searched service is not present in public registry then the composer will invoke a command to third parties and analyze the QoS parameters of service and get service from private registry and send this service to public registry. By using the utility function they get maximized QoS service from private registry and send it to the requester with the help of the composer. All the parameters like performance, reliability, availability and security, etc. were stored in database and they are processed with the help of the QoS processing model Jaeger et al. (2005).

## 4. Analysis on QoS based web service composition techniques

There has been a significant research work on service composition and its approach based on QoS. The concept of QoS has been widely used in many networks and middleware. The work for this approach focuses on the performance of tasks and network. There have been many adoption QoS concepts with their fast growth. The criteria for ranking are non-functional based on QoS metrics.

### *4.1. QoS aware middleware by Liangzhao Zeng et al.*

They proposed a middleware which is used to select web services and satisfy user requirements over QoS (Zeng et al., 2004). They used two selection approaches like task oriented and global oriented. They estimated by using a travel planning and they retrieved by service execution engine in various dimensions. They evaluated QoS of composite service both in static and dynamic environments (Vengattaraman et al., 2011). Their distinct web service ontology identified composite services using ontology's, model describing the cost of the composite services with a task in a dynamic environment and obtained higher computation in dynamic than static process. They represented the aggregation function for various QoS of composite services CS by using plans $p$, operation op ($t$) of a services with $t$.

### *4.2. QoS reference vector of BangYu Wu et al.*

The service selection and tradeoff based on QoS vector for component service and composite service were proposed in Wu et al. (2007). The service selection was based on Reference vector (CRV). They used two algorithms one is to get accurate composite services and the other is to compose a service with QoS. They achieved the fastest speed and highest execution duration.

By using a QoS reference vector generator of composite service and the selection mechanism for the composite service

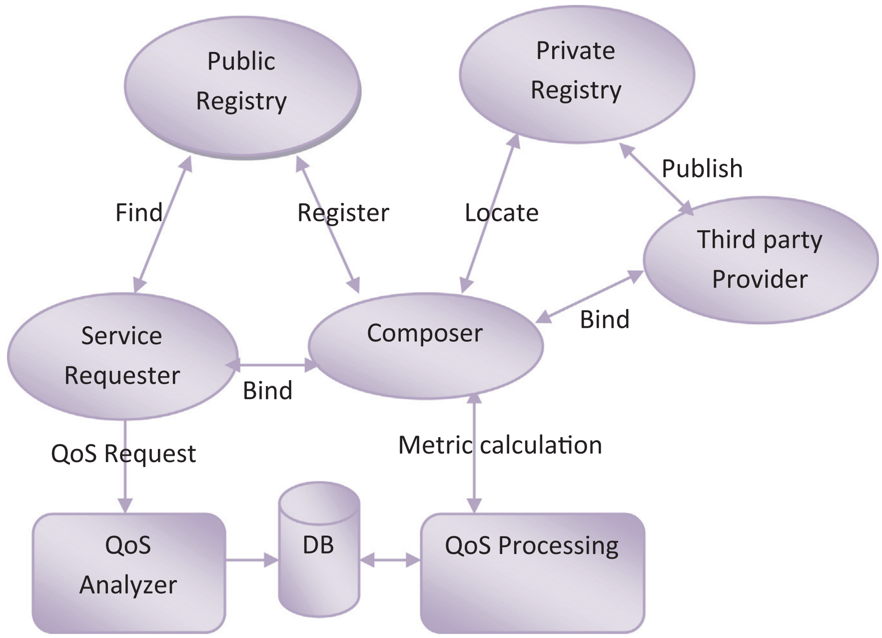


**Figure 1** Web service composition model based on QoS.

were made. For the generation of service, the candidate services are placed in the matrix of ($m$, $p$)

The selection mechanism can be done by

- Providing a composite QoS service.
- Stabilizing matrix based on QoS.
- Executing service selection based on the CRV.

