Journal of Web Engineering, Vol. 9, No.4 (2010) 327-346
© Rinton Press

A QOS ENHANCED FRAMEWORK AND TRUST MODEL FOR EFFECTIVE WEB SERVICES SELECTION

ZHEDAN PAN and JONGMOON BAIK

Korea Advanced Institute of Science and Technology, Republic of Korea
panzhedan@icu.ac.kr, jbaik@kaist.ac.kr

Received July 25, 2008
Revised January 11, 2010

Service Oriented Architecture (SOA) has become a promising paradigm for software development. One of the most important research topics in SOA is Web service selection which means to identify best services among a bunch of services with same or similar functions but having different QoS (Quality of Service). Many previous approaches, such as QoS models with quality criteria and selection algorithm, have been proposed to optimize Web service selection. However, in current research, quality values normally come from service providers, who have high possibility to exaggerate these values for advertisement. It is also argued that reputation based on an average user rating is not enough to indicate the trust degree of Web services and service provider. In addition, handling dynamic nature of Web services is still a challenging problem for dynamical Web service selection. In this paper, these problems are focused. First a QoS enhanced framework for effective Web service selection is proposed. Then a Trust model is built, which is composed of TQoS model, Decision model and Trust correction. It is claimed that a Web service can be regarded as trustful if QoS values received by consumers and tested by registry are no less than QoS values promised by providers. A prototype of the proposed framework is implemented, including SC agent, SR agent and QoS Enhanced SR. In addition, a scenario about a Tour agency's Web service selection according to its business process is implemented. To validate effectiveness of proposed approach, we compared it with other approaches, such as Euclid approach and Fuzzy approach. Numerical simulation shows that proposed approach performances better other approaches in terms of obtained quality values.

Key words: Service Oriented Architecture, Web services, Web service Selection, Quality of services, Trust Model, TQoS, Trust Correction
Communicated by: M. Gaedke & P. Fraternali

1 Introduction

Service Oriented Architecture (SOA) becomes a promising paradigm for software development due to its benefits in cost-efficiency, agility, adaptability, and leverage of legacy investment [1]. SOA enables flexible connectivity of applications or resources by representing every application or resource as services with standardized interfaces. According to the definition of SOA, three parties are involved: service provider, service registry and service consumer. Service provider can publish Web services information to service registry. Service registry serves as a service repository, managing all the registered services and identifying appropriate services according to service consumers' functional requirements. Service consumer can discover services from service registry, choose one service and then bind to the service provider to invoke the discovered services.

There are already some XML-based standards to facilitate SOA implementation. For example, WSDL (Web Service Description Language) [21] is for interface definition and binding, SOAP (Simple Object Access Protocol) [22] for message exchange, SLA (Service Level Agreement)[24] for definition of negotiated contract between two parties, UDDI (Universal Description, Discovery and Integration) [23] for service discovery and BPEL (Business Process Execution Language) [25] for service composition. These standards are foundations of Web service technologies.

However, current service registry can only support Web service discovery based on functional aspects of services. Nowadays, consumers are not only interested in functionalities of Web services, but also qualities of service (QoS), which is a set of non-functional quality criteria like execution time and availability. Therefore, one of the most important research topics in SOA is Web service selection: identify the optimal Web service among a bunch of services offering same or similar functions but having different QoS. This means that service registry should not only select services according to what a service can do, but also how well the service performs. If service registry focuses only on requested functionality to select services, a poor quality, expensive, and time consuming service may be selected.

Several generic service quality criteria such as execution time, cost, reliability and availability, can be adopted for Web service selection. However, problem is that the advertised QoS information of a Web service is not always trustworthy. A service provider may exaggerate QoS information to attract more customers, or the published QoS information may be not the latest one. How consumers trust these QoS data to select Web services?

In addition, currently QoS reputation is considered as a rating of a service over a specific period of time. However, each consumer has his own requirement for each quality criteria. For example, consumers may give high score to services with high execution time if execution time is significant to them, while other customers may give high score to services with low cost if cost is significant to them. Therefore, reputation based on an average user rating is not enough to indicate why Web services are good and whether Web services and service providers are trustful or not [4].

Handling dynamic nature of Web services, such as sudden disappearance of certain Web services, or consumers’ changes of business process or requirements for Web services is still a challenging problem. To achieve the goal of dynamic Web service selection, which enables consumers to discover Web services satisfying their requirements automatically at run time instead of at design time, QoS enhanced Web service selection must be automated dynamically.

In this paper, focusing on the problems mentioned above, QoS enhanced framework for Web service selection is proposed, in which QoS enhanced SR (Service Registry), SR agent, and SC (Service Consumer) agent are added into SOA. Then a Trust model for Web service selection is built, which includes TQoS (Trusted QoS) model, Decision model and Trust correction. It is claimed that a Web service is trustful if QoS values received by consumers and tested by registry are no less than QoS values promised by providers.

The proposed QoS enhanced framework for Web service selection is implemented with a scenario. Dynamic Web service selection according to the Tour agency’s request is simulated. For evaluation, the proposed approach is compared with other approaches for Web service selection, such as Euclid approach and Fuzzy approach. The comparison is based on concepts, Web service ranking, as well as quality values of execution time, cost, availability, reliability, reputation and capability.

The paper is organized as follows. In Section 2, related works about QoS enhanced Web service selection are introduced. In Section 3, QoS enhanced framework for Web service selection is presented. In Section 4, Trust model for Web service selection is described. Section 5 introduces the implementation of the proposed framework and a scenario of Web service selection. In Section 6, validation of the framework and Trust model is provided. Lastly, conclusion and future works are given.

2 Related works

Research on Web service selection has been described in many previous publications. These researches are categorized into 3 groups: Broker-based approach, QoS Model-based approach, and Semantic-based approach.

2.1. Broker-based approach

M.Serhani et al. [2] proposes a QoS Broker based architecture, which includes four components: Web services Broker, Web services provider, Web service clients and UDDI enabled QoS registry. A two phase verification technique and certification from the QoS point of view are also proposed. V. Cardellini et al. [3] proposes a Broker architecture which includes composition manager, selection manager, optimization engine, execution path analyzer and QoS monitoring. Zhengdong Gao et al. [4] propose a QoS-prediction based service selection framework, which includes Service Management Center (SMC), QoS prediction Broker (QoS PB) and QoS predict Service (QoS PS). The core of this framework is to design and implement Back Propagation Neural Network to predict performance of service.

