2012 IEEE Ninth International Conference on Services Computing

Dynamic Service Selection Based on Context-Aware QoS

Donghui Lin, Chunqi Shi, and Toru Ishida
Department of Social Informatics, Kyoto University
Yoshida-Honmachi, Sakyo-ku, 606-8501, Kyoto, Japan
Email: {lindh, ishida}@i.kyoto-u.ac.jp, shi@ai.soc.i.kyoto-u.ac.jp

***Abstract*—With the development of service-oriented computing environments, QoS-aware service selection has been a more and more important research issue. In service composition environments, QoS attributes of atomic services are always aggregated for computing the QoS of the composite services, which has been reported in many previous studies. However, there are situations that some QoS attributes cannot be aggregated for composite services. For example, it is difficult to compute the translation quality of a composite translation service by simply aggregating its component atomic services (machine translation service, morphological analysis service, dictionary service). Moreover, when multiple QoS attributes are used for evaluating services, it is always difficult to maximize all the QoS attributes because there might be anti-correlated relations between them. To address above problems, this paper proposes an approach for selecting services based on context-aware factors of QoS attributes. In our proposed approach, context-aware factors that affect QoS attributes are first extracted from analyzing their correlation with QoS attributes. Then, QoS data is generated based on the extracted factors for QoS prediction and evaluation. Further, dynamic service selection is realized based on QoS prediction and evaluation considering user requirements. We use a case study in the domain of language service with some experiments to show the effectiveness of our approach.**

***Keywords*-service selection; quality of service; user-centered; composite service; QoS prediction; context-aware**

I. Introduction

Service-oriented environments have become more and more important in recent years, where various kinds of Web services and service-based processes are gathered within a certain domain or across domains. Such environments enable people to create, manage their own services, and share their services with each other, while users can get additional value of services by composing them based on their own requirements. With the development of service-oriented environments, Quality-of-Service (QoS) is usually employed for describing general characteristics of Web services [1]. In service composition environments, QoS attributes of atomic services are always aggregated for computing the QoS of the composite services, which has been reported in some previous work, which can be used for service selection [2] [3] [4] [5].

QoS-aware service selection is always defined as a QoS optimization problem, where atomic services that generate maximum overall QoS value with constraints are selected as optimized solution [1] [6] [7] [8]. In most previous approaches, it is assumed that QoS information is pre-existing and QoS attributes of atomic services are aggregatable. Therefore, QoS-aware service selection is based on the declared QoS values.

However, the above approaches have several limitations considering the following problems in the real service composition environments. First, there are situations that some QoS attributes cannot be aggregated for composite services. For example, it is difficult to compute the translation quality of a composite translation service by simply aggregate its component atomic service (machine translation service, morphological analysis service, dictionary service, etc.) Second, QoS values always vary based on different context for different types of service invocation. Third, when there are multiple QoS attributes for services, it is always difficult to maximize all the QoS attributes because there might always be anti-correlated relations between them.

To address the above issues, QoS prediction are proposed in some work including the approaches of usage of history data, user experience and so on [9] [10]. However, context information acquired from service invocation is important for QoS prediction. Moreover, user requirement should be considered for service selection based on QoS prediction. Those points are rarely covered in previous work.

In this paper, we propose an approach for selecting services based on context-aware QoS. Context-aware information is very important in service selection. For example, service response time may vary with the different distance between service user and service entity, or with the different invocation timing. Some examples with more details will be given in Section II based on experiences and observations of operating a service composition infrastructure in the language service domain, the Language Grid [11]. Therefore, it is necessary to consider context factors when modeling QoS attributes, which we will propose in this paper. Moreover, we propose an approach of service selection based on the context-aware QoS model, where user requirement is considered. The proposed service selection approach is expected to be evaluated based on experiments in service composition platforms.

The rest of the paper is organized as follows: Section II provides a motivation example on the language service domain. In Section III, we propose a QoS model with multiple attributes, where each QoS attribute is affected by

978-0-7695-4753-4/12 $26.00 © 2012 IEEE
DOI 10.1109/SCC.2012.95

a series of context factors from user profile information, service request information, and so on. Section IV describes the service selection approach based on our proposed QoS model considering user requirements. Section V introduces a case study based on some experiments and analysis to show how our proposed approach selects services based on user requirements. Section VI introduces some related work from the perspectives of QoS-aware service selection and QoS prediction. Finally, the conclusion is presented in the last section.

II. Motivation Example

To show the importance of context information related to QoS attributes, we use an example of the QoS of language services on the Language Grid[1] in this section.

