Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Interactive Web service choice-making based on extended QoS model*

MOU Yu-jie†, CAO Jian, ZHANG Shen-sheng, ZHANG Jian-hong

(*CIT Lab, Department of Computer Science & Technology, Shanghai Jiao Tong University, Shanghai 200030, China*)

†E-mail: mou-yj@cs.sjtu.edu.cn

Received Feb. 20, 2005; revision accepted Nov. 11, 2005

Abstract: Quality of Service (QoS) is a key factor in Web service advertising, choosing and runtime monitoring. Web service QoS is multi-faceted, fuzzy and dynamic. Current researches focus on implementation level performance assurance, ignoring domain specific or application level metrics which are also very important to service users. Industry Web service standards lack QoS expression. The support for QoS based service choice-making is very limited. We proposed an extended Web service QoS model based on configurable fuzzy synthetic evaluation system. Web service QoS is evaluated dynamically according to the service context. A QoS requirement description model is also given for service QoS requirement definition. An interactive Web service choice-making process is described, which takes QoS as a key factor when choosing from functionally equivalent services.

Key words: Quality of Service (QoS), Web service, Choice-making, Fuzzy synthetic evaluation
doi:10.1631/jzus.2006.A0483 **Document code:** A **CLC number:** TP311

INTRODUCTION

Web service technologies are immerging as a powerful vehicle for organizations that participate in Web based dynamic collaborations. An organization can publish its business processes as Web services to the Internet for potential customers to discover and use them. Compared with traditional fixed inter-organization collaboration relationship, the Web-based dynamic collaboration brings service providers more customers and more profits, and enables consumers to search in a wider range for services that satisfy their goals best. Attracted by the win-win effect, more and more organizations are turning to service providers (Cardoso *et al*., 2004)

The available Web services are added on a daily base, which means a larger selection space for service consumers. Even for a single functional requirement, there may be so many similar services. The coexistence of functionally similar services results from two causative factors. Firstly, there is trade competition among different service providers. Secondly, a single service provider may publish services as families (a group of services that provide similar functionality through various ways and thus have various none-functional features) to meet various customers' individual requirements. Quality of Service (QoS) becomes an important factor for Web service choice-making.

QoS based Web service choice-making needs support from service description model, service requirement description model and service discovery process. Current research on Web service is limited in QoS expression, evaluation and application. Main problems are:

(1) Current QoS assurance solutions focus on common software implementation level performance metrics, such as capacity or accessibility (Mani and Nagarajan, 2002; W3C, 2003). These metrics are important from a technical point of view. However, when selecting "the best one" from functionally equivalent services, especially services of long executing time, complicated process and high cost, a

* Project supported by the National Natural Science Foundation of China (No. 60503041), the Hi-Tech Research and Development Program (863) of China (No. 2004AA104340), the Chinese Semantic Grid Project, and the Science and Technology Commission of Shanghai Municipality (No. 03dz15027), China

customer tends to pay more attention on domain specific metrics.

(2) In existing Web service description model, QoS information is expressed in collection of name-value pairs. This data structure is very limited in expressivity, and takes QoS as static data. However the Web service QoS is in fact multifaceted and dynamic, and so, needs a more expressive semantic model.

(3) In current service advertising and choice-making process the QoS information is not concerned as a key factor. Since there is no model for users to define their specific QoS requirements, QoS based service choice-making process is not possible.

This paper takes the point of view that Web service QoS is a complex synthetics composed of many metrics of different type. QoS evaluation should be performed on user concerned metrics. Each involved metric must be computed and evaluated dynamically according to service contexts. We define an extended Web service QoS model for Web service QoS expression and evaluation. The overall Web service QoS is expressed as a configurable fuzzy synthetic index system. The configuration is based on user's preference, and the evaluation is performed dynamically according to service contexts. We also present a flexible QoS requirement model to express user's requirements on target service QoS. An improved UDDI (Universal Description, Discovery and Integration) process is suggested. The new process model organizes the service provider, UDDI server and the service consumer to participate more actively in a contract-net based interactive service choice-making process.

RELATED WORK

Web service QoS model

Quality of Service has been explored in domains such as networking, real-time applications, middleware and multimedia service for service performance assurance or configuration (Cardoso *et al*., 2004). Current discussions on Web service QoS mainly concerns software implementation level metrics. For example researchers in IBM pointed out that Web service QoS should have seven metrics (Mani and Nagarajan, 2002): availability, accessibility, integrity, performance, reliability, regularity and security. A working paper of W3C (2003) lists more metrics to extend the IBM's model yet is still similar. Besides implementation level metrics that are shared by all Web services are very important for reliable and effective Web service invocation and composition, we believe future competition will also focus in domain specific application level metrics, which show the performance of the service provider's underlying business process that may differ remarkably. We regard Web service QoS as a synthesis covering both implementation level metrics shared by all Web services and the domain specific application level metrics.

