2009 Fifth International Conference on Semantics, Knowledge and Grid

User-Centered QoS in Combining Web Services for Interactive Domain

Arif Bramantoro[1], Toru Ishida[2]

Department of Social Informatics, Kyoto University
Yoshida-honmachi, Kyoto, Japan
[1]arif@ai.soc.i.kyoto-u.ac.jp
[2]ishida@i.kyoto-u.ac.jp

***Abstract* — The success of the emerging service oriented computing relies fully on the Quality of Service (QoS). However, existing QoS techniques do not accommodate users' skills and preferences. We propose user-centered QoS, which is a QoS defined by the interaction between skills/preferences of service user(s) and quality of service provider(s). By implementing user-centered QoS approach, the best service is delivered to users based on the calculation not only the quality of the services but also the skill/information of users. We proposed a novel two-stage approach for combining services in user-centered QoS, i.e. intra-workflow and inter-workflow service selection. Intra-workflow service selection is used to calculate the most optimal QoS value for each composite service. Inter-workflow service selection is used to search for the most optimal combination of composite services by utilizing the QoS values obtained from intra-workflow service selection. In this paper, we provide a concrete example of user-centered QoS in the language services domain. This problem arises when there are multi users with different quality of English using multilingual chat service.**

I. Introduction

We are already in mature era of service oriented computing, with a rapid progress into the complete philosophy or paradigm rather than merely technology. The visionary promise of delivering dynamic creation of loosely coupled information system is almost into reality. Both industrial and research efforts within the vision of service oriented computing are vastly spanning various disciplines, including Quality of Service (QoS).

Having QoS in any concepts and technologies of web service is inevitable; in fact the QoS is implicitly available in all applications and just need to be exploited. However, current QoS researches are not aligned with the definition of *service* in service oriented computing. Researchers tend to define QoS as a one-way concept from service provider to service user. QoS should be based on the interaction between service users and service providers as Zhang et al. define the concept of *service* in their book [22]. Based on this *service* definition, we propose a new concept of user-centered QoS in service oriented computing to emphasize the need of accommodating the interaction between service users and providers. We define user-centered QoS as a QoS that involves users more in the QoS calculation and control based on the interaction between users and the services or the providers of the services, not just one-way definition from providers.

The current QoS researches [3,6] in service oriented computing only take the concept of QoS from network domain for granted. QoS is actually not only about underlying network, but also the capability of service provider and at the same time the user skills or preferences. The current techniques of QoS based web service selection [10], [11] only accommodate few information regarding user's skills or preferences. For example, Chaari *et al*. in [23] provides consumer's requirements, however, the requirements are only limited to the given metrics (reliability, response time, etc) that the consumers do not have other options. The same problem exists in another paper in [24] that provides a capture to user preference (even for dynamically changing preference), but it lacks a flexibility to define new metrics based on the user's needs. Failure in satisfying these requirements will deliver to the user disappointment in using web service.

To address the importance of user-centered QoS in service oriented computing, we need a concrete and complete example of QoS problem. Recently, we faced a fundamental QoS related problem in a real application. This problem arose, when we used multilingual chat service that combines translation services and morphological analyzer services in different languages [14]. We found an interesting situation in this composite service. It started when there were initially two users using the service, Japanese and Chinese users. Japanese user was good in English, but Chinese user had no English capability. The chat service thus provided Japanese-Chinese translation service.

After a while, another user came and wanted to join the conversation. This user was from Indonesia who could speak English considerably enough. Since the Indonesian-English translation was available, the chat service was composed by multi-hop translation service for Japanese-English-Indonesian and Chinese-English-Indonesian. However, the QoS of these multi-hop translation services was not good enough [15]. The translation results were terrible. All users got disappointed of this irritating communication. This irritating problem can be avoided if the user-centered QoS aware service selection is available. The service selection should consider the QoS related multiuser condition and manage this information for QoS calculation together with QoS information from provider. Based on this new QoS calculation, the best combination of services can be selected and delivered to users.

Motivated by the aforementioned problem, we propose a new framework that introduces two-stage approach of service

978-0-7695-3810-5/09 $26.00 © 2009 IEEE
DOI 10.1109/SKG.2009.106

selection for user-centered QoS, i.e. intra-workflow and inter-workflow service selection. We use intra-workflow service selection to calculate the most optimal QoS value for each composite service and inter-workflow service selection to search for the most optimal combination of composite services by utilizing QoS values obtained from intra-workflow service selection. We argue that one-stage service selection is not enough to solve the problem of user-centered QoS, especially in multiuser environment.

The aim of this paper is to optimize a concrete problem of user-centered QoS by using a robust technique and a reliable architecture, even if the environment dynamically changes. We realize that there have been some breakthroughs of QoS researches in service oriented computing. However, we argue that none of these researches can solve the fundamental problems that we found in language services and most likely in other services. Hence, our contributions are as follows: (a) we give a new concept of user-centered QoS in service oriented computing; (b) we present a novel approach of two-stage service selection, i.e. intra-workflow and inter-workflow service selection, in user-centered QoS; (c) we provide a concrete example of user-centered QoS problem to show the importance of accommodating an interaction between users' skill/preference and the service being used.