In their simulation, they got the approximate solution for composite services not the total minimizations. In their experiment they used three parameters, the $x$-axis represents the selected parameter and $y$-axis represents selection cost. They proved that selection cost increased with an increase in parameters. The time complexity can be referred with the help of cost using exhaustive search for the optimization. The complexity for the cost selection is $O\ (N^*)$ than $O\ (N)$.

### *4.3. BCCbSS approach by Hong Qing Yu et al.*

A concept of composition together with service selection algorithm was proposed in Yu and Reiff-Marganiec (2009). They developed a backwards composition context based service selection approach (BCCbSS). It always goes reverse single step to ensure the best work and invoke the selected service. Their algorithm is more fault tolerant and scalable. They evaluated the process with test scenarios and proved the services are dynamic bound and invoked at run-time at design time. This algorithm provides selection and invocation of service composition in a step by step process by collecting services from the template. As there are more services it gives efficiency and scalability. This selection mechanism gets knowledge from existing services by providing run-time search and fault tolerant. BCCbSS approach increases the number of services available and duration for each step remains fixed. They evaluated the services in each step, this algorithm works well and provides scalability, when the workflow increases a gradual increase in run-time is achieved.

Here the global optimization becomes complex and not suitable for the dynamic environment of service selection. So finding a service by this composition will reduce the complexity and it requires efficiency.

### *4.4. MCDM by DongmeiLiu et al.*

To optimize the service selection using QoS constraints, Multiple Criteria Decision Making (MCDM) has been used to merge multiple properties of resources. They divided service selection into two classes; mathematical models are assigned to each class (Liu, 2009). The heuristics selection algorithms are used to solve the models. They used web service design under multiple QoS; they formulate every QoS attribute to $Q_p$ with weight $w_q$. They calculated for each task using SFC and obtained maximum value. According to cost, time and availability the QoS components are varied. The composite services are redesigned whenever the global planning of IP approach is used. They proposed MCDM algorithm in dynamic web service composition using heuristics. They represented four services with the QoS scores and resource scores are marked. Their graph shows that MCDM algorithm is time efficient and scalable.

The service selection by local optimization will compute a QoS score for candidate services and select the best service which has the highest score. The worst case of complexity for $k$ tasks is $o(nk)$. So we went to Global Planning, will find all the eligible combinations of services and then compute QoS scores and select the highest score. This algorithm will yield $o(k^n)$ as its time complexity. For its worst case the time complexity will increase with the given problem.

### *4.5. Multiple QoS attributes by PengChengXiong et al.*

Using the Petri net model they planned a structure for composition on user's preference to choose whether functional or non-functional requirements (Xiong et al., 2009). By using service module architecture they configured web services under multiple QoS objectives (Venkatesan et al., 2005). They obtained the maximum value for all services by using MCDM. This architecture provides a programming model for creating and accumulating the business task in SOA and it deals with both functional and non-functional constraints. Orchestration model is used for a task allocator to receive and assign the task in the interface.

### 4.6. Ant inspired technique by Cristina Bianca Pop et al.

A technique for web service composition by ants with the ant colony optimization was discussed in Chifu et al. (2010). For selection criteria they considered QoS attributes between the services. This technique combines service composition graph model with heuristic and proposed composition graph, ant based selection. They took $S_1$ and $S_2$ as the current service and candidate services, DoM is the degree of matching between the services, $W_{\text{Match}}$ and $W_{\text{QoS}}$ are the corresponding weights. The QoS score has been calculated with $w_i$ which is weight of the QoS attribute, $\text{Attr}_i$ (s) is the $i$th QoS attribute and $n$ is the number of QoS attributes considered. This model considers the cost, runtime, reliability and availability. The quality of a composition has been estimated with the number of services involved in the composition. To improve the efficiency and accuracy of services they organized available services into a cluster group.

For the composition algorithm Canfora et al. (2005), the discovery phase has a maximum runtime of $O(N^2)$, for each $N$ number of clusters. Generally pruning algorithm takes complexity of $O(|V| + |E|)$ with $V$ set of vertices and $E$ set of edges, by considering this pruning algorithm the overall time complexity of this composition algorithm takes $O(N^3)$.