In many researches, Broker based architecture becomes foundation for Web service selection. However, these frameworks are still in concept stage, and there are still not many implementations and practical applications of Broker based framework.

2.2. QoS Model-based approach

Many Web service selection researches focus on modeling QoS and optimizing service selection algorithms. J. Hu et al. [9] identify execution cost, execution time, reliability, availability, and reputation as QoS criteria, propose a decision model of QoS criteria called DQoS for evaluating Web services. L. Yang et al. [10] identified price, duration, reputation, success rate, availability and matching degree as quality criteria, and proposed QoS driven dynamic selection of composite Web services, which takes account of both the QoS properties and interface matching degree. H. Tong et al. [13] proposed a fuzzy multi-attribute decision making algorithm for Web service selection based on quality of service to select the most appropriate one with the highest degree of membership belonging to the positive ideal solution. L.Taher et al. [7] proposed a QoS constraint model to establish association relationship between different QoS properties. A QoS match making algorithm is proposed to map QoS requirements of consumers with the published QoS information of providers. Euclidean distance is used to measure this similarity distance.

However, many QoS model based approaches focus on identifying quality criteria and optimizing selection algorithm, trustworthiness of QoS data is not considered in the QoS model for Web service selection.

2. 3. Semantic-based approach

With the popularity of semantic Web and automatic Web service composition, semantic Web service selection is introduced to focus on meaning matching rather than key word matching. Ontology can help to build the common concepts of QoS model and becomes one of the methods for semantic Web service selection. H. Chua et al. [14] propose a semantic Web service model based on multiple aspects including operations, domain, functions, QoS, business rules and so on. A semantic similarity matching algorithm for multiple aspects is proposed to select Web services. M. Sensoy et al. [15] advocate an objective experience-based (semantic) approach for service provider selection, rather than rating-based service selection. An ontology is built up to represent a consumer’s experience with a service provider to capture subtle details including the context in which the service is requested. E. Michael Maximillien et al. [16] address dynamic service selection via Web Service Agent Framework (WSAF) coupled with a QoS ontology so as to best meet user needs. WSAF incorporates service selection agents that use the QoS ontology and an XML policy language that allows service consumers and providers to expose their quality preference and advertisements.

However, currently semantic based approach is not mature enough to be applied in practical Web service selection.

3 QoS Enhanced Framework for Web Service Selection

In this section, first QoS enhanced framework for Web service selection is introduced. Then detail information about SC agent, SR agent and QoS enhanced Registry is described. Test mechanism and trustable QoS data and so on are discussed.

3.1. QoS enhanced Framework and Web Service Selection Process

Figure 1 shows the proposed QoS enhanced framework for Web service selection. In the proposed framework, service registry is enhanced with Web service selection and QoS data management. On service consumer and service registry side, agents are adopted for collecting QoS data.

The following shows the Web service selection process under the proposed framework:

1) Service provider publishes Web services to service registry. The published information includes WSDL information, functionalities, and QoS data of Web services.

2) After QoS enhanced service register get the registered Web service information, its agent connects to Web services provided by service providers to test Web services using test cases and get QoS data from test.

3) Service consumer discovers Web services from service registry based on the functionalities (or tasks) performed in business processes, and then select Web services with best Web service among these discovered Web services according to customer QoS requirements and Web services’ QoS data from the registry.

4) After identifying all Web services in business processes, Web services are composed and executed.

5) Service consumer side agent uploads QoS data of Web service execution to QoS Registry.

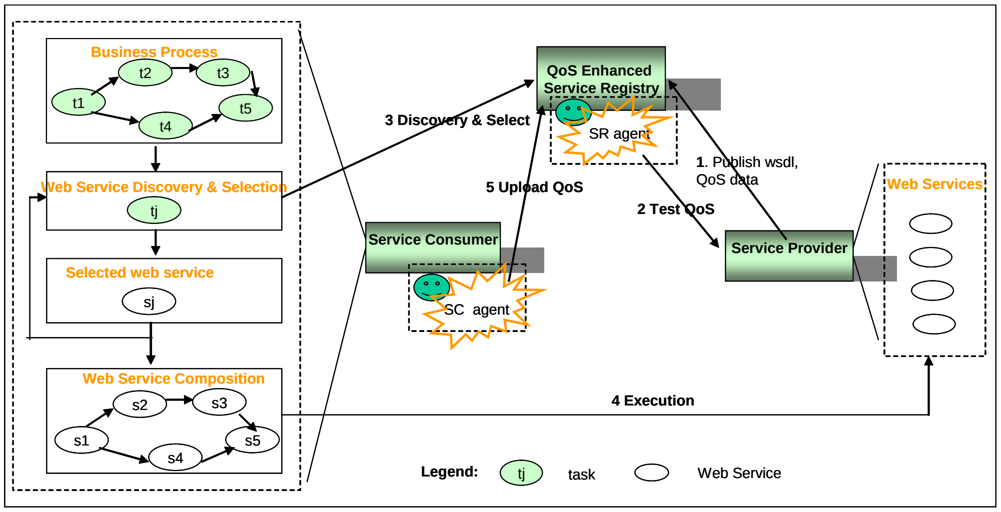


Figure 1. QoS enhanced framework for Web service selection

3.2. SR Agent, SC Agent and QoS Enhanced Service Registry

As mentioned above, advertised QoS data is not surely true. Testing behaviors of Web services by service registry (third party) is a good way to achieve real QoS data of Web services. On QoS enhanced SR, each registered Web service can be implemented with a SR agent for Web service testing and QoS data collection. Test cases can be provided by service provider, or can be generated automatically by service registry, or provided by service consumers. There are some researches on automatically test cases generation based on WSDL[17,18]. However, test case generation is out of the scope of this paper.