The rest of this paper is organized as follows. Section 2 presents our concept of user-centered QoS in service oriented computing. Section 3 describes the approach of intra-workflow web service selection for user-centered QoS, while inter-workflow service selection is in Section 4. A complete description of user-centered QoS problem is described in Section 5. Section 6 shows the architecture of user-centered QoS. Finally, we summarize and conclude the paper in Section 7.

II. User-Centered QoS in Service Oriented Computing

We define user-centered QoS as a different approach of QoS that emphasizes the interaction between service users and service providers. This definition is aligned with the definition of *service* for service oriented computing written in Zhang *et al.*'s book [22] as follows:

> "*Services* represent a type of relationships-based *interactions* (activities) between at least one service provider and one service consumer to achieve a certain business goal or solution objective."

We argue that it is essential to adopt the concept of *interaction* from the definition of service in service oriented computing to the concept of QoS. Although original concept of QoS is from network domain, it is necessary to have distinct concept of QoS in service oriented computing.

In user-centered QoS, the interaction between service users and service providers has several key factors that influence the overall quality. We propose user preferences or skills that can be used as key factors in the interaction. In user-centered QoS framework, any users can give a preference of the service that they want to use or let their skills included in combining web services. This framework provides high flexibility for users to choose what QoS requirements of the services that they prefer to use or appropriate to their skills. Therefore, users will get what they want. For example, user skill of bidding (a combination between trust score, number of sold and bought product) should be considered as a key factor in deciding the best services of internet auction delivered to user.

Another example is a commonly used scenario in many service oriented computing examples, i.e. travel planner services. Suppose there are multi-national passengers who want to travel together. There is a user preference that related to these passengers, which is hospitality. For the users from Asia might consider the hospitality from the flight attendance is importance whereas their other colleges who from Europe and America do not consider this issue. So, there is a different level for hospitality between these users of the same travel service that we have to deal with.

The last example that we use to show that user-centered QoS is a real problem is language service, which exists in both single-user and multiuser environment. In single-user environment, there is a Japanese user who wants to use dictionary service. Since there are two dictionary services available, i.e. English-to-English dictionary service and English-to-Japanese dictionary service, the service selection should consider the QoS related condition of the user, i.e. mother tongue and English capability that can be indicated from language certificate. In multiuser environment, mother tongue and English certificate should be included also in combining different translation services for each user. The example of multiuser language service problem is already explained in introduction section. Due to the limited space of this paper, we use a multiuser based language services as a running example throughout this paper.

In addition to the previously mentioned research problem of QoS, it becomes a common sense amongst researchers in service oriented computing that QoS metrics is related to network domain and, therefore, they adopt the entire network metrics into service oriented computing, such as response time, reliability, availability, and so on. There are only few researches, to our knowledge, that propose a new metric related to particular domain and accommodate user requirements [13], [19]. However, these researches lack a real example in service oriented application and an integrated solution to calculate the metrics. This will cause inability to show the importance of accommodating users in QoS control. A special attention is given to the previous work [25] that provides a flexible framework to change QoS metrics based on user preference. However, this paper still uses network-domain QoS metrics or other QoS metrics, such as price, that is not related to network but is actually used by application.

To solve the problem of user-centered QoS, we need a robust technique and a flexible specification for user-centered QoS. We choose to use and extend constraint optimization technique [20], a well known AI technique to solve many sophisticated problems, such as scheduling, temporal reasoning, resource allocation, etc. Accordingly, the problem of web service selection can be modeled and solved by using constraint optimization technique. Previously, Ben Hassine *et al.* in [7] has formulized Web service composition problem based on a constraint optimization problem (COP), while Channa *et al.* in [8] has proposed the use of constraint satisfaction problem (CSP) in dynamic web service composition. However, these two papers did not include QoS management constraints and even can solve the user-centered QoS problem that we found.

Original constraint optimization problem is characterized with a triplet entities (*X, D, C*) plus objective function. *X* is a finite set of variables associated with finite domains *D* as a list of possible values for each variable, whereas *C* is a set of constraints. In our approach, it is possible to define conditional constraints [2] to accommodate the resource allocation, especially when there is a resource dependent to other resources. Lastly, the objective function is optimized to find a complete assignment of values to all variables and at the same time satisfying the constraints.

In the web service selection point of view, we extend the triplet of constraint optimization problem into quadruplet. A new variable, *P*, is created to accommodate user profile that defines user skills or preferences. As an example, *P* in the language service can be mother tongue and foreign language certification score. Hence, the extended constraint optimization formulization is as follows:

- $X=\{X_1,\ldots,X_n\}$ is a set of abstract web services, with $X_i.IN$ is a set of required input types, $X_i.OUT$ is a set of required output types, $X_i.QOS$ is a set of required QoS types. These requirements are defined as abstract service specifications.
- $D=\{D_1,\ldots,D_n\}$ where D_i a set of concrete web services X_i that can perform the task of the corresponding abstract web services.