### 4.7. An architecture by Rajesh Karunamurthy et al.

A novel architecture for web service composition to build the business model was discussed in Karunamurthy et al. (2012). This architecture derives the composition is based on matching approach, categorizing approach and assembly approach to understand the communications (Chentouf and Khoumsi, 2013) and arche type by using the Isabelle theorem prover and Pellet reasoner. These manipulations put together MSC, SAWSDL and NFSL. Experiments were performed among prototypes and permit composing web services with all individuality in lesser response time. From this, Pellet is easy and provides interfaces to semantic reasoning. This architecture provides functional and semantic characteristics.

The composition graph model summarizes the set of composition for Meta heuristics. The main objective of this method is to obtain a graph for all the possible compositions which satisfies the user's request. In this architecture they added special entity web service composition registry for an authorized source of information. They introduced six more invocations by including standard find, bind and publish. Third party web service provider will provide the primitive services to the requester. The third party publishes their primitive web services in the registry. The composer gets notifications about user request in the registry. When the requester finds the service which is not in the registry, the composer is notified about the composition. The composer will provide this primitive service to a web service registry by getting detailed information from the requester. Binding of services between requester and provider takes place (Venkatesan et al., 2013).

### 4.8. Web service composition with complex structures by HuiyuanZheng et al.

The QoS calculation for composite services by allowing for probability and order was proposed in Zheng et al. (2012). They presented a QoS analysis for composite service. The calculation is based on the preset QoS. The cost of the QoS is reduced by the particular structure for web service composition. The QoS requirement and discovery can be a procedure by itself.

Where $N_p$ is the number of execution paths in the service composition. The QoS may vary from web service to web service, most of the web service providers are not able to concern about the level of QoS provided to their requestors. The QoS calculation for response time and latency is same as that for execution time. By their experiment, they showed the changes in the QoS of the service composition. Their result proved that there is a decrease in latency and increase in response time with a guarantee.

The time complexity of this proposed algorithm was $O(|V| + |A|)$ where $V$ is the number of vertices in the service graph and $A$ is the number arcs, for detecting loops with DFS algorithm (Hassan, 2013). The service graph can be changed into a tree with QoS metrics by the DFS algorithm for handling MEME (multiple entry multiple exit).

### 4.9. Hybrid approach by Mohammad Alrifai et al.

They proposed an explanation to overcome performance issues in QoS composition and this algorithm combines both the global optimization with mixed integer programming and the local selection to find the best web service based on QoS metrics (Alrifai, 2012). This composition for candidate services has satisfied the user's back-to-back QoS requirements. The QoS calculation for some composite services is focused. The complex structures include loop, sequential, conditional and parallel and obtained the aggregation function for each metric. Through this hybrid approach they obtained in performance analysis the computation time and provided with optimal increases with the increase in services and achieved close to optimal results (Rikli and Almogari, 2013). Their hybrid approach shows much scalable because the calculation time increases much slower than the other two methods.

The time complexity of heuristic approach is polynomial whereas the time complexity of knapsack problem is exponential. The time complexity of this approach may be affected by the size of a service, candidate service and service class. It is independent of the number of available services which provides more scalable and better complexity than the global planning approach Jinzhong et al. (2010).

There has been a challenging research work based on QoS metrics for Web service composition. This Table 1 examines on diverse metrics based on various algorithms as proposed by the authors. Each author used different algorithms and approachs to resolve those parameters. Based on Table 1 we have analyzed these metrics with algorithm and designed Table 2.

