

QoS Broker based Architecture for Dynamic Web Service Discovery and Composition

Maya Rathore¹ and Ugrasen Suman²

¹Research Scholar, School of Computer Science & IT, Devi Ahilya University (DAVV)
Indore (M.P.), India, 91-7697575333

²Professor, School of Computer Science & IT, Devi Ahilya University (DAVV)
Indore (M.P.), India, 91-9826953187

¹mayarathore114@gmail.com, ²ugrasen123@yahoo.com

Abstract

With the widespread proliferation of web services, the demand on highly reliable and available services increases to satisfy business and personal needs. As the number of functionally similar web service increases on the Internet, Quality of Service (QoS) based web service selection and composition has gained much attention. However, the existing architectures for runtime service composition requires discovery of all functionally similar services, which consumes too much time. Instead, it can be possible to discover only those services which are highly reliable and thus included in composition. This issue can be solved by introducing a broker based architecture for automated dynamic web service discovery and composition in which the composition operation accepts as input a set optimized services along with their QoS specifications. The proposed architecture is able to reduce the composition time by performing optimized selection of services using Local Selection and Local Optimization (LSLO) approach during service composition. The architecture is fault tolerant and ensures improved reputation prediction accuracy of selected services using Mixed Opinion based Reputation Prediction (MORP) approach. In this paper, a comparative analysis is also performed on the basis of set of criterion to analyze the existing dynamic web service discovery and composition architectures for their strengths and weaknesses.

Keywords: QoS, Service Optimization, Workflow, Reputation, Broker-based architecture, Web Service Composition, LSLO, Non-functional requirement

1. Introduction

Web service technology is becoming popular tremendously because of its potential in many areas. Nowadays, many organizations implement their core business alone and outsource other application services to be run over the Internet. Thus, the ability to integrate heterogeneous cross-organizational services on the web dynamically is important in web service applications. A web service is an interface that describes a collection of operations that are network accessible through standardized XML based interfaces and protocols such as Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL) and Universal Description, Discovery and Integration (UDDI). Every service provides a role, such as service provider, a service consumer and a broker. The increasing number of web service on the can have similar functionality that creates a way for the consumers to use tools and techniques to search for suitable web services based on their requirements. One of the main advantages of web services

is the possibility of composing them for creating new value added services. Service compositions reduce the development time and create new applications. Web service composition is an active and important stream of research in the community of service-oriented computing.

There are two approaches of web service composition; namely, static and dynamic service composition. In static compositions, the aggregation of services is performed at design time. Static web service composition is not flexible as it is not adaptive to the runtime changes when it is in execution. On the other hand, dynamic web service composition aims at overcoming the problems of static web service composition. It provides flexibility for modifying, extending and adapting changes at runtime. The web service environment is highly dynamic in nature and in such a dynamic environment realization of dynamic web service composition has several issues. First, composition process should adapt to the customer requirements with minimal user intervention. Second, composition process should adapt the appropriate component services in a composition that is able to transparently adapt the environmental changes. Third, QoS information published by the service providers should be accurate and up-to-date. Fourth, the number of services available over the web is large and the time required for searching this huge service repository can be long. Finally, assessing the service reputation should not be only consumer centric.

Based on the aforesaid reasons, a broker based architecture is proposed for automated dynamic web service discovery and composition based on QoS. The proposed architecture accepts as input a set of optimized services for composition operation. The proposed architecture uses an optimal approach for finding the most suitable web services according to the consumer's requirements for service composition. It is able to reduce the composition time by performing the optimized selection of services using LSLO approach during service composition [2-3]. Also, the proposed architecture is fault tolerant and ensures improved reputation prediction accuracy of selected services by using MORP approach [6]. It performs active monitoring of services to provide accurate and up-to-date QoS of web services [4].

The remainder of this paper is organized as follows. Section 2 provides the necessary background knowledge and reviews the state-of-the-art related to broker based architectures for web service discovery and composition. Section 3 describes the proposed architecture and elaborates on its key components. Section 4 shows the interaction among various components of the proposed architecture and their operations. In Section 5, the proposed architecture is compared with the existing architectures for their strengths and weaknesses on the basis of seven criteria. Section 5 also evaluates and discusses the performance of proposed architecture. Finally, Section 6 concludes the paper along with the future work.

2. Literature Survey

Various architectures have been proposed by several researchers for dynamic web service discovery and composition. A broker-based architecture is presented for web services discovery in which the Service Quality Analyzer (SQA) is an independent component of broker [7]. SQA helps the safety of web services QoS characteristics against the possible interference of service providers. An approach based on learning automata is also proposed for evaluating web service QoS characteristics for service selection. In this paper, the online behaviour of web services is also considered in which QoS characteristics are changing dynamically. A web service discovery model is proposed in which functional and non-functional requirements are considered for web service discovery [8]. However, feedback is not collected from service consumer as reference to update QoS value. A framework for service selection and ranking is proposed which is based on Web Service Description Model

(WSDM) [9]. The model does not support the verification of QoS before registration of web services in UDDI registry.

Web service discovery architecture based on Multi Agent Systems (MAS) is proposed in which UDDI registry has the capability to store QoS information using t-Model data structures and MAS [10]. Agent services may be used to facilitate service registry access. Multi-agents are introduced in this architecture and thus, three multi agents are used; namely, response agent, certification agent and query agent to perform the interaction with the UDDI. A dynamic web service discovery architecture is presented that contains an extended UDDI to store the QoS information using t-model data structure and WS-QoS broker to facilitate the web service discovery at runtime [10-15]. The sub-components of WS-QoS broker are service publisher, verifier and certifier, service selector and Web Service Storage (WSS). The broker is also a web service through which the UDDI is accessed [11-13]. Its sub-components are service publisher, verifier and certifier, retrieval agent, quality analyzer and WSS. The agent performs the interaction with the UDDI [11, 13].