The Language Grid provides a service composition environment for users to share, create and combine Web services in language domains, which we call language services [11]. The infrastructure of the Language Grid is aimed at connecting two kinds of servers (core nodes and service nodes). Core nodes manage all requests to language services, while service nodes (or we call service entity hosts) actually invoke atomic services. If the requested service is a composite one, core nodes invoke a corresponding Web service workflow that includes multiple atomic services. Registered information of language services is shared among all core nodes. By April 2012, 145 organizations from 18 countries and regions have become users of the Language Grid, and over 170 atomic language services and composite services are provided on the Language Grid[2]. Besides, humans are also possible to be wrapped as Web services on the Language Grid. In the Language Grid, multiple QoS attributes are managed for language services, including both general attribute like response time and cost, and domain specific attribute like translation quality.

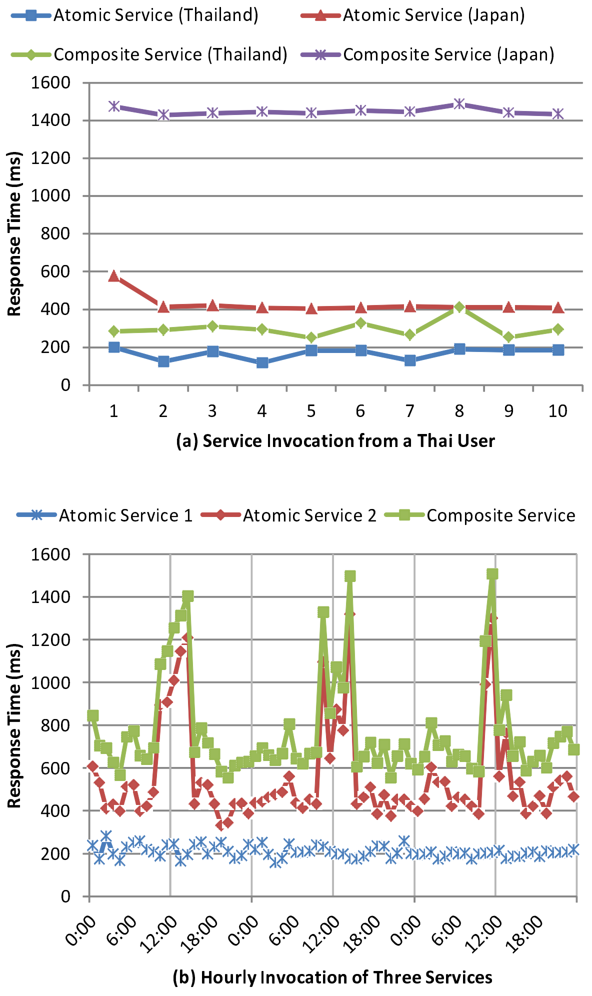


Figure 1. Response time of service invocation: (a) service invocation of services hosted on servers of different physical locations; (b) service invocation of different services hourly.

Figure 1 shows the example of how the context related factors affect response time during dynamic service invocation. Figure 1 (a) is the sample data of response time for service invocation by Language Grid user from Thailand, who invokes an atomic service (J-Server Japanese-English translation service) and a composite service (composite Japanese-English translation service combined with dictionary which is composed by three atomic services on the Language Grid: J-Server Japanese-English translation service, Mecab Japanese morphological analysis service and the Kyoto tourism Japanese-English dictionary). Each service is invoked for 10 times from both the service entity host in Thailand that is near to the user and the service entity host in Japan that is far from the user. The result shows that it takes 2.54 times and 4.87 times of response time for invoking an atomic service and a composite service respectively on the host in Japan than in Thailand for Thai user. Similar experiments have been conducted between Language Grid users in Japan and in Denmark. When invoking an atomic service (J-Server Japanese-English translation service) on the service entity host in Japan, user from Denmark costs 5.39 times of response time than user from Japan. Therefore, *distance of user and service entity* can be regarded as a context related factor for the QoS attribute *response time*. To optimize the QoS attribute of response time, dynamic service invocation control mechanisms were proposed in our previous work [12]. Figure 1 (b) is the sample data of response time for hourly service invocation for three days of three different services, including two atomic services (two translation services) and one composite service (Two-hop translation service which is composed by the above two atomic translation services). The result shows that the

[1] http://langrid.org/

[2] Lists of Language Grid users and services are available in the Language Grid Service Manager (http://langrid.org/service_manager/)

response time of one translation service is steady, while the response time of the other translation service and the composite service periodically appears peak time effect. Therefore, *time slot of the day* is possible to be regarded as a context related factor for the QoS attribute *response time*.

Figure 2 shows the example of how the context factors affect translation quality (adequacy) during dynamic service invocation[3]. In more details, we conducted three type of Japanese-English translation tasks (business documents, temple documents, and college documents). The service invocation number varies from 500 to 2,000 for the three tasks. We analyze the relation between text length and translation quality. The result shows that there is a decreasing trend of translation quality when translating long sentences. Therefore, *length of source text* can be regarded as one of the context factors for the QoS attribute *translation quality*.