**Table 1** Analysis on various approaches for QoS based WSC.

| S. No. | Authors & year of publication | Title of the paper | Theme of the paper | Advantages | Disadvantages | Future work |
|---|---|---|---|---|---|---|
| 1 | Liangzhao Zeng et al. (2004) | QoS-aware middleware for Web services composition (Zeng et al., 2004) | IP Searching provided is better than Exhaustive search. Ag flow support quality driven Web service compositions | Better QoS, lower execution price, performance and execution duration | No guarantee to satisfy global QoS constraints. Computation cost is not considered | Combine global planning and local optimization approaches in order to leverage their relative advantages |
| 2 | BangYu Wu et al. (2007) | Service selection model based on QoS reference vector (Wu and B., 2007) | They specify all services and obtained composite service with optimal QoS value | QoS metrics are not very large. | The time complexity of this approach is high | Knowledge record to crumble user's applications can be constructed |
| 3 | Hong Qing Yu and Reiff-Marganiec (2009) | A backwards composition context based service selection approach for service composition (Yu and Reiff-Marganiec, 2009) | This algorithm walks off through the process and selects services step by step. After selecting services the algorithm goes back and checks for selected services | Fault tolerant, scales well for larger number of services | This algorithm works only for sequential model | Investigate on completeness and Comparing approaches to other planning approach |
| 4 | Dongmei Liu (2009) | A heuristic QoS-aware service selection approach to web service composition (Liu, 2009) | It supports late binding of web services at runtime. Their algorithm based on sequential model and several QoS attributes | Provide QoS based service selection in dynamic web service composition | Each time when a vector is updated, QoS score will rise | They extended with QoS attributes |
| 5 | Peng Cheng Xiong et al. (2009) | Web service configuration under multiple quality-of-service attributes (Xiong et al., 2009) | Web services are formulated based on the functional configuration by Petri nets. Graph structure and algebraic properties are analyzed | It supports the specification of functional requirement and QoS expectations | Concurrency and synchronization. It does not model the time sequences | Deal with business process sequencing and runtime related QoS |
| 6 | Cristina Bianca Pop et al. (2010) | Ant inspired technique for automatic web service composition and selection (Chifu et al., 2010) | Merge a service composition and ant colony optimization For building the composition graph, ant based selection and finding the solution | ACO meta heuristics which is efficient for optimization problems | The computational complexity of the algorithms is in the order of N3 | The current model considers basic QoS. It may be extended with other QoS attributes |
| 7 | Rajesh Karunamurthy et al. (2012) | The novel architecture for web service composition (Karunamurthy et al., 2012) | It extends the standard web service business models and validates the proposed architecture by realizing the new interactions and developing prototypes | Provides the description of the functional behavioral, non-functional and semantic characteristics of web services | The extended business model for web service composition introduces many new interactions | From the business model perspective, the realization scheme for putting and locating interactions could be developed |
| 8 | Huiyuan Zheng et al. (2012) | QoS analysis for web service compositions with complex structures (Zheng et al., 2012) | Calculate QoS for composite services with complex structures by probability and conditions of each execution path | It can generate richer QoS information with Probability. Unstructured conditional patterns with (SEME) | Unstructured loop patterns were not considered with the fixed assigned task | Estimate the probability distributions for services with less frequency use or short life cycle |
| 9 | Mohammad Alrifai (2012) | A hybrid approach for efficient web service composition with end-to-end QoS constraints (Alrifai, 2012) | Method for selecting request from a collection of service candidates with user's continuous QoS requirements | Cost of QoS is condensed by special structure. Web services can join and leave service classes at any | They did not consider number of QoS constraints | The QoS requirement and discovery can be obtained in the preprocessing phase itself |

**Table 2** Assessments on various web service composition algorithms based on QoS metrics.

| Author name | Cost | Execution time | Reliability | Reputation | Availability | Throughput | Trust | Security |
|---|---|---|---|---|---|---|---|---|
| Liangzhao Zeng et al. (2004) | ✓ | ✓ | – | ✓ | ✓ | – | ✓ | - |
| Bang Yu Wu and B. (2007) | ✓ | ✓ | ✓ | – | – | – | ✓ | ✓ |
| Hong Qing Yu and Reiff-Marganiec (2009) | ✓ | ✓ | – | – | – | – | – | – |
| Dongmei Liu (2009) | ✓ | ✓ | – | – | – | – | – | – |
| Xiong et al. (2009) | ✓ | – | ✓ | – | ✓ | – | – | – |
| Cristina Bianca Pop et al. Chifu et al. (2010) | ✓ | ✓ | ✓ | – | ✓ | ✓ | – | ✓ |
| Karunamurthy et al. (2012) | ✓ | ✓ | ✓ | ✓ | ✓ | – | – | ✓ |
| Huiyuan Zheng et al. (2012) | ✓ | ✓ | – | – | ✓ | – | – | – |
| Mohammad Alrifai et al. Mani and Nagarajan (2005) | ✓ | ✓ | – | – | ✓ | ✓ | – | – |