A broker-based framework is presented for securing intelligent web services composition and dynamic selection with QoS attributes [1]. The main functionality of the broker is to match web services according to security constraints and select the best web services to execute a business process. In this paper, the concept of security constraints of web service is considered that must be satisfied by each atomic web service of composition or final service of composition. The components of broker are composer, selector and publisher. The limitation with the proposed architecture is that only one of the QoS parameters is considered during composition and selection of optimal path. A web service discovery architecture is presented that contains two additional components with the traditional service discovery model, which are QoS registry to store the QoS information and QoS broker to facilitate web service discovery [16-17]. The QoS broker registers and edits the web service QoS property values through the QoS registry. The sub components of QoS broker are service selector, service publisher and quality manager.

A broker-based architecture focuses on the selection of web services based on their QoS and transactional properties to facilitate dynamic integration of atomic web services [18]. The objective of this architecture is to select the best web service that satisfies transactional properties as well as the requester's QoS constraints and preferences. It generates a workflow dynamically based on user's request and performs a better web service selection for the composition. The broker contains six main components, such as WFmodeller, WSlocator, transactional WS-Adapter, QoS WS-Adaptor, QoS manager, and WSBPEL generator.

An approach based on Delegation Web Service (DWS) is proposed for designing and developing web service discovery architecture for selecting web service more efficiently and with maximum load balancing [19]. In this approach, load balancing can be achieved by grouping the web services of similar type from the registry by DWS for each consumer's request. In this approach, each similar type of web service has one delegation web service. Each web service will be provided with some threshold value and by reaching above the value will make the service to get overloaded. The monitoring of QoS parameters can be performed using the standard called Web Service Discovery Management (WSDM), which includes two specifications, i.e., Management Using Web Services (MUWS) and Management of Web Service (MOWS). The MOWS and MUWS of the WSDM standards are used to monitor the QoS of each of similar web service consumed by the DWS. In this architecture, DWS is a mediator between service consumer and UDDI.

A WSSM-Q QoS based web services selection model is proposed to provide QoS support for web service publishing and selection [20]. The model manages the QoS of web services which includes defining the QoS model, collecting the QoS information, computing and

maintaining the QoS data. The components of proposed architecture are QoS broker, service consumer, service provider, UDDI, publish and QoS monitor. Dynamic web service composition architecture is proposed which is based on Business Process Execution Language (BPEL) and UDDI registry [21]. It can dynamically bind web services to running business processes and modify service function under exceptional situation. The QoS component is the core of the framework which monitors and manages other parts of the framework. BPEL process designer, BPEL process warehouse, BPEL execution engine, QoS component and UDDI are the components of the framework. The QoS specifications are stored with WSDL file. The architecture does not show about the management of QoS information as well as discovery process.

Broker based architecture is proposed that enables automatic QoS monitoring and adaptation by dynamically changing the execution paths of composed web services when needed [22]. It consists of dynamically changing the execution paths of composed web services by instrumenting the BPEL process. However, the architecture is able to handle only a single QoS constraint for service discovery and selection operation. A scalable architecture for automatic service composition based on four stages is proposed in which orchestration of nested workflows and composition property transformations is added to the existing process [23]. The workflow orchestration manager deals with nested automatic service compositions to reach a final goal that cannot be completed in one composition step. A framework for dynamic web service composition and parameters matchmaking is proposed to overcome the problem of updated services and [24]. In this framework, only the basic flow of composition is specified in the form of flowchart. A QoS-aware model for web services discovery is proposed by introducing QoS broker [30]. The merit of this model is that QoS description is introduced from the beginning of the web service providers publishing their services, and it does not need to modify the standard UDDI interface.

An intelligent broker based architecture to service composition and collaboration is proposed to enhance the power of broker [25]. The broker is capable of decomposing requested services into a number of subtasks, searching for the best fit services for each subtask, and composing and coordinating these services in execution. It employs a planner to generate service composition plans according to service usage and workflow knowledge, dynamically searches for services according to the plan, then invokes and coordinates the executions of the selected services at runtime. Dynamic service composition architecture is proposed which can automatically select a suitable service and dynamically bind it with invocation activity at runtime to improve the flexibility and reliability of BPEL process [26]. In this architecture, a control mechanism is also proposed which can be automatically incorporated into BPEL process to select reliable services and dynamically bind them at runtime.

A context-aware web service composition framework uses context awareness and agent technology into the execution of web service composition [27]. The framework is divided into four layers, such as the application layer, composition layer, coordination layer and service layer. It aims to improve the quality of service composition and provide the more suitable service composition to users at the same time. The basic features of the agent include autonomy, reactivity, adaptability and sociality. Detailed design of the MOdel-based SElf-adaptation of SOA systems (MOSES) is presented [28]. MOSES is a runtime adaptation framework for a SOA-based system architected as a service broker. A framework for dynamic composition of web services using templates in SOA is presented [29]. The key feature of this approach is its ability to react to exceptions such as service failures that may occur at runtime. A layered system for web service composition is proposed in which the input for the system is the specification of the desired service, including both functional and non-functional

requirements [15]. The composition operation accepts as input a generic composition graph which is defined on the basis of functional requirements. Finally, suitable composition is identified in terms of cost and non-functional requirements by solving a bi-objective shortest path problem on the transformed composition graph.