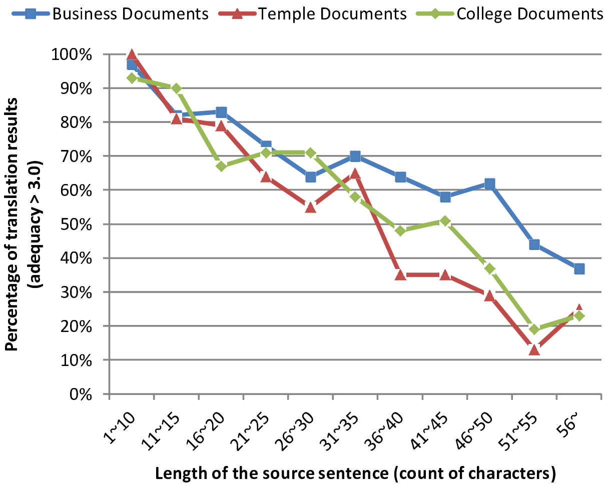


Figure 2. Relation between length of source text and translation quality.

These examples show that context factors are important for QoS attributes in service composition environments. Based on the statistical analysis and some previous work on machine translation quality estimation factors [14], we can get a dependency graph of context-aware factors (leaf nodes) and QoS attributes (response time, translation quality, and translation cost), a part of which is shown as Figure 3. In the following sections, we will show how we model the context-aware QoS attributes for service selection based on user requirements.

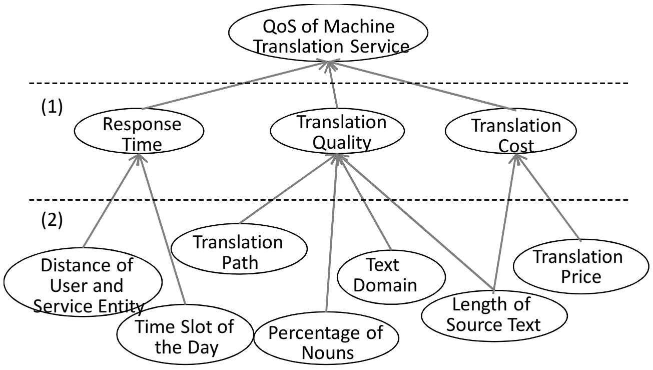


Figure 3. QoS of machine translation services: (1) QoS attributes, (2) context factors of QoS attributes.

III. Context-Aware QoS Model

In this section, the context-aware QoS model will be defined. First, we define several types of services: abstract atomic service, concrete atomic service, and composite service. Abstract atomic service $S_i = \{s_{i1}, s_{i2}, \ldots, s_{in}\}$ consists of all concrete atomic services $(s_{i1}, s_{i2}, \ldots, s_{in})$ that deliver the same functionality but differ in QoS attributes. Composite service $CS = (S_1, S_2, \ldots, S_n)$ is composed of several abstract atomic services that binds concrete atomic services during execution. Composite service can be regarded as a service-based workflow, which has the general characteristics of workflow. Therefore, many theorems and systems of workflow management area can be applied. For example, workflow patterns [15] proposed in previous research can also be used to design composite services. Invocation response time of a composite service can be gained from the aggregation of invocation response time of the atomic services that consist it. In this research, we use the aggregation approach based on workflow patterns [16].

A. QoS Parameters

In this research, we consider both generic QoS attributes like response time, cost, and domain-specific QoS attributes like translation quality for machine translation services. We use $Q_s = (q_1(s), q_2(s), \ldots, q_n(s))$ to represent the vector of QoS values of the service s, where $q_i(s)$ is the i^{th} QoS attribute of the service s.

Next, we use $D_i(s) = (d_{i1}(s), d_{i2}(s), \ldots, d_{in}(s))$ to represent the the vector of context factors that affect the QoS attribute $q_i(s)$, where $d_{ij}(s)$ is the j^{th} factor that affects $q_i(s)$. $q_i(s)$ can be regarded as a function of $D_i(s) = (d_{i1}(s), d_{i2}(s), \ldots, d_{in}(s))$, which is represented as $q_i(s) = X(d_{i1}(s), d_{i2}(s), \ldots, d_{in}(s))$.

$r(u, S, t)$ is the service invocation request from user u to abstract service S at time t.

Table I summarizes the notations and their descriptions used in the context-aware QoS model in the service com-

[3] In the language service domain, the domain specific QoS attributes (translation quality) are always more essential than other nonfunctional QoS attributes. Translations were evaluated on the basis of adequacy and fluency in previous reports [13].

position environments. Although above definitions of QoS attributes are for atomic services, they can also be applied for composite services.

Table I
LIST OF NOTATIONS FOR QOS PARAMETERS.