  $D_i=\{s_{i1},\ldots,s_{ik}\}$ where s_{ij} is a concrete web service of the corresponding X_i with $s_{ij}.IN$ is a set of provided input types, $s_{ij}.OUT$ is a set of provided output types, $s_{ij}.QOS$ is a set of provided QoS types. In semantic matching of web service selection [4], every element of the input set in concrete service specification should be also an element of the input set in abstract service specification and every element of the output set in abstract service specification should be also an element of the output set in concrete service specification. We argue that in QoS based matching every element of the QoS set in abstract service specification should be also an element of the output set in concrete service specification. Therefore, we define semantically matched service specification as follows.
  - $D_i=\{s_{ij} \mid s_{ij}.IN \subseteq X_i.IN \wedge X_i.OUT \subseteq s_{ij}.OUT \wedge X_i.QOS \subseteq s_{ij}.QOS\}$
- $P=\{P_1,\ldots,P_m\}$ is a set of user profile obtained from each user. P_i consists of profile values of user *i*.
- $C=\{C_1,\ldots,C_p\}$ is a set of constraints which contains *CS* as a set of soft constraints with a penalty of $\rho C_i \in[0, 1]$, and *CH* as a set of hard constraints.
- *f(R)* is the objective function to be maximized. The goal is to find the best assignment *R* for the variables in *X* while satisfying all the hard constraints. *R* is the resulted solution of a problem assigned by the instantiation of all variables of the problems. In the web service selection, we define the objective function *f(R)* by using penalty over soft constraints $\psi(R)$ and QoS function *QoS(R)* as shown in Eq. 1.

$$f(R)=QoS(R)-\psi(R) \quad \textbf{(1)}$$

To solve web service selection problem, we have to find the best assignment of the variable *R** such that, all the hard constraints are satisfied while maximizing the following function in Eq. 2.

$$R^*=arg\ \max_{R\in Solution} f(R) \quad \textbf{(2)}$$

The penalty over soft constraints can be calculated by summing the penalties associated to all soft constraints as described in Eq. 3.

$$\psi(R)=\sum_{C_k\in CS}\rho C_k \quad \textbf{(3)}$$

The QoS functions consists of commonly used QoS metrics, such as price, reputation, reliability, availability; and other newly defined QoS metrics from users. The detail QoS function is described in the Eq. 4 where *Q(R)* is a QoS function obtained from existing known aggregation and/or newly defined function for customized QoS metrics and *m* is the number of QoS metrics.

$$QoS(R)=Q_1(R)+Q_2(R)+\ldots+Q_m(R) \quad \textbf{(4)}$$

To calculate each QoS function, we refer to the two papers [5], [13] that provide the aggregation functions of most QoS metrics in network domain, such as time, price, availability, reliability, reputation and success rate. Zeng *et al.* in [5] gives a foundation for QoS aggregation function. Canfora *et al.* [13], on the other hand, provides specific aggregation functions for each workflow constructs and additionally domain-dependent attribute. Our approach handles user-specified attribute differently to what proposed in [13]. We argue that QoS aggregation function for user-specified attribute should be defined freely by users (or third parties, such as service brokers) based on particular domain.

III. Intra-Workflow Service Selection

In this section, we give a detail explanation of intra-workflow service selection whereas inter-workflow service selection will be explained in the next section. As introduced partly in the first section, we provide a concrete problem of user-centered QoS in the multiuser environment. Our approach in solving user-centered QoS problem in multiuser environment is based on the two-stage service selection, i.e.: intra-workflow and inter-workflow service selection. Intra-workflow service selection is used to calculate the most optimal QoS value for each composite service. Inter-workflow service selection is used to search for the most optimal combination of composite services by utilizing QoS values obtained from intra-workflow service selection. To see the relation between these two service selections, we provide an interaction model as described in Fig. 1.

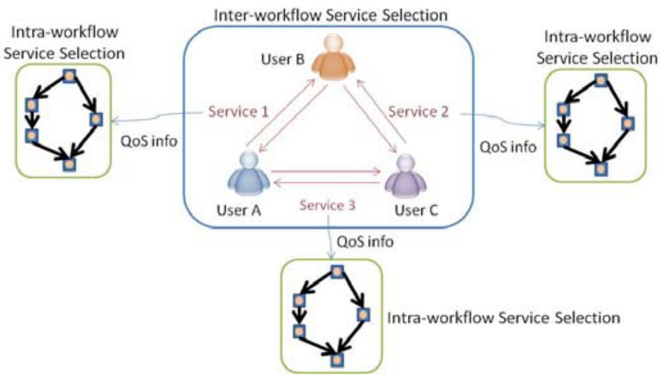

Fig. 1. Interaction model between inter-workflow and intra-workflow service selection

It is clearly seen from Fig. 1 that each service in inter-workflow service selection has QoS value resulted from intra-workflow service selection. In a real world, the service used by each user might be in the form of composite service. In case it is composite service, we need to calculate QoS based on service workflow. The calculation of QoS in each workflow is performed in intra-workflow service selection. Since there are some possible services for each users, QoS of each possible service should be calculated separately in intra-workflow service selection. In intra-workflow service selection, QoS calculation for each workflow is based on the most optimal solution of concrete services. In other words, intra-workflow service selection calculates the total QoS value of all concrete services composed in one workflow.