According to service consumers' experience and intuition, it is easy for them to estimate the satisfaction degree or reputation of Web services, but it is not easy for them to know whether they already get the quality of services promised by service providers, that is, whether quality of services received is the same as or higher than quality of services promised by service providers. So SC Agent is to monitor the behavior of the Web service from service consumer side through the execution of Web services and collect QoS data of Web services execution. Since automatic invocation is an unsolved problem, in this paper, a minimalistic approach such as SOAP pinging is used [26].

QoS enhanced service registry is considered as a third party with extended capability to manage QoS data objectively. Besides of managing Web services and discovering Web services based on functionalities, it performs several other functionalities. QoS enhanced SR needs to identify the best Web service to service consumers according to their QoS requirements. In addition, it also takes charge of collecting QoS data of Web services from SC agent and SR agent, and calculating QoS score. In short, QoS enhanced SR plays the following 2 additional functions:

1) Collect QoS data and Calculate QoS score
2) Select Web services according to consumers' QoS requirements

Difference between service discovery and service selection is obvious: service discovery is to identify Web services based on functionalities, while service selection is to identify Web services based on non-functionalities.

3.3. Test Mechanism of SR

According to the proposed framework, Service Registry needs to test Web services provided by service providers. Test includes "what to test", "when to test" and "how to test". Section 4. 1 introduces more information about "what to test" for QoS based Web services selection. Section 5.1 provides detail information about "how to test". This section mainly describes "when to test".

- New: First of all, when Web service is newly registered, it will be scheduled to test directly. This will be full test and all the QoS criteria will be tested.
- Changed: When the information of registered Web services, especially QoS information is updated by service providers, Web services will be scheduled to test. Updated or changed QoS information will be focused on the test.
- Different: If a certain Web service's newly collected QoS data from service consumer differs too much of that from service registry agent, then Web service will be scheduled to be tested. For example, this difference can be 50% increase or decrease.
- Maintained: As a kind of regular test or maintenance test, Web services will be tested every two months. With the increase of number of Web services, this time duration can be extended to three months or more considering the performance of service registry.

3.4. Discussion about trustable QoS data

Since we argue that QoS data from service provider is not that trustable, we identify two other sources of QoS data which are more trustable.

1) Service Registry

Currently quality certification of software products is performed by third and famous party based on the same quality criteria and testing mechanism. Normally Service Registry is third party owned by large and famous company or parties, such as Microsoft and IBM. These companies are trusted by consumers. Test done by Service Registry for each Web service will be based on same, certified and reasonable mechanism. Therefore, collected QoS data are more objective and more trustable.

2) Service Consumer

Currently many e-commerce services provide user rating on certain products. For example, eBay provides rating information of consumers' opinion about the products according to their usage. Service consumer agent can monitor the behavior of Web services according to consumers' real usage. With the increase of QoS data from many service consumers, based on Central Limit theory, QoS data will approach to a certain mean value, which can be very objective and trustable.

3.5. Economic consideration of the framework

Compared to standard SOA structure, in the proposed framework, SR agent and SC agent are added. Adding these agents has Pros and Cons. As for Pros, trustable QoS data can be attainable so that right web services can be selected. As for Cons, SOA structure becomes more complicated and more functions needs to be instrumented on Service Registry side and Service consumer side, which lead to certain cost. Some economic reasons for adding agents are as follows:

SR agent: One of the functionality of SR agent is to monitor Service Provider. By monitoring Service Provider, Service Registry can get real QoS data so that the right web services can be selected to Service Consumer. If Service Consumer can get the right web services, they can trust Service Registry and use it continually. This is the economic reason for pushing Service Registry to monitor Service Provider.

SC agent: On the Service Consumer side, SC agent is needed to be instrumented. By instrument it to measure some criteria automatically and providing these data to Service Registry, Service Consumer can benefit to get better and trustable web services. This is Win-win strategy and can be long term sustainable.

4 Trust Model For Web Service Selection

This section is to describe a Trust model for Web service selection based on the proposed framework proposed in the previous section. First is to introduce six QoS criteria identified for Web service selection. Next is to propose a TQoS model composed of 3 aspects: service provider advertisement, SR agent and SC agent. Then normalization of QoS data, Decision model and Trust correction are provided.

4.1. QoS Criteria for Web service

According to the generality and importance to Web service selection, six QoS criteria for Web service selection is identified, which are execution time, cost, availability, reliability, reputation, and capacity.

- Execution time: Execution time Qtime(s) is the time duration (or turn-around time) from service request to response. It is composed of transmission time and processing time.
- Cost: Cost Qcost(s) is the amount of money the users or clients must pay to the providers for the usage of the Web services.
- Availability: Availability Qav(s) is the ratio of available time to total time. Total time is composed of available time and down time. So Qav(s) =1-down time/ total time.
- Reliability: Reliability Qrel(s) is the ratio of success requests to all the requests. Here success means that consumer can get the correct information they need. So Qrel(s) = successful requests/ total requests.
- Reputation: Reputation Qrep(s) is the scale value (such as value from 1 to 10) to indicate public's opinion on quality of services as well as the credits of service providers.

- Capacity: Capacity Qcap(s) is the number of concurrent users who can use the services. Capacity of Web services has important impact on execution time and reliability [2].

4.2. TQoS Model

Based on the proposed Web service selection framework, for each service (s), QoS data comes from 3 parties: service provider advertisement, service registry agent and service consumer agent. Therefore a TQoS model is composed of these 3 aspects:

$$TQoS\ (s) = <SP\ (s),\ SR\ (s),\ SC\ (s)>$$

Where SP represents service provider advertisement, SR represents service registry agent, SC represents service consumer agent.