Although the existing architectures support runtime service integration and adaptation, still there are some limitations with the existing architectures. First, focus is given on the selection of service providers rather than on services to be composed [16-17]. Second, non-functional information published by the service providers may not always be trustworthy [16-17, 31]. Third, most of clients are not experienced enough to acquire the best selection of web service based on its described QoS characteristics [7, 31]. Fourth, most of the architectures assess the service reputation by considering only service users rating which may be maliciously provided [7, 9-14, 16-17, 20, 30]. Finally, most of the architectures perform global optimization of services during composition which causes service composition process slower [1, 15, 23, 25, 27-28]. Also, composition operation accepts as input a generic composition graph which is defined on the basis of functional requirements.

Based on the aforesaid limitations, a broker based architecture based on QoS is proposed for automated dynamic web service discovery and composition in which the composition operation is performed on the basis of optimized services alike [1, 15, 17, 22-23, 28]. The major components of the architecture are, service optimizer, Provider-Consumer-Broker-QoS (PCB-QoS) classification model, service discovery and composer and service publisher. The proposed architecture uses an optimal approach for finding the most suitable web services according to the consumer's requirements for service composition. It is able to reduce the composition time by performing the optimized selection of services using LSLO approach during service composition [3]. Also, the proposed architecture ensures the accuracy of selected services by using MORP approach [6]. It performs the active monitoring of services to provide accurate and up-to date QoS of services [2, 4].

3. Proposed Architecture

The proposed architecture for QoS based web service discovery and composition is shown in Figure 1 which the QoS broker is the central component for interaction. The main components of this architecture are, service optimizer, PCB-QoS classification model, service discovery and composer, service publisher, UDDI repository, service broker's repository, service consumer and service provider.

3.1 Service Provider

Service Provider's (SP) are the user who provides their services to consumer. The service provider describes the functional and non-functional QoS specification with the web service. The QoS broker can register and publish the web service in UDDI registry after the successful verification of provided functional and non-functional QoS specification.

3.2. Service Consumer

Service Consumer's (SC) are the user who request services from the broker. Generally, they do have problems with discovering and selecting services. Service consumers send request to QoS broker for web service discovery with functional and non-functional QoS specification. The broker finds the matching services and sends back to the consumer.

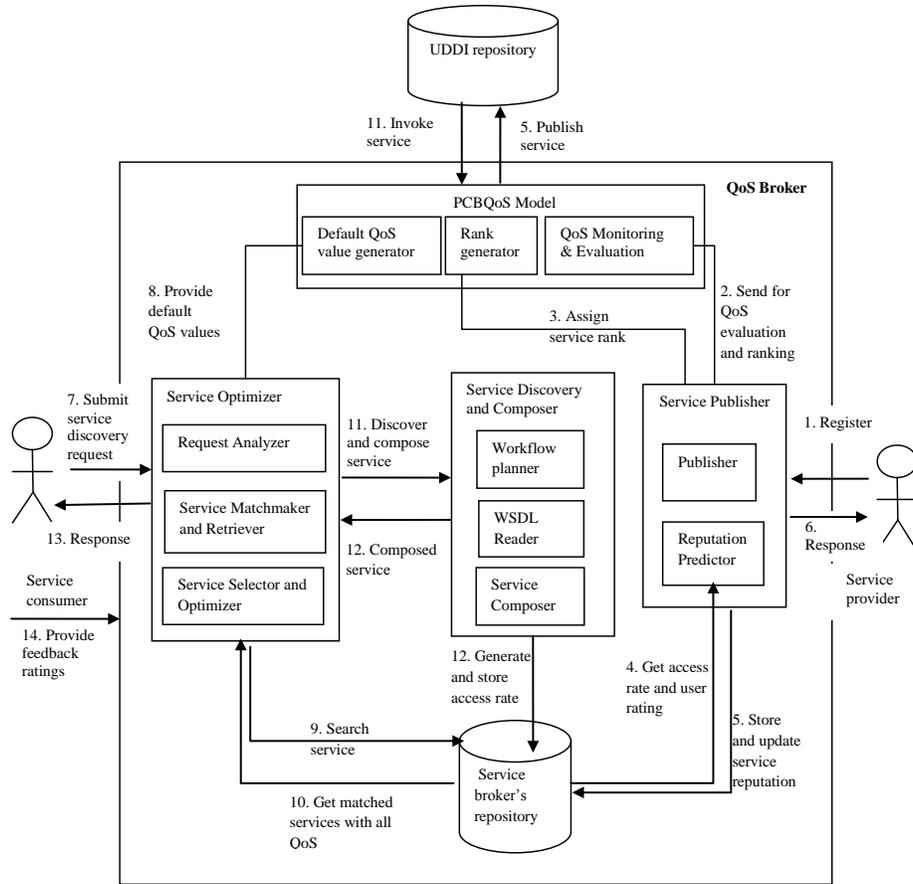


Figure 1. QoS Broker based Architecture for Dynamic Web Service

3.3. QoS Broker

It is an interface based web service through which service consumer and service provider both can interact to each other. QoS broker receives service discovery requests from service consumers, find the services that match their requirements and then returns the matched services to the consumer. Service provider can register and publish new services with the help of QoS broker interface. It works as a trustable third party mediator between the service consumer and provider. It is responsible for discovery, selection and composition of web services.