As an example of intra-workflow service selection, let us take a part of user-centered QoS problem in the language services. In the language service, we can compose a translation service with the community dictionary service to increase the quality of translation [1]. One of the workflow for possible concrete composite service between Japanese user and Indonesian user is ja-id translation service as described in Fig. 2. The detail calculation of QoS based on objective function will be explained in Section 5.

The formulization for this workflow is as follows:

- $X=\{X_1, X_2, X_3, X_4, X_5\}$, where:
  - X_1: Morphological analyzer service;
  - X_2: ja-en translation service;
  - X_3: en-id translation service;
  - X_4: Community dictionary service;
  - X_5: Term replacement service;
- $D=\{D_1, D_2, D_3, D_4, D_5\}$, where
  - D_1: {mecab at NTT, ICTCLAS, KLT at Kookmin University, treetagger at IMS Stuttgart};
  - D_2: {JServer at Kyoto-U, JServer at NICT, WEB-Transer at Kyoto-U, WEB-Transer at NICT};
  - D_3 : {ToggleText at Kyoto-U, ToggleText at NICT};
  - D_4: {Life Science Dictionary, Natural Disasters Dictionary, Kyoto Tourism Dictionary at NICT, Academic Terms Dictionary at NII};
  - D_5: {TermRepl service};

  (For *the* sake of simplicity, we omit the input and output parameters of D_i)
- $C=CS \cup CH$, in this intra-workflow service selection, however, we only employ hard constraints so that the objective function focuses on calculating the aggregated QoS values, where:
  - CH including (due to page limitation, only example constraints are shown)
    - C_1: For multi hop translation, $X_2.OUT=X_3.IN$;
    - C_2: For composite service which involves X_2 and X_4 (translation service and multilingual dictionary), serverLocation(X_2)=serverLocation(X_4);
    - C_3: For morphological analysis used together with community dictionary services, partialAnalyzedResult($X_1.OUT$) $\in X_4.IN$.

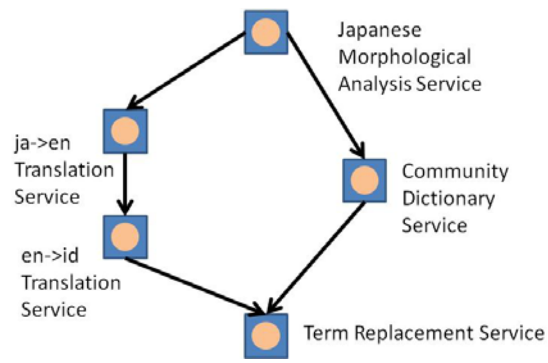

Fig. 2. A workflow of Japanese-Indonesian translation service

IV. Inter-Workflow Service Selection

In inter-workflow service selection, there is a combination of services between users in multiuser environment. One user can have different service from the service used by other users. This combination is not necessarily related to the control of workflow, such as sequence, split, choice and loop. The relation of services used by each user is more likely in the form of constraints. The main task of inter-workflow service selection is to find the best combination of services that meet the QoS constraints based on QoS related condition of users and the quality of the service itself.

To solve our formulization of user-centered constraint optimization problem for QoS, we use a simple search algorithm for constraint optimization problem. Our algorithm is based on the basic search algorithm for constraint optimization, branch-and-bound algorithm [20]. The aim of using this algorithm is to find the best solution by extending backtracking search to traverse the search space seeking all solutions. It maintains the value of objective function so far, which is so called a lower bound. In addition, for each partial solution, the algorithm also computes an upper bound using a bounding evaluation function, which overestimates the best-solution in objective function that can extend the partial solution. Therefore, when the upper bound of the partial solution is less than the lower bound, the partial solution can be aborted, and the algorithm backtracks, pruning the subtree below the partial solution. The algorithm returns to the previous partial solution and attempts to find a new assignment to X.

We have to slightly modify this algorithm to incorporate user-centered QoS in constraint optimization. The modification is related to the checking whether the QoS information of current domain's workflow is already calculated or not. If the QoS information is not yet calculated in intra-workflow service selection, then the algorithm will call intra-workflow function to calculate the QoS of the current domain. The intra-workflow function is similar to the search algorithm for inter-workflow service selection. The difference is that the intra-workflow function delivers the optimized QoS information of particular domain, not the optimized solution.