Notation	Description
S_i	abstract atomic service i
s_{ij}	concrete atomic service j for S_i
CS	composite service
u_i	service user i
Q_s	the vector of QoS values of the service s
$q_i(s)$	the i^{th} QoS attribute of the service s
$D_i(s)$	the vector of factors that affect the $q_i(s)$
$d_{ij}(s)$	the j^{th} factor that affects $q_i(s)$
$r(u, S, t)$	service request from user u to abstract service S at time t

B. Normalization of QoS Attributes

Since different QoS attributes have different computation methods, it is necessary to normalize different QoS attributes when multiple attributes are considered for service selection. In this research, the normalization uses the min-max Equation (1). $max(q_k(s))$ and $min(q_k(s))$ are the maximum value and minimum value of QoS attribute $q_k(s)$ that can be expected for all the services. If a QoS attribute is the type of gaining, it is called positive, e.g, the adequacy of translation result. Otherwise, the attribute is type of the paying, and it is called negative. e.g., the cost that should be paid for invoking services.

$$q'_k(s) = \begin{cases} \frac{q_k(s)-min(q_k(s))}{max(q_k(s))-min(q_k(s))} & \text{if attribute is positive} \\ \frac{max(q_k(s))-q_k(s)}{max(q_k(s))-min(q_k(s))} & \text{if attribute is negative} \end{cases} \tag{1}$$

C. User-Centered Evaluation

We define QoS requirements from a user as $C(s) = (c_1(s), c_2(s), \ldots, c_n(s))$, where $c_l(s)$ is the l^{th} QoS constraint from the user over service s. We use $P(s) = (p_1(s), p_2(s), \ldots, p_n(s))$ to define whether the QoS constraint is satisfied or not. For each QoS constraint $c_l(s)$, $p_l(s)$ is calculated as Equation (2).

$$p_l(s) = \begin{cases} 1 & \text{if } c_l(s) \text{ is satisfied} \\ 0 & \text{if } c_l(s) \text{ is not satisfied} \end{cases} \tag{2}$$

We use $w_k \in [0, 1]$ to define the weight of c_k that is decided by user priority of QoS constraints, where $\sum_{k=1}^{n} w_k = 1$.

To evaluate the user satisfaction, we use two types of utility functions: utility of feasibility $Utility_F(s)$, and utility of optimity $Utility_O(s)$.

Utility of feasibility is used to evaluate whether the service selection is feasible based on user requirements, and is calculated by Equation (3). $Utility_F(s) = 1$ means that all the QoS constraints from the user are satisfied. Otherwise, there is one or more constraints are not satisfied.

$$Utility_F(s) = \sum_{k=1}^{n} p_k(s) \cdot w_k \tag{3}$$

Utility of optimity is used to evaluate whether the service selection is optimal based on user requirements, and is calculated by Equation (4). $Utility_O(s)$ is meaningful only when $Utility_F(s) = 1$. $w'_k \in [0, 1]$ is the weight of $q_k(s)$ that is decided by user priority of QoS attributes, where $\sum_{k=1}^{n} w'_k = 1$.

$$Utility_O(s) = \sum_{k=1}^{n} q'_k(s) \cdot w'_k \tag{4}$$

In previous work, Alrifai *et al.* proposed several types of relations of different QoS attributes: independent, correlated, and anti-correlated [8], which is also useful in our research. In the independent type, the values of two QoS attributes are independent to each other, e.g., the QoS attributes of response time and translation quality in a translation service. In the correlated type, a service that is good in one attribute is also good in the other attribute. In the anti-correlated type, there is a clear trade-off between the two attributes, e.g., the QoS attributes of translation quality and cost in a translation service.

IV. SERVICE SELECTION BASED ON QOS PREDICTION AND USER REQUIREMENTS

In the previous section, we defined the context-aware QoS model for service selection. In this section, we will introduce the service selection approach based on QoS prediction and user requirements. Figure 4 shows the framework of context-aware service selection. For each service invocation request, user may decide the QoS requirements.

The framework can be mainly divided as four steps as follows.

1) *Context information generation*: once the service invocation request is received, context information (defined as $D_i(s)$ for each QoS attribute $q_i(s)$) will be generated based on user profile information, QoS data, and service invocation request information.
2) *QoS prediction*: QoS attributes of each concrete service in the group of abstract service will be predicted based on context information generated in the first step. There are several approaches for predicting QoS: average of past values, current value, linear model, time series forecasting, and so on [6], which might be applied for different situations respectively. We use the approach of computing average of past values in this work.
3) *Service selection*: After obtaining the predicted values for QoS attributes of all candidate services, service