3.4. Service Optimizer

Service Optimizer (SO) selects the most appropriate services among the collection of functionally similar services by using the proposed LSLO optimization technique [3]. Its sub-components are:

3.4.1 Request analyzer: Request Analyzer (RA) analyses that the user provides the QoS requirements along with their functional requirement or not. If he/ she is unable to specify the QoS requirements due to lack of knowledge then this component considers the default QoS parameters along with the consumers' functional requirements for efficient service selection using PCB-QoS model.

3.4.2 Service matchmaker and retriever component: Service Matchmaker and Retriever (SMR) component searches the requested service along with the specified QoS requirements in the broker's local service repository and retrieves the functionally similar services along with their QoS values. This component also computes the overall quality score of each selected service by using the Average Real-Value Sum Method (ARSM) approach [2].

3.4.3 Service selector and optimizer: Service Selector and Optimizer (SSO) component is responsible for the selection of best candidate services among the collection of functionally similar services. This task is performed by applying the LSLO approach on the set of services [3].

3.5. Service Discovery and Composer

Service Discovery and Composer (SDC) component explores the functionality of each service which is selected by service selector and optimizer component and composes them in order to fulfill a complex request. This component discovers and composes only those services which are best suited to consumer requirements and decrease the chances of service unavailability. Its sub-components are:

3.5.1 Workflow planner: It is responsible for the generation of proper execution sequence of candidate services to be read by WSDL reader for a composition process. It is the software platform for the execution of the BPEL process and represents the user front-end for the composite service through broker.

3.5.2 WSDL Reader: It accepts as input all the selected WSDL file and read them for the process of service composition dynamically.

3.5.3 Service Composer: It performs the composition of abstract services which are highly reputed and trustable. It composes the services in a manner in which they arrive.

3.6 Service Publisher

Service Publisher (SP) facilitates the service registration, updating and deletion of web service related information, monitoring services; generate rank value and prediction of service reputation. Services to be registered are monitored for evaluating the specific functional and non-functional parameters by the QoS broker. Its sub-components are service publisher and reputation predictor.

3.6.1 Publisher: The service provider can publish their services with the help of this component. This component monitors the services for the specific non-functional parameters and accordingly assigns ranks.

3.6.2 Reputation Predictor: This component evaluates the service reputation with the help of user rating and access rate [6] for improving the trustworthiness of a service.

3.7 PCB-QoS Classification Model

It performs the following responsibilities with the help of its sub-components, which are default QoS value generator, rank generator and QoS monitoring and evaluation:

3.7.1 Default QoS value generator: As the QoS based service discovery and composition requires the specification of non-functional requirements along with the functional

requirements. This sub-component provides default QoS values to RA as the default specification of services non-functional requirements for the process of web service selection. This component is helpful in situation where the service consumer is unable to specify their non-functional requirements due to lack of knowledge.

3.7.2 Monitoring and Evaluation: It performs monitoring and evaluation of QoS parameters of registered services onto UDDI repository in order to prevent from malicious details.

3.7.3 Rank Generator: It generates the rank value for monitored services based on the aggregated QoS score obtained through ARSM approach [2].

3.7.4 UDDI Registry: A UDDI registry service manages information about service providers, service implementations and service metadata. Service providers can advertise the services they offer to UDDI through broker service. Service consumers can use UDDI to discover services that suit their requirements and to obtain the service metadata needed to consume those services through broker. The UDDI registry is connected to the QoS broker as a backend registry using simple object access protocol (SOAP) connections. During processing of the requests, the QoS broker acts as a client to query the UDDI registry for information.

3.8 Service Broker's Repository

Service Broker's Repository (SBR) stores the functional and non-functional QoS specification of services collected by QoS broker from both service provider and consumer. The service brokers will then use the stored information for its various operations performed for web service discovery and composition.

4. Architectural Component Interaction

The broker is designed with four major functional components; namely service publisher, service discovery and composer, PCB-QoS model and service selector and optimizer. A set of functions is defined for each component to fulfil the requester's objective of selecting the best web service that satisfies QoS constraints. The interactions among various components are provided in subsequent sections.

4.1. Interaction Among Various Components for Service Discovery and Composition Operation

The interaction among various components of architecture for runtime web service discovery and composition is shown in Figure 2. The various components involved are, request analyzer, PCB-QoS classification model, QoS broker, service optimizer, service discovery and composer, service consumer and UDDI. The following sequence of operation shows the interaction among various components.

- The service consumer invokes the broker interface to request service discovery with functional and QoS requirements.
- The broker first analyses the request through request analyser to determine its nature whether it is simple type or complex type service discovery request. Secondly the RA also analyzes whether the QoS parameters are specified with the request or not.