Various research papers consider around 10 web services and mostly they focus on QoS parameters like execution cost, execution time and availability as important factors. In Figs. 2–4 we have shown the comparison in graph with respect to the QoS parameters like execution cost, execution time and availability according to the author's algorithm. Most of the authors kept "high" or "low" as value for availability. The analysis of this graph shows variations in number services to QoS parameters with respect to various research algorithms described in Table 1. Each author used various algorithms as described in Table 1 and from the analysis of this graph, we can use more number of services and include other QoS parameters like reliability, reputation, security and throughput.

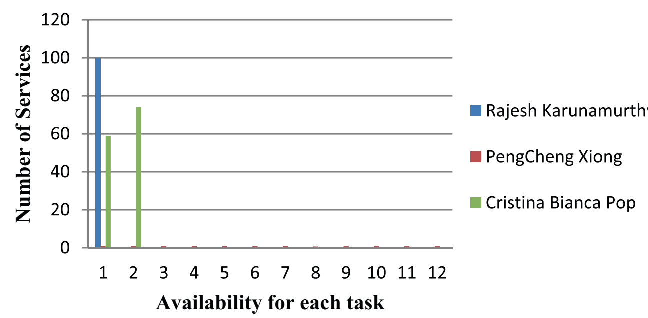


**Figure 4** Variation in availability for different web service composition searches.

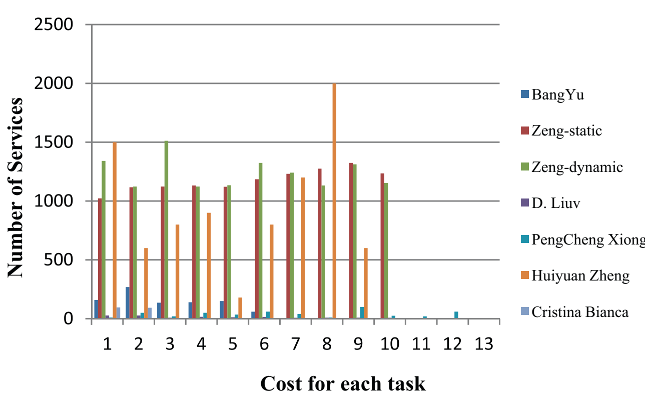


**Figure 2** Variation in the cost for different web service composition searches.

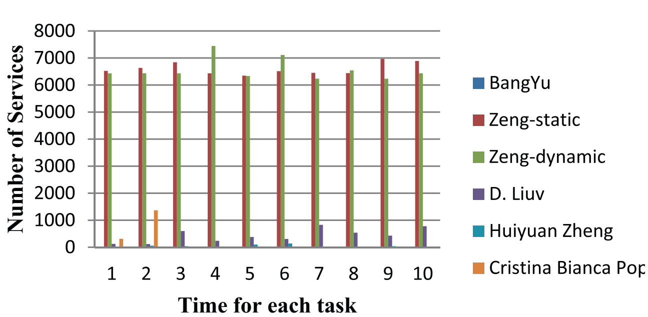


**Figure 3** Variation in execution time for different web service composition searches.

Fig. 2 depicts the variation in the cost metric for various web service composition techniques. Based on the QoS metric cost quality as a factor, we have compared various algorithms proposed for web service composition. The graph shows that the cost for each task varies with algorithms, it varies in cost quality from 1 to 12 dollars and set of candidate services from 500 to 2500 services.

In Fig. 3, we have compared various algorithms proposed by authors by considering the QoS metric time quality as a factor. The graph shows that the time for each task varies with algorithms, it varies in time quality from 1 to 6 s and set of candidate services from 1000 to 7500 services.