1) SP advertisement: For each service, service provider provides quality information on execution time, cost, availability, reliability, and capacity of the service through extended WSDL or SLA. Reputation can be service providers or companies' own opinion about their reputation. QoS from service provider is modeled as follows:

$$SP(s)=< QP_{time,}\ QP_{cost}\ ,QP_{av,}\ QP_{rel,},\ QP_{rep},\ QP_{cap}>$$

2) SR agent: For each service, Service Registry agent can test service qualities objectively and automatically. Quality data such as execution time, availability, reliability, and capacity can come from Registry's testing, while reputation is the opinion value given by service registry and cost is from Registry's survey on related information about Web service cost. QoS from QoS Registry is modeled as follows:

$$SR(s)=< QR_{time,}\ QR_{cost},\ QR_{av,}\ QR_{rel,}\ QR_{rep},\ QR_{cap}>$$

3) SC agent: For each service, service consumer agent can monitor QoS data of execution time, availability, reliability, capacity, and cost. Reputation is the value given by service consumer according to their usage experiences of Web services. QoS from service consumer side agent is modeled as follows:

$$SC(s)=<QC_{time,}\ QC_{cost},\ QC_{av,}\ QC_{rel},\ QC_{rep},\ QC_{cap}>$$

Here, QCtime, QCcost, QCav, QCrel, QCrep, QCcap are the average values of quality data obtained from all the service consumers.

For example,

$$QC_{time} = \sum_{i=1}^{N} QC^{i}_{time}/N \qquad \text{(N is total number of service consumers.)}$$

4.3. QoS Data Normalization

Each quality data has its own unit. For instance, the unit of execution time can be millisecond, while the unit of cost can be dollars. Normalization of QoS data is needed to standardize these values into one unit. One popular methodology for normalization is to standardize them between 0 to 1 [9]. All the later calculations are based on the normalized values of quality data. Here, QoS data is divided into two types:

- Benefit criteria: the higher the value, the higher the quality
- Cost criteria: the higher the value, the lower the quality

In the proposed QoS model, quality criteria such as execution time and cost are cost criteria, while other quality criteria are benefit criteria.

The following formula is used to normalize QoS matrix A=$[aij]_{m*n}$ into A'=$[a'ij]_{m*n}$.

(1) Cost criteria

$$a'ij = \frac{\max(aj) - aij}{\max(aj) - \min(aj)} \quad \max(aj) \mathrel{!=} \min(aj)$$

$$a'ij = 1 \quad \max(aj) = \min(aj)$$

(2) Benefit criteria

$$a'ij = \frac{aij - \min(aj)}{\max(aj) - \min(aj)} \quad \max(aj) \mathrel{!=} \min(aj)$$

$$a'ij = 1 \quad \max(aj) = \min(aj)$$

4.4. Decision Model

According to QoS model in the previous section, we build our decision model based on Multiple Attribute Decision Making (MADM). Weights methods are used to reflect the quality criteria's relative importance to service consumers. The decision model is as follows:

$$\text{Score} = Wp * Score_{SP} + Wr * Score_{SR} + Wc * Score_{SC} + \delta\, trust$$

$$Score_{SP} = \sum_{i=1}^{6} Wi * QPi; \quad Score_{SR} = \sum_{j=1}^{6} Wj * QRj;$$

$$Score_{SC} = \sum_{k=1}^{6} Wk * QCk$$

Here Wp, Wr, and Wc are relative weights of SP advertisement, SR agent and SC agent to Web service selection, and Wp +Wr+Wc=1. Wi, Wj, Wk are the corresponding weight of quality criteria in SP advertisement, SR agent and SC agent. QPi, QRj, QCk are normalized quality values of each criteria. δ trust stands for trust correction, which is described in detail in next section.

4.5. Trust Correction

Trust is different from reputation. Trust is an objective value to illustrate the confidence level on QoS information, which can be calculated according to the similarity of the QoS promised and QoS received, while reputation is public's opinion about the quality criteria of services and it is collective evaluation of a group of consumers or users [6].

Here it is assumed that trust is based on the QoS promised by providers, QoS tested by registry, and QoS received by consumers. A Web service is trustful, if QoS values received by consumers and tested by registry are no less than QoS values promised by providers. We also assume service consumer trust service registry on QoS information. So we model trust correction in two parts:

- Trust between service provider and registry
- Trust between service provider and service consumer

Since the quality values are already normalized between 0 and 1, the trust correction is modeled as follows:

$$\delta\, trust = (\delta(p,r) + \delta(p,c))/2$$
$$= \left(\left(\sum_{i=1}^{2} (QP^{i} - QR^{i}) + \sum_{i=3}^{6} (QR^{i} - QP^{i}) \right)/6 + \left(\sum_{i=1}^{2} (QP^{i} - QC^{i}) + \sum_{i=3}^{6} (QC^{i} - QP^{i}) \right)/6 \right)/2$$

Here p stands for provider, r stands for registry and c stands for consumer.

The following shows some interpretations about the effects of trust correction on Web service selection:

- According to the trust correction formula, Web services with better real QoS data will have higher possibility to be selected.
- Web services with low advertised QoS data will have higher possibility to be selected. Therefore this modeling can avoid service providers to exaggerate Web services' performance.
- Service providers will not try to lower their advertised QoS, because decision model will be used for final decision.

From this modelling, Web services which have positive QoS performance (obtained QoS no less than advertised QoS) will have high possibility to be selected.

5 Implementation

This section is to describe a simple prototype of the proposed framework and a scenario. First is to introduce the implementation of SR agent, SC agent, and QoS enhanced Registry. Next is to describe a scenario of a Tour agency's Web service selection and its implementation. After that, the whole picture of implementation is presented, including the framework implementation and the scenario implementation.

5.1. Framework Implementation

A prototype of the proposed framework was implemented, including service provider, service consumer, SR agent, SC agent and QoS enhanced Registry. The whole framework is implemented in the following environment:

- Pentium 4 CPU 3G
- 1.49G RAM.
- Visual Studio .NET for developing Web services
- SQL Server Data Base for saving QoS data
- IIS for deploying Web services

5.1.1 Service Test in SR agent

Each registered Web service is implemented with an agent on service Registry to test QoS information. Here, test cases are provided by service consumers. Detail information about implementation of SR agent for QoS data testing and collection is presented in Table 1.