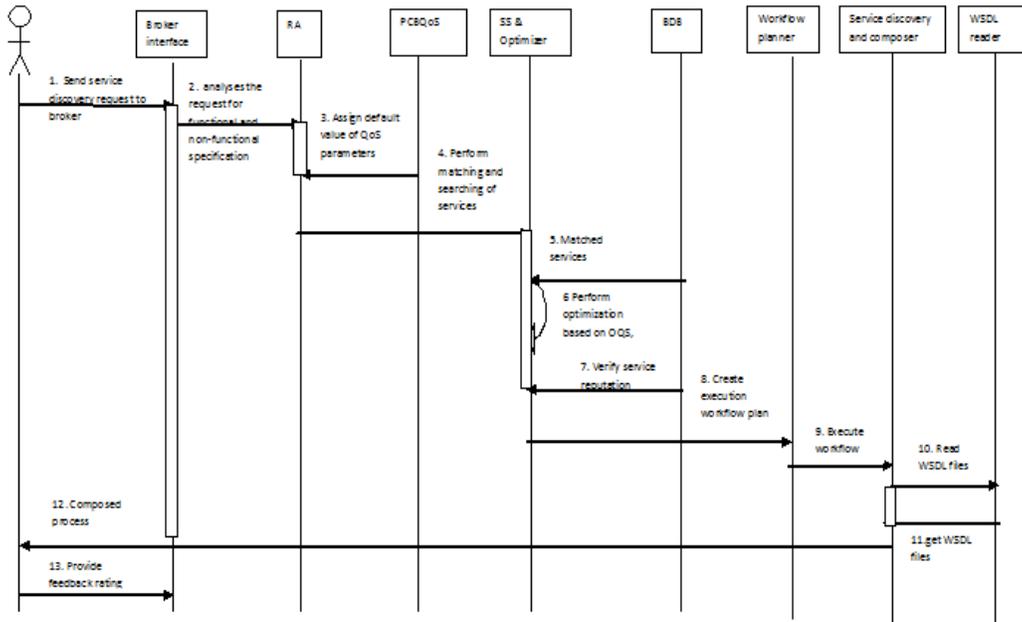


Figure 2. Sequence Diagram for Service Discovery and Composition

- If no QoS requirements are specified with the request then default QoS parameters are attached from PCBQoS model and send back to RA, otherwise skip.
- The broker then searches the services with their functional and non-functional requirements in its local service repository using searching and matching components. Functionally similar services are retrieved from the repository along with their aggregated QoS score [2, 4].
- The retrieved services along with their aggregated QoS score are forwarded to service optimizer and selector component which optimizes the services based on their QoS score using LSLO approach and select the best service among them [6].
- The reputation of each selected candidate service is then evaluated by the reputation prediction component (RP) to assure its trustworthiness and reliability.
- The reputed services are provided to discovery component to explore the functionality of each service.
- The discovered services are sent to workflow planner to generate the execution workflow plan for integrating services into a single service.
- Service discovery and composer component executes the workflow and extracts the contents of each WSDL file using WSDL reader.
- WSDL reader reads the WSDL files and sends back to Service discovery and composer component.
- Return integrated service to service consumer to invoke.

- Consumer sends the feedback about the service after invocation through broker interface.

4.2 Interaction among Various Components for Service Publishing Operation

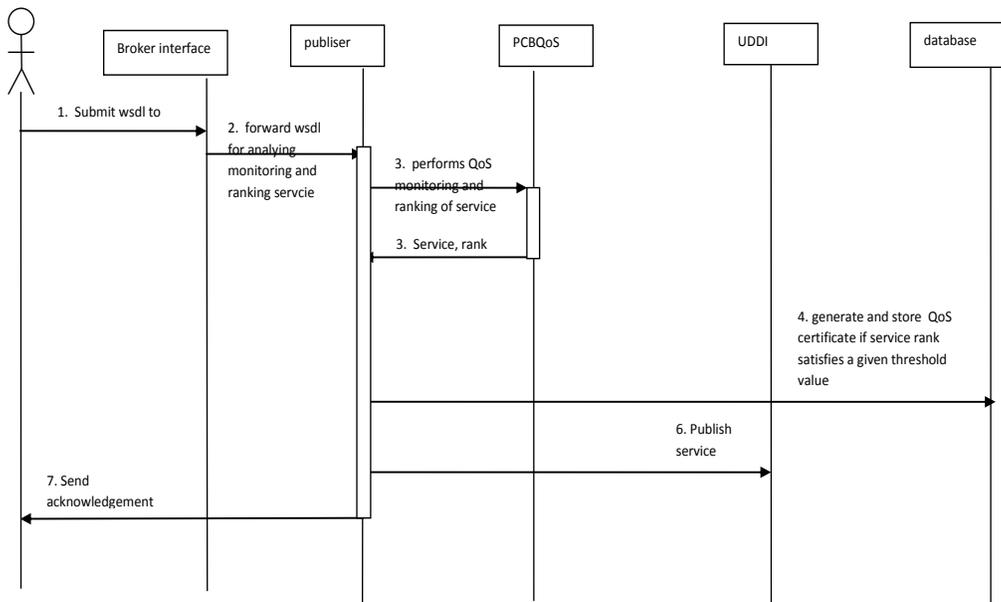
The sequence of operation for service publishing is shown in Figure 3. The various components involved are, broker interface, publisher, PCB-QoS model classification, UDDI, service provider and broker’s local database. The sequence of operations performed by this component is,

- The service provider first registers the service to broker through broker interface.
- The broker analyses the service for the verification of its specified functionality.
- After the verification of its functionality, broker monitors the service for the specific QoS parameters and generates the QoS report.
- This QoS report is used to assign initial rank value to a newcomer service.
- This rank value is stored into the broker’s database which is further used by consumer to select the best service.