In Fig. 4, we have compared various algorithms proposed by authors by considering the QoS metric availability quality as a factor. The graph shows that the availability for each task varies with algorithms, it varies in the availability of services from 1 to 6 h and set of candidate services from 10 to 100 services.

## 5. Discussion

Table 1 summarizes the composition approaches described in the previous section according to QoS factor within a composition. From our analysis, we can derive some perceptive about composing services based on QoS which can be a challenging task and requires runtime and sequencing with less frequency reuse. These approaches allow composition of QoS aware services to satisfy performance, security, availability and reliability. QoS based composition of service and providing a utility function to evaluate all the QoS parameters for each service.

This literature contains a large work on QoS for composite services and the hybrid approach to combine global and local selection of composite services and extend the architecture with third party business model. We have discussed many related works which use composite services using composition patterns like sequential, parallel, loop and conditional patterns.

## 6. Conclusions

In this paper, we have analyzed and compared various composition approaches. We claim that service composition plays a major role in interoperability for business partners. This research work focuses on various QoS models and highlights the challenges for QoS in Web service composition as it is necessary for providing a contract between service providers and requesters. Therefore, service providers should provide high-quality web service with QoS requirements. We have defined QoS and its role in web service composition. The QoS based service selection for composite services and to select a service for each task from its service group. Web services are composed based on QoS metrics by evaluating the utility function and thus maximizing the overall QoS using the hybrid approach in composition patterns. As a result, we can estimate QoS for composite services for each execution based on composition patterns. Our future work will focus on the QoS trade-off at runtime by providing scalability and integration with third party business model.

## Acknowledgements

This work is a part of the Research Projects sponsored under the Major Project Scheme, UGC, India, Reference Nos: F.No.41-619/2012 (SR), dated 01 July 2012 and F. No. 40-258/2011(SR), dated 29 June 2011. The authors would like to express their thanks for the financial supports offered by the Sponsored Agency.

## References

Aggarwal, R., Verma, K., Miller, J., et al., "Constraint driven Web service composition in METEOR-S", Proceeding of IEEE International Conference on Service Computing (SCC'04), Shanghai, China, Sep. 2004.

Al Masri, E., Mahmoud, Q.H., 2007. QoS-based discovery and ranking of web services, In: Proc. Int'l Conf. Computer Comm. and Networks (ICCCN '07), pp. 529–534.

Alrifai, Mohammad, Risse, Thomas, Nejdl, Wolfgang, 2012. A hybrid approach for efficient web service composition with end-to-end QoS constraints, Journal ACM Transactions on the Web (TWEB) Web 6, 2, Article 7, May 2012.

Andrews, T., et al., 2003. Business process execution language for web services (BPEL4WS) 1.1. Online: <http://www-106.ibm.com/developerworks/webservices/library/ws-bpel>, May 2003.

Bellwood, T., et al., 2005. Universal description, discovery and integration specification (UDDI) 3.0. Online: <http://www.uddi.org/pubs/uddi-v3.00-published-20020719.html>.

Box, D., et al., 2001. Simple object access protocol (SOAP) 1.1. Online: <http://www.w3.org/TR/SOAP/>.

Canfora, G., Penta, M.D., Esposito, R., Villani, M.L., 2005. An approach for qos-aware service composition based on genetic algorithms. In: GECCO'05: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, New York, NY, USA. ACM pp. 1069–1075.

Cardoso, J., Sheth, A., Miller, J., Arnold, J., Kochut, 2004. Quality of service for workflows and web service processes, <http://scholar.google.com/url?sa> = U&q = http://chief.cs.uga.edu/~jam/home/theses/cardoso_dissert/CSM%2BQoS-WebSemantics.

Chandrasekaran, S., Miller, J.A., Silver, G.S., Arpinar, B., Sheth, A.P., 2003. Performance analysis and simulation of composite web services. Electronic Markets: The International Journal of Electronic Commerce and Business Media 13 (2), 120–132.