Table. 1 QoS data test and collection in SR agent

QoS Criteria	**Data collection**
Execution time	Execute Web service test cases to calculate differences between requests start time and requests end time, and save it to database.
*Cost	According to survey of service provider or other materials to find out the price of Web services.
Availability	Access Web services randomly to check their availability and save it to database. If Web service is unavailable, access Web service every 3 minutes until Web service is available. Calculate this time difference.
Reliability	Calculate the ratio of number of successful requests to number of all requests according to successful requests in database.
*Reputation	The 1-10 scale value given by service Registry objectively according to their opinion about Web service and service provider.
Capacity	According to QP_{cap} $(=n)$ provided by service provider, simulate n service users to request Web service concurrently. If all requests are returned successfully in reasonable time, that capacity is the tested capacity. Otherwise, repeat simulation with number of concurrent users (n-1).

* Cost and reputation are manually collected (by Registry), while other quality data are collected automatically.

5.1.2 Service Monitor in SC agent

Each service consumer is implemented with an agent to collect QoS information of a Web service when the Web service is requested. Detail information about implementation of SC agent for QoS data collection is presented in Table 2.

Table 2. QoS data monitor and collection in SC agent

QoS criteria	**Data collection**
Execution time	The average time difference from request sent to request received.
Cost	Calculate the average cost for requesting a Web service.
Availability	Calculate the average ratio of number of responded requests to number of all requests. (All the responded requests have *end time* in DB).
Reliability	Calculate the average ratio of number of successful requests to number of all requests.
*Reputation	Calculate average of 1-10 scale value given by service consumers according to their experiences of usages.
Capability	By Boxplot theory, if execution time of a request is larger than 1.5 Inter Quartile Range (IQR), the number of concurrent requests at that time is the capacity. Otherwise, the largest number of concurrent users is capacity.

* Reputation are manually collected (by Consumer), while other quality data are collected or calculated automatically.

5.1.3 Web service Selection in QoS Enhanced Registry

As mentioned in Section 3.2, in QoS enhanced Registry, there are 2 main functions: selecting optimal Web services, and collecting QoS data of Web services and calculating QoS score.

Optimal Web services selection is performed according to the proposed Trust model. After filtering according to customers' quality requirements, based on decision model, services with the highest score will be selected.

If newly registered or updated Web services are monitored, SR agent will test QoS data and send QoS data to QoS enhanced Registry. As to collecting QoS data from SC agent, in order not to overload QoS enhanced Registry, QoS data of Web services are collected on a daily basis. After collecting QoS data, QoS scores of Web services will be calculated. If QoS data of a Web service is changed, QoS score of that Web service will be re-calculated according to decision model of the proposed Trust model.

Web services are selected as the following steps:

- First step is to get the QoS data for each discovered Web services.
- The next step is to filter Web services according to consumers' QoS requirements.
- Final step is to rank the filtered Web services and selected the Web service with the highest score.

5.2. Experimental Scenario

This section is to describe about the implemented scenario for Web service selection. First we describe the scenario. Then we analyze the Web service selection for this scenario with collected QoS data.

5.2.1 A Scenario and its Execution

Let's assume that there are four kinds of services included in a certain business process of Tour agencies: Tour service, Ticket service, Hotel booking service and Payment service, as shown in Figure 2. These services are sequential connected and work as flows. The whole workflow is executed as follows. According to customers' request, first select tour service, and then according to selected tour information, such as location, select ticket service to get the arrival time. Then select hotel booking services to book hotel based on arrival time. Finally select payment service to pay for tickets and hotel fee. Instead of functional matching, here selecting each kind of services according to non-functional requirements is focused on.

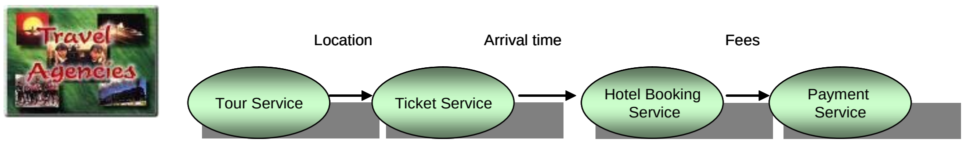


Figure 2. Business process of a tour agency

We implemented 10 Web services for each kind of Web services in the business process. As for Web service selection, initially Web services are selected according to QoS data from service provider advertisements. With the increase of historical data about Web services, Web services are selected according to QoS data not only from advertisements, but also from SR agent and SC agent. During the simulations, it is found that QoS data varies dynamically according to network condition and resource allocation, especially for execution time. Average value of each criteria for each service is calculated. Here, ticket services are used as an example and historical data of ticket services in QoS Registry are shown in the following Table 3.

Table 3. QoS data and score of 10 ticket services with similar functionality

service name	QPtime, QPcost, QPav, QPrel, QPrep, QPcap						QRtime, QRcost, QRav, QRrel,QRrep, QRcap						QCtime, QCcost, QCav, QCrel, QCrep,QCcap						Trust	Total Score
tickets1	65	0.14	0.92	0.9	10	50	165	0.14	0.88	0.8	9	15	175	0.14	0.82	0.8	9	10	-0.1059	0.5374
tickets2	40	0.2	0.98	0.95	9	25	140	0.2	0.88	0.85	8	20	150	0.2	0.84	0.9	8	5	-0.0247	0.6057
tickets3	40	0.16	0.96	0.88	9	50	155	0.16	0.88	0.88	8	50	165	0.16	0.96	0.9	9	30	0.054	0.7703
tickets4	40	0.15	0.95	0.85	8	25	142	0.15	0.88	0.8	7	25	152	0.15	0.85	0.8	8	4	-0.0058	0.6483
tickets5	45	0.18	0.98	0.92	7	40	155	0.18	0.88	0.82	6	40	165	0.18	0.87	0.8	7	8	-0.0178	0.6193
tickets6	55	0.15	0.88	0.9	9	30	165	0.15	0.88	0.8	8	15	175	0.15	0.68	0.8	8	6	-0.0505	0.5527
tickets7	70	0.15	0.92	0.94	8	45	180	0.15	0.88	0.84	7	30	190	0.15	0.72	0.7	6	9	-0.0899	0.5212
tickets8	38	0.1	0.91	0.92	8	35	148	0.1	0.88	0.82	7	35	158	0.1	0.81	0.8	8	20	0.0257	0.7395
tickets9	50	0.18	0.92	0.92	10	40	150	0.18	0.88	0.84	9	40	160	0.18	0.92	0.9	10	30	0.0459	0.7417
tickets10	25	0.15	0.88	0.92	10	35	135	0.15	0.88	0.82	9	35	145	0.15	0.78	0.8	7	7	-0.0161	0.6614

Service Provider **Service Registry** **Service Consumer**

5.2.2 Analysis of Web services selection in Scenario

By adopting the Trust model proposed, the trust score as well as total score of QoS for each service is calculated, which are also shown in Table 3. Here, 95% confidence interval is used to calculate trust scores. Wp, Wr, and Wc equal to 0.20, 0.45, and 0.35 respectively. For the quality criteria in each aspect, the corresponding weights equal to [0.3,0.2,0.12,0.18,0.1,0.1] respectively.