4.3. Interaction among Various Components for Reputation Assessment

The sequence of operation for reputation assessment of services by service publisher component is shown in Figure 4. The various components involved are, broker interface, UDDI, broker’s repository, service consumer and reputation assessment. The sequence of operation is:

- Service consumer invokes the broker interface to search and discover the service.



nFigure 3. Sequence Diagram for Publishing Service

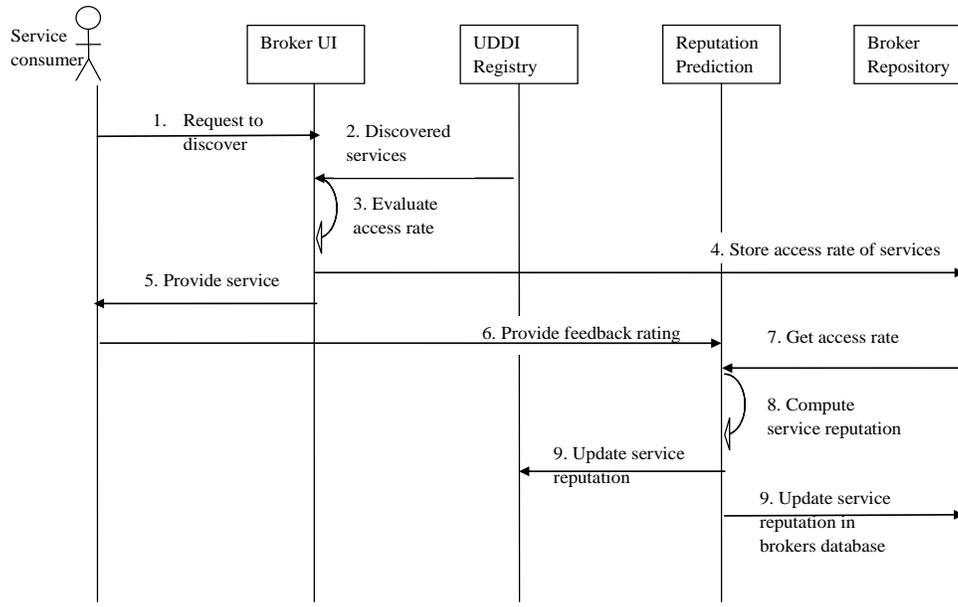


Figure 4. Sequence Diagram for Prediction of Service Reputation

- Broker searches the UDDI as well as local repository for discovering the services for consumer.
- Simultaneously broker stores the access rate of the requested service into its local repository.
- Service consumer provides feedback about the service after invocation to reputation prediction component of service broker to store.
- Reputation prediction component gets the access rate from the repository and assess the service reputation using MORP [6].
- Update the service reputation on its local repository as well as UDDI repository.

5. Discussion

Table 1 shows the comparative analysis of various existing broker based architectures with the proposed architecture. The strength and weakness of various QoS aware web service discovery and composition architectures can be identified by analyzing them on the basis of following seven criteria:

C1: Is the architecture able to construct the non-functional properties automatically if not specified?

C2: Is the requester allowed to specify the desired QoS properties for selection?

C3: Does the architecture supports selection mechanism defined for multiple QoS properties?

C4: Does the architecture assess the service reputation based on user ratings only?

C5: Is the architecture meant for provider, consumer and broker related operations?

C6: Is the architecture able to discover only those services which are selected for composition?

C7: Does the architecture supports service consumer involvement in service selection?

Table 1. Evaluating Existing Broker based Architecture with the Proposed Architecture

| Approaches | Architectures/ framework | Criteria | | | | | | |
|--------------|-----------------------------|----------|-----|-----|-----|-----|-----|-----|
| | | C1 | C2 | C3 | C4 | C5 | C6 | C7 |
| broker based | [9] | No | Yes | Yes | Yes | Yes | No | Yes |
| | [10-14][16-17][20][31][33] | No | Yes | Yes | Yes | Yes | No | Yes |
| | [7] | No | Yes | Yes | Yes | Yes | Yes | Yes |
| | [22] | No | Yes | Yes | No | Yes | No | No |
| | [25] | No | Yes | Yes | NA | Yes | No | Yes |
| | [27] | No | Yes | Yes | No | Yes | Yes | Yes |
| | [28] | No | Yes | Yes | NA | Yes | No | Yes |
| | [15] | No | Yes | Yes | NA | No | No | Yes |
| | Proposed architecture | Yes | Yes | Yes | No | Yes | Yes | Yes |
| Broker-less | [24] | No | Yes | Yes | NA | No | No | Yes |
| | [26] | No | Yes | NA | NA | No | No | No |

Table 1 show that the proposed architecture meets the above specified criteria as compared to existing web service discovery and composition architecture. First, it is able to able to construct the non-functional properties automatically if not specified, whereas, existing architectures does not provide this feature. Second, it generates the workflow plan by using only reputed candidate services during composition. Finally, the service reputation is not fully dependent on user ratings like existing architectures.

The proposed architecture is also evaluated by computing its time complexity for service selection and composition and comparing it with the time complexities of other architecture as shown in Table 1. It has been observed that the time complexity of most of the existing architectures are $O(n^3)$, $O(2^n)$ and that of LSLO approach is $O(n^2 \log n)$ for service discovery and composition operation [6]. The computational time complexity (in ms) of 12 approaches are compared [1, 17-19, 22-23, 25-27, 30-32] to study the variations in execution time as shown in Figure 5. The performance of proposed architecture is better than those of existing architectures having complexity of $O(n^3)$ and $O(2^n)$ for a limited number of task. In Figure 5, execution time will be increased linearly with the increase in the number of task. This experiment is conducted to investigate the increase in computation time with the increase in number of candidate web services for each task. In this experiment, the number of abstract web services is fixed to 15 and the number of concrete web service for each of the abstract service depends on the search results. As the number of task increases, the time complexity of proposed architecture also increases.