Chentouf, Zohair, Khoumsi, Ahmed, 2013. A high abstraction level approach for detecting feature interactions between telecommunication services". Journal of King Saud University-Computer and Information Sciences. Elsevier, Vol. 25, pp. 99–115.

Chifu, V.R., Salomie, I., Dinsoreanu, M., David, T., Acretoaie, V., 2010. Ant-inspired technique for automatic web service composition and selection, In: 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing-2010.

Chinnici, R., et al., 2003. Web services description language (WSDL) 1.2. Online: <http://www.w3.org/TR/wsdl/>.

Dhavachelvan, P., Uma, G.V., 2005. Multi-agent based framework for intra-class testing of object-oriented software. International Journal on Applied Soft Computing. Elsevier, pp. 205–222, Vol. 5, No. 2.

Dhavachelvan, P., Uma, G.V., Venkatachalapathy, V.S.K., 2006. A new approach in development of distributed framework for automated software testing using agents, In: International Journal on Knowledge-Based Systems, Elsevier, Vol. 19, No. 4, pp. 235–247, August 2006.

Feng, Xingzhi, Ren, Yi, Hu, Jianqiang, Wu, Quanyuan, Jia, Yan, 2007. A model for service composition with multiple QoS constraints, In: Proceedings of the International Conference on Computing: Theory and Applications (ICCTA'07).

Fourar-Laidi, Hakim, 2013. A smart card based framework for securing e-business transactions in distributed systems. Journal of King Saud University-Computer and Information Sciences. Elsevier, Vol. 25, pp. 1–5..

Hassan, M.R., 2013. An efficient method to solve least-cost minimum spanning tree (LC-MST) problem. Journal of King Saud University-Computer and Information Sciences. springer, Vol. 24, pp. 101–105.

Hull, R., Su, J., 2005. Tools for composite web services: a short overview. SIGMOD Rec. 34, 86–95.

Jaeger, M.C., Mühl, G., Golze, S., 2005. QoS-aware composition of web services: a look at selection algorithms. In: International Conference of Web Services (ICWS'2005), July 11–17, Orlando.

Jin, L.J., Machiraju, V., Sahai, A., 2006. Analysis on service level agreement of web services, Hewlett Packard. Retrieved October 5, 2006, from http://athena.union.edu/~hemmendd/Gradseminar/hpl.pdf.

Jinzhong, Li, Jiewu, Xia, Weidong, Tang, 2010. A review of QoS-based web service selection algorithms. Appl. Res. Comput. 27 (10), 3622–3627.

Karunamurthy, Rajesh, Khendek, Ferhat, Glitho, Roch H., 2012. The novel architecture for web services composition. J. Netw. Comput. Appl. 35.

Hwang CL and Yoon K, Multiple Criteria Decision Making Lecture Notes in Economics and Mathematical Systems. Springer-Verlag, 1981.

Dongmei Liu, Zhiqing Shao, Caizhu Yu1, Guisheng Fan, 2009. A heuristic QoS-aware service selection approach to web service composition, In: IEEE/ACIS International Conference on Computer and Information Science.

Liu, Y., Ngu, A.H.H., Zeng, L., 2004. QoS computation and policing in dynamic web service selection. In: Proc. Int'l Conf. World Wide Web (WWW) (Alternate Track Papers & Posters), pp. 66–73.

Mani, A., Nagarajan, A., 2005. Understanding quality of services for web services. Retrieved July 1, 2005. from: <http://www.ibm.com/developerworks/library/wsquality.html>.

Martin, D., et al., 2003. DAML-S (and OWL-S) 0.9 draft release. Online: <http://www.daml.org/services/daml-s/0.9/>, May 2003.

Nurhayati, Wan, Rahman, Wan A.B., Meziane, DrFarid, 2008. Challenges to describe QoS requirements for web services quality prediction to support web services interoperability in electronic commerce. Commun. IBIMA 4.

Papazoglou, Michael P., Dubray, Jean-jacques, 2004. A survey of web service technologies.

Ran, S., 2003. A model for web services discovery with QoS, ACM Inc., 4(1), 2003, pp. 1–10, ISSN 1551 9031.