As any other search algorithms in constraint optimization technique [9], our algorithm produces the complexity of NP-Hard. Here, we argue that the function of intra-workflow is rarely executed. This is due to that the workflow does not easily change over the time and a new service is not added frequently. Furthermore, in our architecture this function can be executed in offline processing. Therefore, the number of constraints and services is fixed and we can maintain the complexity of this algorithm in polynomial time not NP-Hard anymore, unless for a worse case when the workflow changes or there is a new service added in the set of concrete web services frequently.

As an example of inter-workflow service selection, let us take a part of user-centered QoS problem in the language services. The problem of multilingual chat service can be formulized as follows (the detail service selection with objective function will be explained in Section 5):

- $X=\{X_1, X_2, X_3, X_4, X_5, X_6\}$, where
  - X_1: service from Japanese user to Chinese user;
  - X_2: service from Chinese user to Japanese user;
  - X_3: service from Japanese user to Indonesian user;
  - X_4: service from Indonesian user to Japanese user;
  - X_5: service from Chinese user to Indonesian user;
  - X_6: service from Indonesian user to Chinese user;
- $D=\{D_1, D_2, D_3, D_4, D_5, D_6\}$, where
  - D_1: {ja-en, ja-zh, en-zh, no translation service};
  - D_2: {zh-en, zh-ja, en-ja, no translation service};
  - D_3: {ja-en, ja-id, en-id, no translation service};
  - D_4: {id-en, id-ja, en-ja, no translation service};
  - D_5: {zh-en, zh-id, en-id, no translation service};
  - D_6: {id-en, id-zh, en-zh, no translation service};
- $P=\{P_1, P_2, P_3\}$, where
  - P_1 is a user profile of Japanese user.
    P_1.mother_tongue=Japanese, P_1.english_writing_skill=0.8, P_1.english_reading_skill=0.9;
  - P_2 is a user profile of Chinese user.
    P_2.mother_tongue=Chinese, P_2.english_writing_skill=0.1, P_2.english_reading_skill=0.2;
  - P_3 is a user profile of Indonesian user.
    P_3.mother_tongue=Indonesian,
    P_3.english_writing_skill=0.6,
    P_3.english_reading_skill=0.6;
- $C=CH \cup CS$ (we will present the soft constraints CS in Section 5), where
  - Hard constraints CH, where each user should type in one language (although it is possible to type more than one languages in chat services, we assume that the user preference of one language is a hard constraint), including
    - C_1: X_1=ja-en => (X_3=ja-en $\vee$ X_3=ja-id);
    - C_2: X_1=ja-zh => (X_3=ja-en $\vee$ X_3=ja-id);
    - C_3: X_1=en-zh => X_3=en-id;
    - C_4: X_2=zh-en => (X_5=zh-en $\vee$ X_5=zh-id);
    - C_5: X_2=zh-ja => (X_5=zh-en $\vee$ X_5=zh-id);
    - C_6: X_2=en-ja => X_5=en-id;
    - C_7: X_4=id-en => (X_6=id-en $\vee$ X_6=id-zh);
    - C_8: X_4=id-ja => (X_6=id-en $\vee$ X_6=id-zh);
    - C_9: X_4=en-ja => X_6=en-id;
      (For simplicity, we omit the other way around of the constraints C_{10} to C_{18})
    - C_{19}: X_1=no_translation => (X_3=no_translation $\vee$ X_3=en-id);
    - C_{20}: X_2=no_translation => (X_5=no_translation $\vee$ X_5=en-id);
    - C_{21}: X_4=no_translation => (X_6=no_translation $\vee$ X_6=en-zh).
      (For simplicity, we omit the other way around of the constraints C_{22} to C_{24})

The complete set of the hard constraints from C_1 until C_{24} is described in Fig. 3.

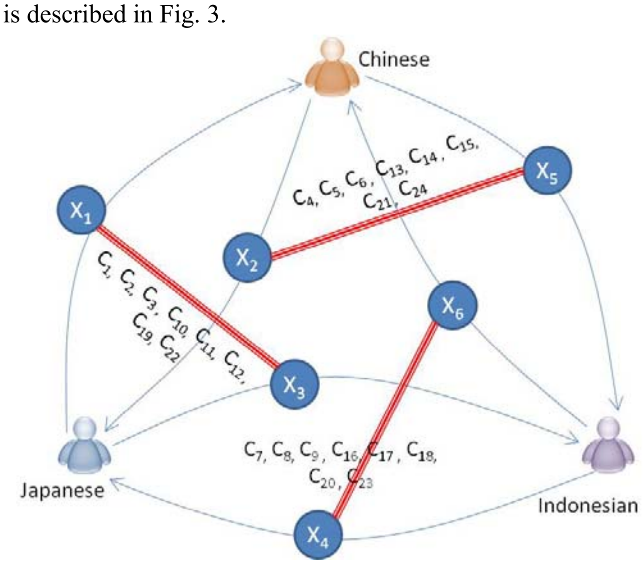

Fig. 3. Simplified constraint graph for hard constraint examples in intra-workflow service selection

V. USER-CENTERED QOS IN MULTIUSER ENVIRONMENT

In this section, we present a real scenario that shows the problem of user-centered QoS in detail. This scenario involves a complete set of web services and frequently used by real users, i.e. the Language Grid [16]. The Language Grid is a service oriented collective intelligent platform to collect and share language services. Delivering QoS on the Language Grid is challenging because there are many applications with different characteristics and requirements compete for all language resources [17].