6. Conclusion and Future Work

In this paper, a broker based architecture is proposed for automated dynamic web service discovery and composition in which the composition operation is defined on the

basis of optimized services with their QoS specifications. The proposed architecture is able to reduce the composition time by performing the optimized selection of services using LSLO approach during service composition. In this architecture, the broker is capable of composing services dynamically using only those services which are trustable and are of high reputation. It can be helpful in reducing the composition time of services as well as provide up-to-date service details to service consumer. The proposed architecture is fault tolerant and able to handle the composition process during failure. Apart from the various advantages, the only limitation with the proposed architecture is that it is unable to generate different composition paths dynamically. The implementation details and the performance evaluation of the proposed architecture will be presented in the next version of this paper.

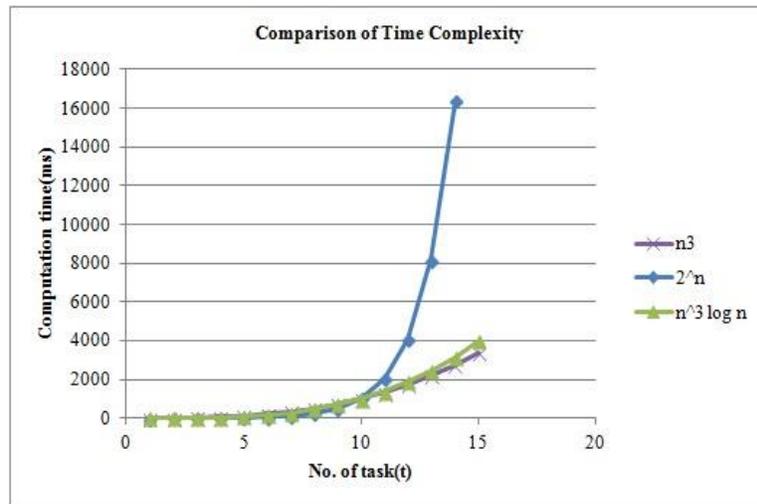


Figure 5. Comparison of Time Complexities in Existing and Proposed Service Selection Approaches

References

- [1] M. H. Agdam, "A flexible and scalable framework for QoS aware web services composition", 5th International Symposium on Telecommunications (IST), (2010) December 4-6, pp. 521-526.
- [2] M. Rathore and U. Suman, "An ARSM approach using PCB-QoS classification for web services: A multi-perspective view", Proceedings of the International Conference on Advances in Computing, Communications and Informatics(ICACCI), (2013) August 22-25, pp. 165-171.
- [3] M. Rathore and U. Suman, "Web service selection algorithm for dynamic service composition using LSLO approach", Proceedings of the International Conference on Informatics, Electronics & Vision (ICIEV), Bangladesh, (2013) May 17-18, pp. 1-6.
- [4] M. Rathore and U. Suman, "Evaluating QoS parameters for ranking Web service", Proceedings of the 3rd International Advance Computing Conference (IACC), (2013), pp. 1437-1442.
- [5] M. Rathore and U. Suman, "A quality of service broker based process model for dynamic web service composition", Comput. Sci., vol. 7, (2011), pp. 1267-1274.
- [6] M. Rathore and U. Suman, "A Mixed Opinion based Reputation Prediction Approach for Web Service Discovery", Communicated.
- [7] R. Tabein, M. N. Moghadasi and A. Khoshkbarforousha, "Broker-based Web Service Selection using Learning Automata", Proceedings of the International Conference on Service Systems and Management, (2008), pp. 1-6.
- [8] V. Diamadopoulou, C. Makris, Y. Panagis and E. Sakkopoulos, "Techniques to support web service selection and consumption with QoS characteristics", Journal of Network and Computer Applications, vol. 31, no. 2, (2008), pp. 108-130.