According to the calculated data, Web services are ranked according to the total scores. Symbol "s1>s2" means service 1 is better than service 2 for selection. Based on total score, 10 services are ranked as follows: tickets3> tickets9 > tickets8 > tickets10 > tickets4 > tickets5 > tickets2> tickets6 > tickets1 > tickets7. Due to the high trust score of 'tickets3', it has highest total scores and will be selected. If only considering QoS data from service provider, 'tickets10' will be selected, which is actually not as good as 'tickets3' in terms of quality.

5.3. Whole Picture of Implementation

The whole implementation includes the proposed Web service selection framework and the scenario, as shown in Figure 3. Service providers provide 10 Tour services, 10 Ticket services, 10 Hotel services and 10 Payment services, which have similar functionality but different QoS respectively. The following shows a simplified process of Web service selection:

1) QoS enhanced Service Registry collects QoS data of these 40 Web services, and QoS data is tested by SR agent.

2) As a service consumer, the Tour agency selects Web services dynamically according to its business process and QoS data in QoS enhanced Service Registry.

3) After selecting all the services needed, Web services are composed and executed.

4) Then QoS data from SC agent is uploaded to QoS Registry.

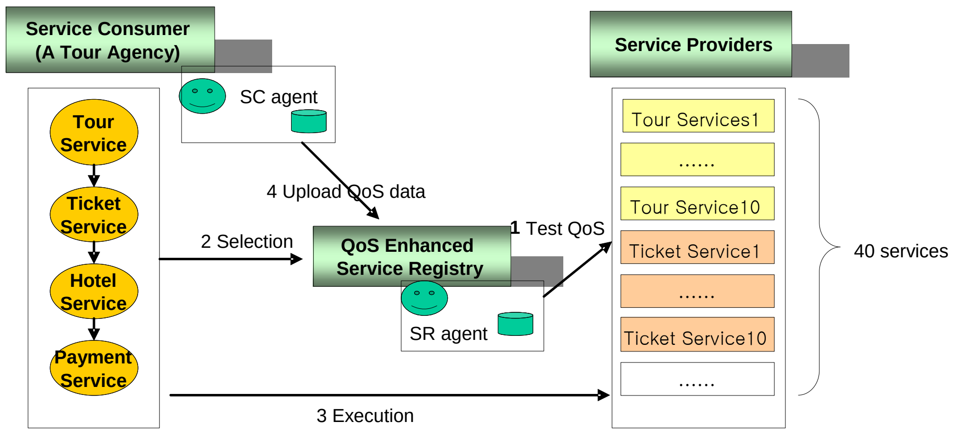


Figure 3. Whole picture of implemented framework with scenario

6 Validation

In order to evaluate the effectiveness of the proposed approach, it is compared with Euclid approach, and Fuzzy approach. Comparisons are based on 3 aspects: their concepts on Web service selection, their effects on Web service ranking and their effects on final obtained execution time, cost, availability, etc. For this validation, 1000 Web services in the scenario are simulated. Results show that according to proposed approach, services that do not provide trustful QoS performance are less likely to be selected than those that provide trustful QoS performance to consumers.

6.1 Euclid QoS model and Fuzzy QoS Model

As described in related works, Euclid approach is to find the nearest wsi to the QoS specifications of the consumer, (wsi means one of a set of Web services WS that have the same functional properties, WS = {ws1, ws2, ..., wsn},where i is the number between 1 and n) [7]. Web services with the minimum Euclidian distance will be selected. The Euclidean distance measure for evaluating the similarity between two vectors ti=(ti1,ti2…tis) and tj=(tj1,tj2…tjs) is calculated as follows:

$$\text{Dis }(ti,tj)= \sqrt{\sum_{h=1}^{s}(tih-tjh)\hat{}2}$$

Dis(ti, tj) stands for the distance between ti and tj.

As described in related works, Fuzzy approach is used to select the most appropriate Web service with the highest degree of membership belonging to the positive ideal solution as shown in the following g [13]. The membership function u(si) is defined as follows:

$$u(\text{si }) = \frac{1}{1+(dig/dib)\hat{}2}$$

Where:

$$dig=\sqrt{\sum_{j=1}^{5}[wj(gj-a'ij)]\hat{}2}\ ;\ dib=\sqrt{\sum_{j=1}^{5}[wj(a'ij-bj)]\hat{}2}$$

g= (g1,g2,g3,g4,g5)= (max(a’i1),max(a’i2), max(a’i3),max(a’i4),max(a’i5));
b= (b1,b2,b3,b4,b5)= (min(a’i1),min(a’i2), min(a’i3),min(a’i4),min(a’i5));

Where, *wj* represents the weight of jth quality criterion.

a’ij represents the normalized QoS value, where i represents the ith web service, j represent the jth quality criteria.

6.2. Comparison based on Concept

The comparison of proposed approach with other approaches based on concepts is shown in Table 4. These models are different in key idea and service selection algorithm.

Table 4. Comparison of proposed approach with other approaches on concept

Approaches	**Key Idea**	**Service Selection Algorithm**
Proposed approach	Build Trust model and consider 3 sources of quality data and trust corrections.	According to decision model, Web services with **highest** score will be selected.
Euclid approach	Find the nearest wsi to the QoS specifications of the consumer.	Web services with the **minimum** Euclidian distance will be selected.
Fuzzy approach	Identify the most appropriate Web service with the highest degree of membership belonging to the positive ideal solution.	Web services with **highest** utility values will be selected.