QoS metric applicable to language service is accuracy which consists of the combination between fluency and adequacy [15]. Fluency refers to well-formed grammar, contains correct spellings, adheres to common use of terms, titles and names, is intuitively acceptable and can be sensibly interpreted by a native speaker. Adequacy refers to the degree to which information present in the original is also communicated in the translation.

In the case of multilingual chat service, user-centered QoS approach is needed to calculate the information of user's ability in language and the accuracy of translation. When initially there were two users, a Japanese user with good English and a Chinese user with no English, the composition should automatically select the translation service from Chinese to Japanese and vice versa. After an Indonesian user who can speak English a little joined the conversation, the composition should recalculate QoS information from each translation service and compare it with each user's language capability. In this case, the chat service should include Chinese-English translation for communicating Chinese and Indonesian users; but no translation service (English only communication service) for Indonesian and Japanese users. This is due to the poor quality of Japanese-Indonesian and Chinese-Indonesian translation services, which use multi-hop translation services with English as a pivot language [14]. We provide Fig. 4 to clearly understand this problem.

In intra-workflow service selection, the objective function is used to retrieve the optimized QoS value of each workflow. Hence, the aim of this objective function is not to find the best solution but rather than to retrieve the QoS value of composite services that can be used by inter-workflow service selection. We use the same objective function in Eq. 1 modified to compromise with the characteristic of language service's quality of service. The cascaded translation service represented with sequential workflow reduces the overall quality. The multi-hop translation service represented by two translation services in sequence workflow gives the most significant influence to the overall quality and therefore should be given the biggest weight amongst others, i.e. 0.6 for ja-en and en-id translation services. However, we use multiplication for these services since the quality becomes much decreasing if we combine two translation services as in the following Eq. 5.

$$f(R) = 0.2 \times s_{1j}.accuracy + 0.6 \times s_{2j}.accuracy \times s_{3j}.accuracy + 0.1 \times s_{4j}.accuracy + 0.1 \times s_{5j}.accuracy \quad \textbf{(5)}$$

We assume that the accuracy value from each language service in this implementation is available from language evaluation system utilizing human evaluation system or automatic one such as BLEU [12]. As a result of intra-workflow service selection, the most optimal QoS accuracy value for ja-id translation service is delivered by the combination of {mecab at NTT, WEB-Transer at NICT, ToggleText at NICT, Kyoto Tourism Dictionary at NICT, TermRepl service}.

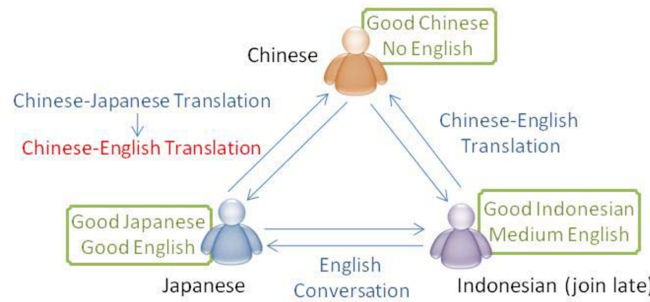

Fig. 4. Multilingual chat service problem

Inter-workflow service selection can use the resulted QoS values obtained from intra-workflow service selection. We introduce a new function to estimate the quality of message (QoM) that calculates each possible abstract translation service between two users (represented by users' profile). In this case, we consider mother tongue of user, English writing skill and reading skill as user profile. We define (QoM) function sent by one user represented by user profile P_i and received by another user represented by user profile P_j that uses translation service X_n in Eq. 6.

$$QoM(P_i, X_k, P_j) = Accuracy(P_i.writing\_skill(X_k.input\_language)) \times X_k.accuracy \times Accuracy(P_j.reading\_skill(X_k.output\_language)) \quad \textbf{(6)}$$

In inter-workflow service selection, the objective function is used to find the best solution. This function consists of penalty over soft constraints *ψ(R)* and QoS function *QoS(R)* as described in Eq. 1. Since QoS function in this case is calculated based on user-defined QoS metrics, i.e. translation accuracy values of each service, the QoS function is modified from Eq. 4 as the summation of QoM function in Eq. 6 which is described in the following Eq. 7.

$$QoS(R) = \sum_{\substack{X_k \in R \\ ServiceInBetween(P_i,P_j)=X_k}} QoM(P_i, X_k, P_j) \quad \textbf{(7)}$$

The most optimal result for this problem is {en-zh translation service, zh-en translation service, no translation service, no translation service, zh-en translation service, en-zh translation service}.