- [9] G. Zou, Y. Xiang, Y. Gan and D. Wang, "Agent-based web service selection and ranking framework with QoS", Proceedings of 2nd IEEE International Conference on Computer Science and Information Technology (ICCSIT), (2009), pp. 37-42.
- [10] T. Rajendran and P. Balasubramanie, "An Efficient Multi-Agent-Based Architecture for Web Service Registration and Discovery with QoS", European Journal of Scientific Research, vol. 60, no. 3, (2011), pp. 421-432.
- [11] T. Rajendran and P. Balasubramanie, "Efficient Approach towards an Agent-Based Dynamic Web Service Discovery Framework with QoS Support", International Symposium on Computing, Communication, and Control (ISCCC 2009), Proc .of CSIT, IACSIT Press, Singapore, vol. 1, (2011).
- [12] T. Rajendran and P. Balasubramanie, "An Efficient WS-QoS Broker Based Architecture for Web Service Selection", International Journal of Computer Applications, vol. 1, no. 9, (2010).
- [13] T. Rajendran and P. Balasubramanie, "An Efficient Architecture for Agent based Dynamic Web Service Discovery with QoS", Journal of theoretical and applied information technology, vol. 15, no. 2, (2010).
- [14] T. Rajendran and P. Balasubramanie, "An Optimal Broker-Based Architecture for Web Service Discovery with QoS Characteristics", International Journal of Web Services Practices, vol. 5, no. 1, (2010), pp. 32-40.
- [15] S. Chakhar, "QoS-Enhanced Broker for Composite Web Service Selection", Proceedings of 8th International Conference on Signal Image Technology and Internet Based Systems, (2012) November 25-29, pp. 533-540.
- [16] D. Antony D'Mello, V. S. Ananthanarayana and T. Santhi, "A QoS Broker Based Architecture for Web Service Selection", Proceedings of 2nd Asia International Conference on Modeling and Simulation (AICMS), (2008), pp. 101-106.
- [17] S. Karimi and M. S. Babamir, "A QoS Broker Based Architecture for Secure Intelligent Web Service Composition", Proceedings of First International Conference on Communication Engineering, (2010), pp. 110-115.
- [18] K. Johny, "A New Broker-Based Architecture for TQoS Driven Web Services Composition", Proceedings of Int. Conf. on Advances in Computer Engineering, (2011) December 10-12, pp. 1-4.
- [19] G. Vadelou, "A QoS Based Web Service Selection through Delegation", International Journal of Scientific and Engineering Research, vol. 2, no. 5, (2011).
- [20] L. Sha, G. Shaozhong, C. Xin and L. Mingjing, "A QoS based Web Service Selection Model", International forum on Information Technology and Applications, Chengdu, (2009), pp. 353-356.
- [21] S. Pei and D. Chen, "Research on Dynamic Web Service Composition Framework Based on QoS", Information Technology Journal, China, vol. 8, no. 10, (2011).
- [22] Q. Ming, F. Khendek, A. Serhani, R. Dssouli and R. Glitho, "Automatic QoS adaptation for composite web services", Proceedings of International Conference on Innovations in Information Technology (IIT), (2008) December 16-18, pp. 180-184.
- [23] P. Incheon, C. Wuhui and N. H. Michael, "A Scalable Architecture for Automatic Service Composition", IEEE Transactions on Service Computing, vol. 7, no. 1, (2014), pp. 82-95.
- [24] M. Allauddin and F. Azam, "Dynamic web service composition and parameter matchmaking", International Journal of Computer Applications, vol. 36, no. 9, (2011).
- [25] Y. Zhang and Z. Hong, "An Intelligent Broker Approach to Semantics-based Service Composition", Proceedings of 35th IEEE Annual Computer Software and Applications Conference, (2011), pp. 20-25.
- [26] C. Po-Yuan, H. San-Yih and L. Chien-Hsiang, "A Dynamic Service Composition Architecture in Supporting Reliable Web Service Selection", Proceedings of 5th International Conference on Service Science and Innovation, (2013), pp. 237-238.
- [27] S. Wenxi, Z. Xiuguo, Y. Yuan and H. Tingting, "Context-aware web service composition framework based on agent", Proceedings of International Conference on Information Technology and Applications, (2013) November 16-17, pp. 30-34.
- [28] V. Cardellini and S. Iannucci, "Designing a Broker for QoS-driven Runtime Adaptation of SOA Applications", Proceedings of IEEE International Conference on Web Services (ICWS), (2010) July 5-10, pp. 504-511.
- [29] K. Rajaram and C. Babu, "Template based SOA framework for dynamic and adaptive composition of Web Services", International Conference on Networking and Information Technology, (2010) June 11-12, pp. 49-53.
- [30] Y. Gang, L. W. Chan, Y. Jun and C. Shi, "A QoS aware model for web service discovery", First International Workshop on Education Technology and Computer Science (ETCS), vol. 3, (2009) March 7-8, pp. 740-744.
- [31] H. Pon, J. Akilandeswari and R. Sarathkumar, "Dynamic Web Services Discovery and Selection Using QoS-Broker Architecture", International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, (2012) January 10-12, pp. 1-5.
- [32] M. Alrifia, T. Risse and W. Nejdl, "A hybrid approach for efficient Web service composition with end to end QoS constraints", ACM Transactions on the Web, vol. 6, no. 2, Article 7, (2012), 31 pages.

- [33] G. Amirthayogam, M. Rathinraj and A. Gayathri, "Web Service Discovery with QoS-An Agent-Based Approach", International Journal of Futuristic Science Engineering and Technology, vol. 1, no. 1, (2013).

Authors



Maya Rathore, has received her Master Degree in Computer Applications from Rajiv Gandhi Technical University, Bhopal, India and pursuing PhD in Computer Science from Devi Ahilya University Indore, India under the supervision of Dr. Ugrasen Suman. Presently she is working as a Assistant Professor at Shri Vaishnav Institute of Technology & Science, Indore, India. She is having 8 years of teaching and 5 years of research experience. His areas of research are Software Engineering, Service-Oriented Computing, Distributed computing and Cloud computing. She has published around 15 research papers in various national and international conferences and journals.



Dr. Ugrasen Suman, has received his Master Degree in Computer Applications from Rani Durgawati University Jabalpur and PhD Degree in Computer Science from Devi Ahilya University Indore, India. Presently he is working as a Professor at School of Computer Science & Information Technology, Devi Ahilya University, Indore, India. He is having more than 12 years of teaching and research experience.

His areas of research are Software Engineering (Software Reuse, Software Maintenance, Evolution & Reengineering, Agile Methodologies, Empirical Analysis, Software Architectures), Information System, Service-Oriented Computing, Knowledge Management & Mining. He has guided Two PhD Scholars, Four M.Tech by Research Scholars, 32 M.Tech Dissertations and more than 55 other PG Projects. Currently he is guiding Eight PhD Scholars and Four M.Tech Scholars.

He has authored a book Software Engineering: Concepts & Practices published by Cengage Learning, ISBN: 9788131519301, 2013. He has published more than 60 research papers in National & International Journals/ Conferences. He is also working as Dy. Coordinator of a UGC-SAP research project of Data Mining and Software Engineering.

He has organized several workshops and a National Conference on Architecturing Future IT Systems in 2008. He is a reviewer/Editorial member of computer science journals/ conferences in various publications. He has chaired various technical sessions in conferences and delivered lectures. He is a Member of IEEE, IEEE-CS, Life Member of CSI, Senior Member of IACSIT, and IAENG.