6.3 Comparison based on Ranking

According to each model, 10 Web services in the example scenario about ticket service selection are scored and ranked differently, as shown in Table 5 and Figure 4.

Table 5: Comparison of proposed approach with other QoS models on ranking

Approaches	**Scores(from s1 to s10)**	**Ranking**
Proposed approach	0.5374 0.6057 0.7703 0.6483 0.6193 0.5527 0.5212 0.7395 0.7417 0.6614	s3> s9 > s8 > s10 > s4 > s5 > s2> s6 > s1 > s7.
Euclid approach	0.4944 0.7120 0.6445 0.6549 0.6765 0.5617 0.4931 0.6570 0.6239 0.7282	s7 >s1 > s6 > s9 > s3 > s4> s8 > s5 > s2 > s10.
Fuzzy approach	0.1576 0.5862 0.8675 0.7186 0.4456 0.2765 0.0576 0.7181 0.3753 0.9511	s10 > s3> s4 > s8 > s2 > s5> s9 > s6 > s1> s7.

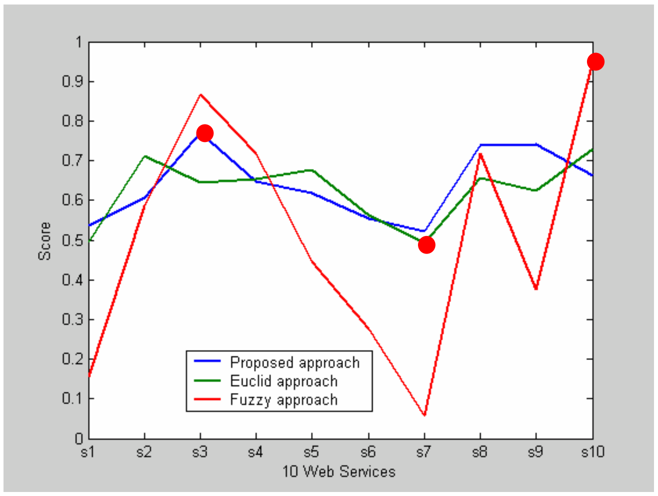


Figure 4. Comparison of proposed approach with other approaches on ranking

From Figure 4, it is found out each approach has its own preference for Web service selection. By Euclid approach, ‘s7’ is the most close to user requirements and will be selected, although it is not high quality according to proposed approach. By Fuzzy approach, ‘s10’ will be selected. In summary, each model shows its own preference in Web service selection which leads to different selection results. More validation about the effectiveness is needed to show which model is better.

6.4 Comparison based on Obtained Quality Values

In order to clearly illustrate the effectives of each approach, experiments were done to measure the obtained quality values like execution time, cost, availability, reliability, reputation and capability. When concurrent consumers submit their service requirements for a task in our Web service selection scenario, Euclidean approach, Fuzzy approach and our proposed approach are applied for services selection. The average quality of all services assigned to concurrent requesters (consumers) by different approaches is adopted for evaluating the performance of each decision making algorithm. The number of concurrent requesters is from 50 to 500 and the experimental results are shown in Figure 5.

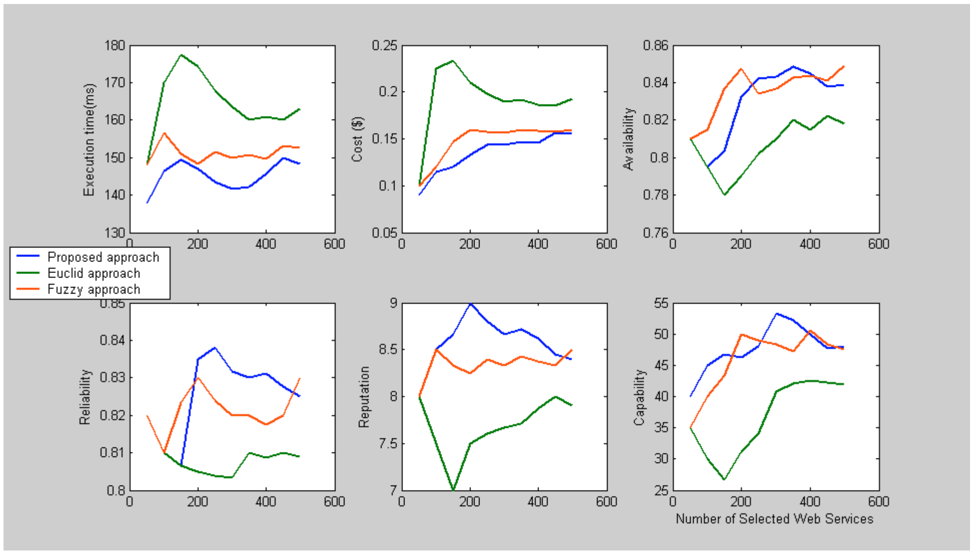


Figure 5. Comparison of proposed approach with other QoS models on obtained quality values

From Figure 5, the following analysis and conclusion can be reached:

1) Euclidean approach (green line) perform worst

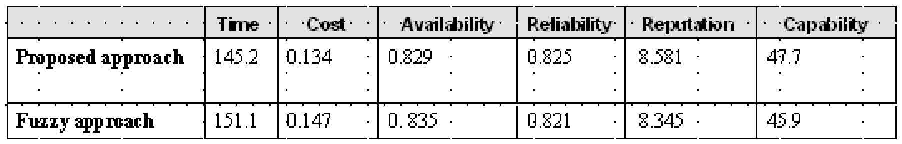
This may be because of Euclidean approach focus on finding QoS closest to QoS requirements instead of Web service with the best quality values.

2) Similar performance of proposed approach (blue line) and Fuzzy approach (red line)

Figure 5 shows that proposed approach and Fuzzy approach are close together and have the similar trend. This may be because both approaches focus on finding positive solutions and try to identify service with best performance.

Total average QoS score according to proposed approach and Fuzzy approach is shown in Table 6. From this table, we find out that proposed approach shows a higher performance than Fuzzy approach in terms of obtained quality values.