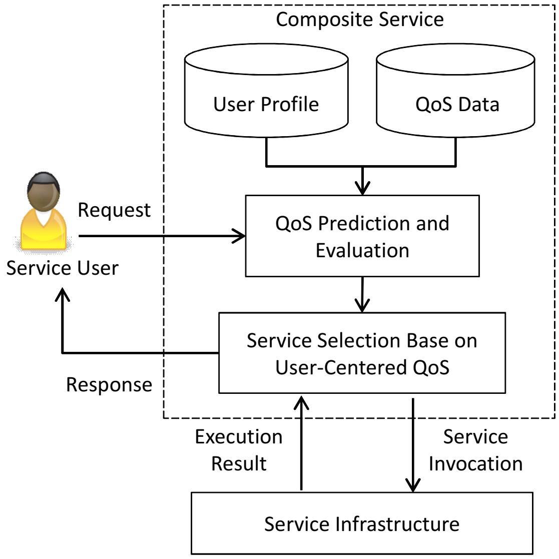


Figure 4. Context-aware service selection framework.

selection is conducted based on the utility of feasibility and utility of optimity that we have defined in section III. **Algorithm 1** shows how the selection process is conducted. Predicted utility of feasibility $Utility_F^p(s_i)$ of each candidate service is computed based on the values of the QoS attributes and user requirements. If the candidate is feasible ($Utility_F^p(s_i) = 1$), predicted utility of optimity $Utility_O^p(s_i)$ will be further computed. The candidate service with the largest utility of optimity will be selected.

4) *QoS evaluation*: When service is selected based on QoS prediction and user requirements, the service will be executed for the user. Then, the real values of QoS attributes will be obtained from the execution (some QoS attribute might need afterward evaluation), which will be used for updating the QoS prediction mechanism. **Algorithm 2** shows the QoS evaluation process.

Algorithm 1 Service Selection Based on QoS Prediction

```
procedure ServiceSelection ($r(u, S, t)$, $C$)
$s^*$ (selected service)
$MaxUtilityO \leftarrow 0$
/*Get the context data based on the invocation */
for all Concrete atomic service $s_i \in S$ do
    for all QoS attribute $q_j(s_i) \in Q_{s_i}$ do
        Get context $D_j(s_i) = (d_{j1}(s_i), \ldots, d_{jn}(s_i))$
        /*Get the prediction value of QoS attribute $q_j(s_i)$*/
        $q_j^p(s_i) \leftarrow X(D_j(s_i))$
    end for
    Compute predicted Utility of feasibility $Utility_F^p(s_i)$
    based on Equation (3)
    if $Utility_F^p(s_i) = 1$ then
        Compute predicted Utility of optimity $Utility_O^p(s_i)$
        based on Equation (4)
        if $Utility_O^p(s_i) > MaxUtilityO$ then
            $MaxUtilityO \leftarrow Utility_O^p(s_i)$
            $s^* \leftarrow s_i$
        end if
    end if
end for
return $s^*$
```

Algorithm 2 QoS Evaluation

```
procedure QoSEvaluation ($r(u, S, t)$, $C$)
/*Get selected service */
$s^*$ (selected service)
$s^* \leftarrow$ ServiceSelection ($r(u, S, t)$, $C$)
/*Update prediction value based on real QoS value */
Get real QoS value $Q_{s^*} = (q_1(s^*), \ldots, q_n(s^*))$
for all QoS attribute $q_i(s^*) \in Q_{s^*}$ do
    $k \leftarrow$ number of appearance times of value $D_i(s^*)$
    $X(D_i(s^*)) \leftarrow \frac{1}{k} \cdot (X(D_i(s^*)) \cdot (k-1) + q_i(s^*))$
end for
```

V. Experiment and Analysis

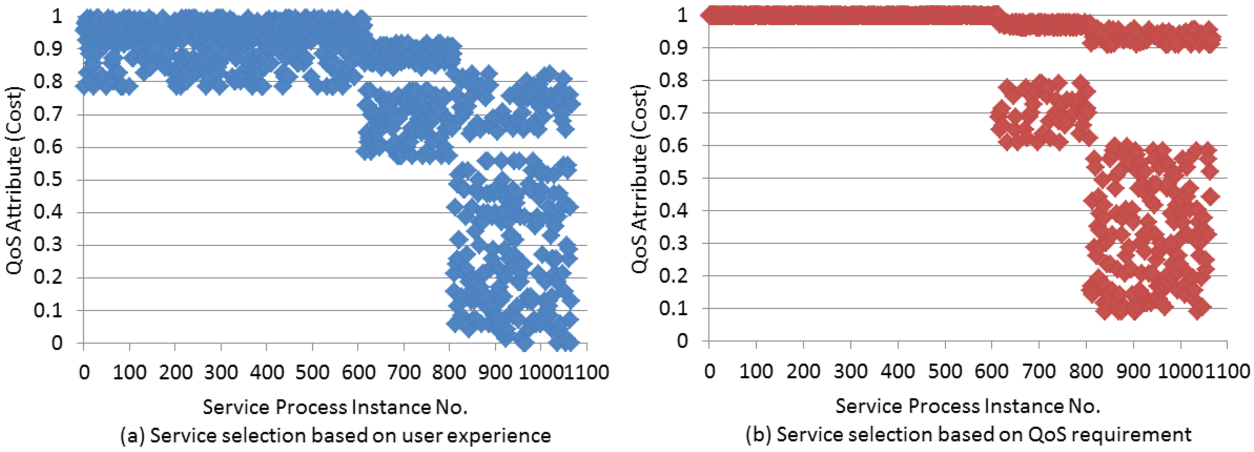
A. Experiment Settings

To observe and analyze how the proposed approach is effective for service selection, we conduct an experiment in the language service domain. The experiment settings are as follows.