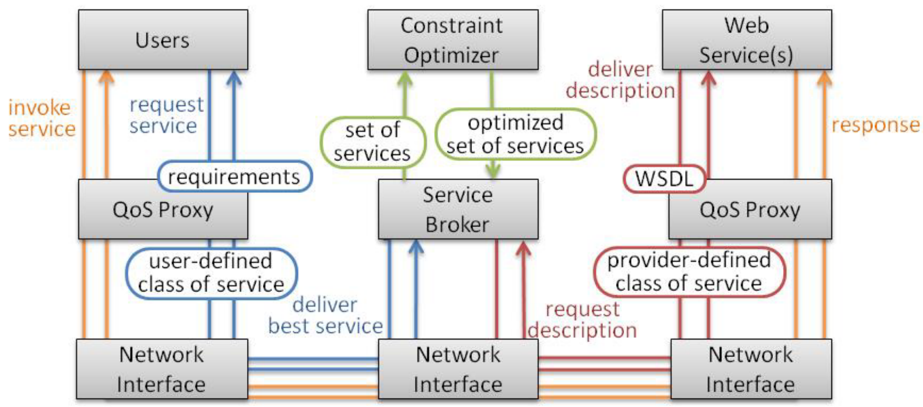

Fig. 5. User-centered QoS architecture

VI. User-Centered QoS Architecture

In this section, we implement user-centered QoS in a real system by designing the user-centered architecture for web service selection. To support user-centered QoS framework, we extend the original version of QoS proxy as been previously introduced in [21]. In our architecture, the job of QoS proxy is to translate user requirements of web services and QoS into user-defined class of service. Another job of QoS proxy is to translate WSDL into provider-defined class of service. These two classes of service can be evaluated in constraint optimizer sent by service broker. Fig. 5 illustrates a complete architecture between web service user(s), service broker, and web service provider(s).

In this architecture, each provider can offer different classes of service for different QoS and each class of service can be utilized by more than one user. By having these two kinds of class of service, there is flexibility for users to (re)define their own QoS metric with their own QoS value. This architecture also has an advantage of allowing users to create a new QoS metric based on their needs if the existing class of service is not suitable for them.

The scenario in our architecture is as follows. Initially user requests a service by defining her requirements through QoS proxy in which translating the requirements into class of service and sending it to service broker. Service broker then requests service descriptions based on broker's own database or third party, such as UDDI, to service provider. Getting a description request by service broker, a service provider sends his class of service that is previously translated by QoS proxy from WSDL. The next step is running the constraint optimization algorithm based on the constraints inside user-defined and provider-defined class of service. The constraints together with a set of potential services sent by service broker fed into constraint optimizer to produce a number of feasible services which then can be ranked to find the optimal solution. The final step is the service invocation from user after receiving the best service from service broker.

VII. Conclusion

In this work, we proposed a new concept in service oriented computing, i.e. user-centered QoS in combining web services. User-centered QoS is a QoS defined by the interaction between service user(s) and the service itself. The previous concept of QoS in service oriented computing is a QoS that is delivered by service provider to service user. This is contradicted to the concept of *service* in service oriented computing that should be based on the provider and user interaction. This is also against the fact that the best practices of most service oriented applications, especially in multiuser environment, need the QoS interaction between user skills / preferences and provider. Three examples are given in this paper, QoS of travel planner service used by multi-national passengers use with different judgment on hospitality factor, QoS of multimedia services decided on user's behaviour, and QoS of language service based on language capability of each user. In this paper, we gave a complete explanation of user-centered QoS problem to the last example, i.e. language service.

In this paper, we presented a fundamental QoS related problem. This problem arose when we used multilingual chat service that combines several language services, such as translation services and morphological analyzer services in different languages. It started when there were two users using the service. They were Japanese and Chinese users. The Japanese user was good in English, but the Chinese user had no English capability. The chat service thus should automatically provide Japanese-Chinese translation service. After a while, another user from Indonesia who could speak English considerably enough joined the conversation. Since the Indonesian-English translation was available, the chat service was composed by multi-hop translation service for Japanese-English-Indonesian and Chinese-English-Indonesian. However, the QoS of these multi-hop translation services was not good enough. All users got disappointed of this irritating communication. In user-centered QoS aware, the chat service should automatically provide no-translation chat service between Japanese and Indonesian since they have a quality of

English much better than the QoS of multi-hop translation services.

In our experiment, the problem of user-centered QoS cannot be solved in one-stage of service selection. Therefore, we proposed a novel two-stage approach for combining services, i.e. intra-workflow and inter-workflow service selection. Intra-workflow service selection is used to calculate the most optimal QoS value for each possible workflow. Inter-workflow service selection is used to search for the most optimal solution by utilizing the QoS values obtained from intra-workflow service selection. This two service selections utilize the modified technique of constraint optimization and a reliable architecture based on user-defined and provider-defined class of service.

Acknowledgment

This research was partially supported by a Grant-in-Aid for Scientific Research (A) (21240014, 2009-2011) from Japan Society for the Promotion of Science (JSPS), and also from Global COE Program on Informatics Education and Research Center for Knowledge-Circulating Society.