Table 6. Comparison with fuzzy approach on obtained quality values



	Time	Cost	Availability	Reliability	Reputation	Capability
Proposed approach	145.2	0.134	0.829	0.825	8.581	47.7
Fuzzy approach	151.1	0.147	0. 835	0.821	8.345	45.9

7 Conclusions and Future Work

Web services selection plays an important role for service composition and business process work flow. It can help service consumers to identify the best services in terms of quality.

Many researchers proposed QoS models with quality criteria and selection algorithm to optimize service selection. However, currently sources of these quality values for Web service selection are not trustable, and trust degree to service providers is still not well modeled. In dynamical Web environment, Web services should be selected dynamically (run time) instead of statically (design time).

Motivated by these problems, in this paper, first a QoS enhanced framework for Web service selection is presented. Based on this framework, a Trust model is proposed, which is composed of TQoS model, Decision model and Trust correction. It is claimed that a Web service can be regarded as trustful if QoS values received by consumers and tested by registry are no less than QoS values promised by providers. The proposed framework is implemented, including SC agent, SR agent and QoS Registry. A scenario about a Tour agency's Web service selection according to its business process is also implemented. To validate effectiveness of proposed framework and Trust model, proposed approach is compared with other approaches for Web service selection, such as Euclid approach and Fuzzy approach. Numerical simulations are adopted for the comparisons. Results show that Fuzzy approach has similar performance with proposed approach, however proposed approach performances a little better other approaches in terms of obtained quality values.

As for the future work, we plan to implement GUI for proposed QoS enhanced Registry and agents to make the approach more complete. Since we did the implementation on a local machine, including Web services and service consumers, more researches are needed to be done on the published online Web services. The implementation of QoS enhanced Registry still needs further research. Only a simple QoS enhanced Registry for Web service selection is implemented. Other functions of QoS enhanced Registry need further research and implantation. For example, as for dynamic Web service selection, automatic test case generation according to WSDL should be included in QoS enhanced Registry. Another future work is to use Web Services Search Engine [27] to replace registry, which we hope can bring better research results.

Acknowledgements

This research was supported by the MKE(Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement)" (IITA-2008-(C1090-0801-0032))

References

1. N. Eric, L. Greg, understanding SOA with Web services (Upper Saddle River, NJ: Addison Wesley, 2005).
2. M. Serhani, R. Dssouli, A. Hafid, H.Sahraoui, A QoS Registry based architecture for efficient Web service selection, Proc. IEEE Conf. on Web service, 2005.
3. V. Cardellini, E. Casalicchio, V. Grassi, R. Mirandola, A Framework for Optimal Service Selection in Registry-based Architectures with Multiple QoS classes, Proc. IEEE Conf. on Services Computing Workshops, 2006.
4. Z. Gao, G. Wu, Combing QoS-based service selection with performance prediction, Proc. IEEE Conf. on e-Business Engineering, 2005.
5. S. Kalepu, S. Krishnaswamy, S. Loke, Verity: A QoS metric for selecting Web services and providers, Proc. IEEE Conf. on Web information Systems Engineering Workshop, 2004.
6. L.H. Vu, M. Hauswirth, K. Aberer, QoS-based service selection and ranking with trust and reputation management, Proc. OTM'05, R. Meersman and Z. Tari (Eds.), LNCS 3760, p.p. 466-483, 2005.
7. L. Taher, R. Basha, H. Khatib, Establishing Association between QoS properties in Service Oriented Architecture, IEEE(NWeSP), 2005.
8. Janarbek ,Learning-based Trust Model for optimized Web services selection, Master thesis, Information and Communications University, 2007.
9. J. Hu, C. Guo, H. Wang, P. Zou, Quality Driven Web services Selection, Proc. IEEE Conf. on e-Business Engineering, 2005.
10. L. Yang, Y. Dai, B. Zhang, Y. Gao, Dynamic selection of composite Web services based on a genetic algorithm optimized new structured neural network, Proc. IEEE Conf. on Cyberworlds, 2005.
11. G. Yeom, T. Yun, D. Min, A QoS model and testing mechanism for quality-driven Web service selection, Proc. IEEE workshop on Software Technologies for Future Embedded and Ubiquitous Systems, 2006.
12. M. Jaeger, G. Muhl, S. Golze, QoS-Aware composition of Web services: an evaluation of selection algorithms, Springer-Verlag Berlin Heidelberg, LNCS 3760, 2005, 646-661.
13. H. Tong, S. Zhang, A fuzzy multi-attribute decision making algorithm for Web services selection based on QoS, Proc. IEEE Asia-Pacific Conference on Services Computing, 2006.
14. H. Chua, S. M.F.D Syed Mustapha, Web services Selection based on Multiple-Aspect Similarity Function, IEEE/WIC/ACM International conference on WIIATW, 2006.
15. M. Sensoy, P. Yolum, A Context-Aware Approach For Service Selection Using Ontologies, AAMAS, 2006.
16. E. Michael Maximilien, M. P. Singh, A Framework and Ontology for Dynamic Web service Selection, IEEE Internet Computing, 2004.
17. X. Bai, W. Dong, WSDL-Based Automatic Test Case Generation for Web Services Testing, International Workshop on Service-Oriented System Engineering, SOSE, 2005.
18. H. M. Sneed, S. Huang, WSDLTest-A Tool for Testing Web Services, Eighth IEEE International Symposium on Web Site Evolution (WSE), 2006.

19. R. Kugyte, L. Sliburyte, A standardized model of service provider selection criteria for different service types: a consumer-oriented approach, ISSN 1392-2785 Engineering Economics, No 3 (43), 2005.
20. Z., J. Baik, "QoS Broker-Based Trust Model for Effective Web Service Selection", IASTED SEA 2007, Cambridge, Massachusetts, USA, Nov. 19–21, 2007.
21. WSDL: http://www.w3.org/TR/wsdl
22. SOAP: http://www.w3.org/TR/soap/
23. UDDI: http://www.uddi.org
24. SLA: http://www.research.ibm.com/wsla/
25. BPEL: http://en.wikipedia.org/wiki/Business_Process_Execution_Language.
26. SOAP pinging: http://www.jeckle.de/freeStuff/soaping/index.html
27. Web Services Search Engine : http://www.seekda.com