1) *Task*: The task used in the experiment is to translate 1,068 sentences of college related document from Japanese to English, the length (number of Japanese characters) of which vary from 5 to 70 with the average length 20.

2) *Atomic Service / Composite Service*: Our experiments are based on several service-based Japanese-English translation processes: atomic machine translation service, composite translation service (composed by atomic machine service, morphological analysis service, and dictionary service), composite translation service combined with monolingual modification, composite translation service combined with monolingual modification and bilingual confirmation, and pure human translation process. The language services, composite Web services and human services are provided by the Language Grid.

3) *QoS Attributes*: QoS in the language service domain consists of non-functional QoS dimensions (cost, response time, etc.) and functional QoS dimensions (translation quality: fluency and adequacy). In this experiment, we mainly evaluate the translation cost and translation quality (adequacy) of the processes.



(a) Service selection based on user experience

(b) Service selection based on QoS requirement

Figure 5. Comparison of value of QoS attribute (Cost).

The reason why we choose to analysis these two QoS attribute lies in that they are of the anti-correlated type; there is a clear trade-off between the two attributes.

4) *User Requirement*: The user requirement for two QoS attributes is $q_{adequacy}(s) > 0.75$ and $q_{cost}(s) > 0.80$ respectively after normalization.

In this task, the price of a monolingual human service and bilingual human service is 10 times and 100 times of a Web service (machine translation, morphological analysis, dictionary). However, only bilingual human service is able to ensure the translation quality. Therefore, it is difficult for selecting service processes and Web services. We use our proposed approach for this problem by using the contextual dependency as shown in Figure 3.

B. Analysis and Discussion

To compare our proposed approach with the traditional approach, we conducted two different types of experiments. The first type is a real-world experiment, where service selection is based on user experience. During the experiment, the user uses two types of service processes: composite translation service combined with monolingual modification and bilingual confirmation, and pure human translation process. Since both processes include bilingual human service, the normalized value of translation quality (adequacy) is 1. The second type is a simulation experiment based on the data obtained from the first experiment and uses our proposed service selection approach, which select different atomic services and composite services based on the context information and users' QoS requirements.

Figure 5 shows the result of QoS attribute (cost) distribution for the two types of experiments, among which Figure 5 (b) is the result of the second experiment. The horizontal axis indicates the 1,068 instance, which is ordered by the length of sentences. From the result, we can see that the performance of QoS attribute (cost) has significant improvement for relatively short sentences. The reason is that machine translation services and composite translation services without human services are used frequently based on the QoS prediction and user QoS requirement. However, the cost for long sentences does not have significant changes with the first experiment. That is because composite translation services with human services are more effective to keep the translation quality for long sentences based on the context information. Figure 6 shows the result of QoS attribute (adequacy) distribution for the second experiment. The result shows that most of the normalized QoS attribute (adequacy) value is over 0.75. However, the performance is not so good for long sentences because the QoS adaptation considers the balance of the two QoS attributes.

We further analyze the average QoS values for the 1,068 instances. The QoS values in the first experiment is $q_{cost} = 0.77$ and $q_{adequacy} = 1.00$, while the QoS values in the second experiment is $q_{cost} = 0.86$ and $q_{adequacy} = 0.80$ respectively. The result shows that the QoS attribute (cost) does not meet the user requirement because the experiment focus too much on the QoS attribute (adequacy). In contrast, the average values of the two QoS attribute both meet the user requirement because our proposed approach considers both the feasibility and optimity of the QoS for service selection.

VI. Related Work

In this section, we introduce some related work from the perspectives of QoS-aware service selection and QoS prediction. Then, we show the features of our research by comparing our approach with the previous work.

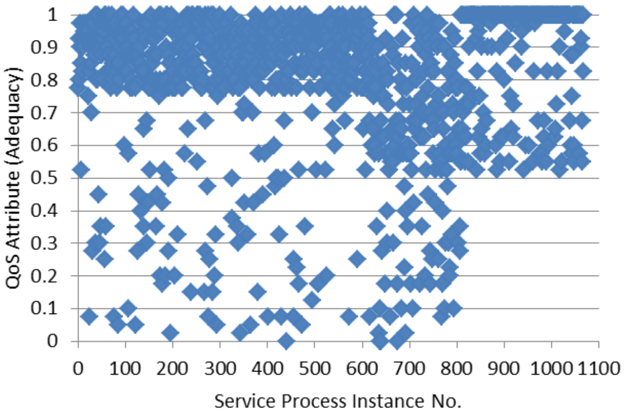


Figure 6. Value of QoS attribute (Adequacy) after applying service selection based on QoS requirement.