References

[1] Y. Murakami, T. Ishida, T. Nakaguchi, "Infrastructure for Language Service Composition," in *Proc. SKG'06*, 2006.

[2] S. Mittal, B. Falkenhainer, Dynamic constraint satisfaction problems. in *Proc. AAAI'90*, 1990, pp. 25–32. MIT Press.

[3] S. Ran, "A Model for Web Services Discovery with QoS," *ACM SIGecom Exchange*, vol. 4, issue 1, 2003.

[4] M. Paolucci, T. Kawamura, T.R. Payne, K. Sycara, "Semantic Matching of Web Services Capabilities," in: *Proc. ISWC'02*, 2002.

[5] L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, Q.Z. Sheng, "Web Engineering: Quality Driven Web Service Composition," in *Proc. WWW'03*, 2003, ACM Press.

[6] A. Sahai, J. Ouyang, V. Machiraju, K. Wurster, BizQoS, "Specifying and Guaranteeing Quality of Service for Web Services through Real Time Measurement and Adaptive Control," Hewlett-Packard Labs Technical Report HPL-2001-134, 2001.

[7] A.B. Hassine, S. Matsubara, T. Ishida, "Constraint-based Approach to Horizontal Web Service Composition," in *Proc. ISWC'06*, 2006, LNCS, vol. 4273, pp. 130-143. Springer-Verlag.

[8] N. Li, S. Channa, A.W. Shaikh, X. Fu, "Constraint Satisfaction in Dynamic Web Service Composition," in *Proc. DEXA'05*, 2005, pp. 658-664.

[9] L. Li, J. Wei, T. Huang, "High Performance Approach for Multi-QoS Constrained Web Services Selection," in *Proc. ICSOC'07*, 2007, pp. 283-294.

[10] L. Zeng, H. Lei, H. Chang, "Monitoring the QoS for Web Services," in *Proc ICSOC'07*, 2007, pp. 132-144.

[11] C. Zhang, R.N. Chang, C. Perng, E. So, C. Tang, T. Tao, "QoS-Aware Optimization of Composite-Service Fulfillment Policy," in *Proc ICSOC'07*, 2007, pp. 11-19.

[12] K. Papineni, S. Roukos, T. Ward, W. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proc. ACL'02*, 2002, pp. 311-318.

[13] G. Canfora, M.D. Penta, R. Esposito, M.L. Villani, "A framework for QoS-aware binding and re-binding of composite web services," Journal of Systems and Software, vol 81, issue 10, pp. 1754-1769, 2008.

[14] M. Tanaka, T. Ishida, Y. Murakami, S. Morimoto, "Service Supervision: Coordinating Web Services in Open Environment," in *Proc. ICWS'09*, 2009.

[15] R. Tanaka, Y. Murakami, T. Ishida, "Context-Based Approach for Pivot Translation Services," in *Proc. IJCAI'09*, 2009.

[16] T. Ishida, "Language Grid: An Infrastructure for Intercultural Collaboration," in *Proc. SAINT'06,* 2006, pp. 96-100.

[17] A. Bramantoro, M. Tanaka, Y. Murakami, U. Schäfer, T. Ishida, "A Hybrid Integrated Architecture for Language Service Composition," in *Proc. ICWS'08*, 2008, pp. 345-352.

[18] I. Matsumura, T. Ishida, Y. Murakami, Y. Fujishiro, "Situated Web Service: Context-Aware Approach to High Speed Web Service Communication," in *Proc. ICWS'06*, 2006, pp. 673-680.

[19] V. Deora, J. Shao, W.A. Gray, N.J. Fiddian, "A Quality of Service Management Framework Based on User Expectations," in *Proc. ICSOC'03*, 2003, pp. 104-114. Springer, Heidelberg.

[20] R. Dechter, *Constraint Processing*, Morgan Kaufmann, San Francisco, 2003.

[21] M. Tian, A. Gramm, T. Naumowicz, H. Ritter, J. Jchiller, "Efficient Selection and Monitoring of QoS-aware Web services with the WS-QoS Framework," in *Proc. WI'04*, 2004, pp. 152-158. IEEE Press.

[22] L. J. Zhang, J. Zhang, H. Cai, *Services Computing*. Springer-Verlag, 2007.

[23] S. Chaari, Y. Badr, and F. Biennier, "Enhancing web service, selection by qos-based ontology and ws-policy," in *Proc. SAC '08, 2008*, pp. 2426–2431, New York, NY, USA, ACM.

[24] H. Q. Yu and S. Reiff-Marganiec, "A method for automated web service selection," in *Proc. SERVICES'08, 2008*, pp. 513–520, USA.

[25] S. Lamparter, A. Ankolekar, R. Studer, and S. Grimm, "Preference-based selection of highly configurable web services," in *Proc. WWW'07*, 2007, pp. 1013–1022, USA, ACM.