Rikli, Nasser-Eddine, Almogari, Saad, 2013. Efficient priority schemes for the provision of end-to-end quality of service for multimedia traffic over MPLS VPN networks. Journal of King Saud University-Computer and Information Sciences. Elsevier, Vol. 25, pp. 89–98.

Senthilkumaran, T., Sankaranarayanan, V., 2013. Dynamic congestion detection and control routing in ad hoc networks. Journal of King Saud University-Computer and Information Sciences. Elsevier, Vol. 25, pp. 25–34.

Vengattaraman, T., Abiramy, S., Dhavachelvan, P., Baskaran, R., 2011. An application perspective evaluation of multi-agent system in versatile environments. International Journal on Expert Systems with Applications, Elsevier 38 (3), 1405–1416.

Venkatesan, S., Dhavachelvan, P., Chellapan, C., 2005. Performance analysis of mobile agent failure recovery in e-service applications, In: International Journal of Computer Standards and Interfaces, Elsevier, Vol-32, No.1–2, pp. 38 43. ISSN: 0920–5489.

Venkatesan, S., Saleem Basha, M.S., Chellappan, C., Vaish, Anurika, Dhavachelvan, P., 2013. Analysis of accounting models for the detection of duplicate requests in web services. Journal of King Saud University-Computer and Information Sciences. springer, Vol. 25, pp. 7–24..

Victer Paul, P., Saravanan, N., Jayakumar, S.K.V., Dhavachelvan, P., Baskaran, R., 2012. QoS enhancements for global replication management in peer to peer networks. Future Generation Computer Systems, Elsevier 28 (3), 573–582.

Victer Paul, P., Rajaguru, D., Saravanan, N., Baskaran, R., Dhavachelvan, P., 2013. Efficient service cache management in mobile P2P networks. Future Generation Computer Systems, Elsevier 29 (6), 1505–1521.

Wang, G., Chen, A., Wang, C., Fung, C., Uczekaj, S., 2004. Integrated quality of service (QoS) management in service-oriented enterprise architecture, In: Proceedings of the 8th IEEE Enterprise Distributed Object Computing Conference, September 2004, pp. 21–32.

Wu, B., Chi, C., Xu, S., 2007. Service selection model based on QoS reference vector, In: Proceedings of IEEE Congress on Services, 2007.

Xiong, Peng Cheng, Fan, Yu Shun, Zhou, Meng Chu, 2009. Web service configuration under multiple quality-of-service attributes. IEEE Transactions on Automation Science and Engineering 6 (2).

Ye, N., Gel, E.S., Li, X., Farley, T., Lai, Y.C., 2005. Web server QoS models: applying scheduling rules from production planning. Computers & Operations Research 32, 1147–1164.

Youngdahl, W.E., Kellogg, D.L., 1997. Relationship between service customers' quality assurance behaviors, satisfaction, and effort: a cost of quality perspective. J. Operations Manag. 15 (1), 19–32.

Yu, T., Lin, K.J., 2005. Service selection algorithms for web services with end-to-end QoS constraints. J. Inf. Systems E-Business Manag. 3 (2), Springer.

Yu, T., Lin, K.J., 2005. A broker-based framework for QoS-aware web service composition, e-Technology, e-Commerce and e-Service, In: Proceedings EEE'05, pp. 22–29, April 2005.

Yu, H.Q., Reiff-Marganiec, S., 2009. A backwards composition context based service selection approach for service composition, services computing SCC '09. IEEE Int., 419–426.

Zeng, L., Benatallah, B., Dumas, M., et al., 2003. Quality driven web service composition. In: Proceeding of 12th International World Wide Web Conference (WWW03), Budapest, Hungary, May 2003.

Zeng, L., Benatallah, B., Ngu, A., et al, 2004. QoS-aware middleware for web services composition. IEEE Trans. Software Eng. (TSE) 30 (5), 311–327.

Zheng, Huiyuan, Zhao, Weiliang, Yang, Jian, Bouguettaya, Athman, 2012. QoS analysis for web service compositions with complex structures. IEEE Transactions on Services Computing.