A. QoS-Aware Service Selection

The work of Zeng *et al.* [1] was among the earliest ones for QoS-aware service composition. The authors proposed a multidimensional QoS model for Web service composition and several optimization approaches for service selection in both static environment and dynamic environment. QoS aggregation was late used widely for service selection. Liu et al. described the detail computation process for QoS selection, including normalization, and weighted aggregation [2]. Wang et al. described not only the process of normalization, and weighed aggregation, but also the QoS ontology description [3]. The process of filtering Web service according to user preference was described by Cao et al [17]. Eyhab and Qusay suggested Web service relevancy function for QoS based discovery, which aggregates QoS properties after maximum normalization. Besides, it suggested that both QoS manager and Crawler are necessary in QoS based discovery. Huang et al. described the QoS aggregation for different workflow patterns in a composition service [4]. Some work also considered the domain specific QoS attributes. For example, Canfora et al. emphasized the important of combination of Application specific QoS [5]. Ma et al. described the semantic view of QoS based selection, with emphasis on domain specific property [18]. Mobedpour and Ding proposed a QoS requirement description strategy, which is a fuzzy model of expression and matching. They suggested calculating Euclidean distance between requirement and properties from services for ranking. Besides, a process wizard, which is used to guide user to formulate a query, has been designed [19]. Comuzzi and Pernici depicted the QoS property dimensions and matchmaking process in detail, where the requirement is satisfied equally by ranges of each property, rather than different point has different satisfaction [20]. In our research, we consider both general QoS attributes and domain specific QoS attribute. The feature of our QoS model is that we consider important factors for predicting and evaluating QoS attributes for service selection based on studies in real applications, which is different from previous research. Moreover, we consider how to meet user requirements based on the QoS for service selection.

B. QoS Prediction

Previous work on QoS prediction included approaches of usage of history data, user experience and so on [9] [10]. Wu and Yang suggested a model-driven approach for QoS prediction, which analyzes the activity from BPEL model and makes use of average value for prediction [21]. Shao et al. proposed a prediction strategy, which makes similarity mining based on Pearson correlation coefficient, and making prediction based on collaborative filtering based on other consumers' QoS experiences [22]. Zeng proposed a simplified QoS prediction-enable SOA architecture, which analyzes event processing log, mines data according to existing model, and computes prediction. The prediction criteria is learned decision tree [9]. Malak proposed a multi-agent based QoS prediction architecture, which includes consumer agent, selection agent, monitor agent and QC agent. The prediction mechanism includes double quantization, forecasting [23]. Amirijoo et al. described a QoS prediction method for real time system, includes decision maker, controller, and reconfiguration [24]. In our research, we first extract some important factors for QoS attributes, and then predict the QoS attributes by analyzing the important factors that can be acquired from user profile information, history QoS data and service request information. The feature of this approach is that QoS prediction is dynamically conducted for each service request. Moreover, the QoS prediction mechanism can be improved by getting the feedback from users.

VII. Conclusion

There are three issues in QoS-aware service selection and service composition. First, sometimes QoS attributes cannot be aggregated for composite services. Second, QoS values may vary with different context for service invocation. Third, when there are multiple QoS attributes for services, it is always difficult to maximize all the QoS attributes because there always exists anti-correlated relations between them. In this paper, we propose an approach for selecting services based on context-aware QoS.

This paper has two main contributions. First, a context-aware QoS model is proposed considering the relations between QoS attributes and the context information by service invocation, user profile and QoS data. In the proposed context-aware QoS model, user-centered evaluation is considered in two aspects: utility of feasibility and utility of optimity. Second, a service selection approach is proposed based on the context-aware QoS model, where user requirement is considered. Moreover, we use a case study of service selection in language service domains to verify our proposed approach. The experiment results show that context information is effective for QoS prediction and service selection. Our future work will mainly focus on the QoS prediction algorithms for different types of service selection problems.

Acknowledgments

This research was supported by Service Science, Solutions and Foundation Integrated Research Program from JST RISTEX.

References

[1] L. Zeng, B. Benatallah, A. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "QoS-aware middleware for web services composition," *IEEE Transactions on software engineering*, vol. 30, no. 5, pp. 311–327, 2004.

[2] Y. Liu, A. H. Ngu, and L. Z. Zeng, "Qos computation and policing in dynamic web service selection," in *Proceedings of the 13th international WWW Alt*. New York, NY, USA: ACM, 2004, pp. 66–73.

[3] X. Wang, T. Vitvar, M. Kerrigan, and I. Toma, "A qos-aware selection model for semantic web services," in *Proceedings of the 4th ICSOC*. Springer, 2006, pp. 390–401.

[4] A. F. M. Huang, C.-W. Lan, and S. J. H. Yang, "An optimal qos-based web service selection scheme," *Inf. Sci.*, vol. 179, pp. 3309–3322, September 2009.

[5] G. Canfora, M. D. Penta, R. Esposito, F. Perfetto, and M. L. Villani, "Service composition (re)binding driven by application-specific qos," in *Proceedings of the 4th ICSOC*, 2006, pp. 141–152.

[6] B. Cavallo, M. Di Penta, and G. Canfora, "An empirical comparison of methods to support qos-aware service selection," in *Proceedings of the 2nd International Workshop on Principles of Engineering Service-Oriented Systems*. ACM, 2010, pp. 64–70.

[7] M. Alrifai and T. Risse, "Combining global optimization with local selection for efficient qos-aware service composition," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 881–890.

[8] M. Alrifai, D. Skoutas, and T. Risse, "Selecting skyline services for qos-based web service composition," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 11–20.

[9] L. Zeng, C. Lingenfelder, H. Lei, and H. Chang, "Event-driven quality of service prediction," *Service-Oriented Computing–ICSOC 2008*, pp. 147–161, 2008.

[10] Z. Zheng, H. Ma, M. Lyu, and I. King, "Qos-aware web service recommendation by collaborative filtering," *Services Computing, IEEE Transactions on*, no. 99, pp. 1–1, 2011.

[11] T. Ishida, "Language grid: An infrastructure for intercultural collaboration," in *IEEE/IPSJ Symposium on Applications and the Internet (SAINT-06)*, 2006, pp. 96–100.

[12] D. Lin, Y. Murakami, and M. Tanaka, "Dynamic service invocation control in service composition environments," in *Services Computing (SCC), 2010 IEEE International Conference on*. IEEE, 2010, pp. 25–32.

[13] J. White, T. O 'Connell, and F. O 'Mara, "The ARPA MT evaluation methodologies: evolution, lessons, and future approaches," in *Proceedings of the First Conference of the Association for Machine Translation in the Americas*, 1994, pp. 193–205.

[14] X. Wan, H. Li, and J. Xiao, "Cross-language document summarization based on machine translation quality prediction," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 917–926.

[15] W. Van Der Aalst, A. Ter Hofstede, B. Kiepuszewski, and A. Barros, "Workflow patterns," *Distributed and parallel databases*, vol. 14, no. 1, pp. 5–51, 2003.

[16] M. Jaeger, G. Rojec-Goldmann, and G. Muhl, "QoS aggregation for Web service composition using workflow patterns," in *Proceedings of the Enterprise Distributed Object Computing Conference, Eighth IEEE International*. IEEE Computer Society Washington, DC, USA, 2004, pp. 149–159.

[17] J. Cao, J. Huang, G. Wang, and J. Gu, "Qos and preference based web service evaluation approach," in *Eighth International Conference on Grid and Cooperative Computing, (GCC 2009).*, 2009, pp. 420 –426.

[18] Q. Ma, H. Wang, Y. Li, G. Xie, and F. Liu, "A semantic qos-aware discovery framework for web services," in *IEEE International Conference on Web Services (ICWS 2008)*, sept. 2008, pp. 129 –136.

[19] D. Mobedpour and C. Ding, "User-centered design of a qos-based web service selection system," *Service Oriented Computing and Applications*, pp. 1–11, 2011.

[20] M. Comuzzi and B. Pernici, "A framework for qos-based web service contracting," *ACM Trans. Web*, vol. 3, pp. 10:1–10:52, July 2009.

[21] J. Wu and F. Yang, "A model-driven approach for qos prediction of bpel processes," in *Service-Oriented Computing ICSOC 2006*, ser. Lecture Notes in Computer Science, vol. 4652. Springer Berlin / Heidelberg, 2007, pp. 131–140.

[22] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized qos prediction for web services via collaborative filtering," in *IEEE International Conference on Web Services (ICWS 2007)*, july 2007, pp. 439–446.

[23] J. Malak, M. Mohsenzadeh, and M. Seyyedi, "Web service qos prediction based on multi agents," in *Computer Technology and Development, 2009. ICCTD '09. International Conference on*, vol. 1, nov. 2009, pp. 265 –269.

[24] M. Amirijoo, A. Tesanovic, T. Andersson, J. Hansson, and S. Son, "Finite horizon qos prediction of reconfigurable firm real-time sytems," in *Embedded and Real-Time Computing Systems and Applications, 2006. Proceedings. 12th IEEE International Conference on*, 0-0 2006, pp. 233 –241.