Web service description standards and semantic models are little concerned with QoS expression. The Web Service Description Language (WSDL) describes Web services as a set of endpoints operating on messages. It mainly talks about communication and invocation, and does not define QoS features (W3C, 2004). The DAML-S (the DARPA Agent Markup Language for Web Service) and succeeding OWL-S (Web Ontology Language for Services) describe Web services semantically in three parts: the service profile, the process model and the service grounding. The service profile ontology includes a QoS description, which is a collection of QoS metric name-value pairs (DAML-S, 2005; OWL-S, 2004). However as we will show bellow the service QoS is a hierarchical synthesis composed of many inter-related metrics of different types. The OWL-S QoS description is limited in expressiveness and needs extension (Cardoso *et al*., 2004).

In most Web service description models, QoS is expressed as static data (DAML-S, 2005; OWL-S, 2004). Researchers have mentioned that QoS of services, including Web services, should be evaluated dynamically. Experts from workflow domain are applying process QoS expression and evaluation method to Web service QoS. Cardoso *et al*.(2004) introduced a service workflow QoS model. QoS of each activity is estimated based on historical data, and the whole service workflow's QoS is calculated from activity QoS through a regressive algorithm. Similar description and estimation efforts are also mentioned in the CrossFlow project (Busse, 2005; Chandrasekaran *et al*., 2002). However, QoS models in these projects cover only common performance metrics such as time, cost or reliability. What is more, QoS estimation is performed on historical data and the underlying service process structure. Here we point out that service context information is another important cause of QoS data floating, and should be taken into consideration in QoS estimation.

Web service choice-making

One of the main challenges in Web service based application is the choice of a Web service that suits the user requirement best. Service matchmaking has been explored through different ways. The IOPE based approach checks matchmaking on service inputs, outputs, pre-conditions and effects between user requirement and service description (Paolucci *et al*., 2003). In process-based matchmaking, message exchanges that occur during the service process are distilled into a service invocation workflow. A service is considered to be a match if its message workflow is compatible with that of a service requirement description (Wombacher *et al*., 2004a; 2004b). There is also semantic description based service matchmaking, where service advertisements and requirements are both defined as concepts, and matchmaking is carried out through consumption relationship checking (Gonzalez-Castillo *et al*., 2001; Trastou *et al*., 2001). In semantic description based approach the target service definition can contain QoS requirements. However the Web service description semantic model does not contain a QoS facet, and the QoS requirements are expressed dispersedly as property constraints on functionality terms. Such a model is not expressive and flexible enough for QoS based Web service selection.

The above situation is understandable because current researches aim at "find a match". However as Web service number is growing dramatically, soon we will have to consider "select the best one" from a set of functionally or semantically similar services.

Web service discovery and invocation follow a standard process: A service provider describes and publishes a service to a UDDI server, then waits for invocation. Per a service request the UDDI server checks for appropriate services from its records and returns service information to the consumer. Then the consumer invokes the service. The provider does not participate in the service selection phase, and the UDDI server does not care whether the collaboration process is successful or not (UDDI, 2004). However as we will explain later, Web service QoS is dynamic, and evaluation of some important QoS metrics may rely on provider side contexts, so provider's participation is needed in QoS based service selection process. We also suggest that UDDI server trace the service process and collect consumer feedback evaluation in order to provide better UDDI service in the future.

The above analysis shows that for QoS based service choice-making, the following efforts are needed: (1) The existing Web service description model must be enriched with QoS semantics; (2) A QoS requirement description model must be defined for customer to define targeting service QoS requirement; (3) Current UDDI process should be improved to encourage the consumer, the UDDI server and the provider to participate more actively in the QoS based Web service choice process. The following three sections show how to achieve these three targets.

AN EXTENDED WEB SERVICE QOS MODEL

Web service QoS characters: a discussion

Web service QoS has some immanent characters that are important for QoS models and evaluating, which existing Web service QoS models fail to cover. We analyze these characters and discuss how to embody them in a QoS model. We set up our Web service QoS description model on the basis of what are discussed here.

1. Multifaceted

Web service QoS is composed of multiple metrics. The metrics are inerter-related and form a hiberarchy. The simple and plain structure used in current Web description model cannot fully express the inter-relationship between QoS metrics.

We classify the metrics into three catalogs (Cardoso *et al*., 2004; Busse, 2005): Operational performance, which describes service implementation level metrics such as the 7 metrics of IBM QoS model; General metrics, which describe other commonly shared metrics such as time, cost, invocation time or so; domain specific metrics, which cover QoS metrics defined only in a domain ontology of that Web service, for example the hotel_level. Each metric can be composed of several sub metrics and can be extended with new metrics. The overall QoS model turns out to be a tree-like hierarchical system. When choosing from a set of functionally similar services, a consumer usually cares more for the overall QoS figure instead of some single metric.

2. Fuzzy and diversity

Some metrics can be quantitatively evaluated, while others can only be evaluated qualitatively. QoS model should allow for fuzziness and various expressions in QoS expression, and unify them for synthetical evaluation, which is out of the existing QoS model's capability.

We classify metrics into four expression types (Liu, 2004): A numerical metric is evaluated using a number. A regional metric is evaluated using a numerical region [*Min*, *Max*]. A lingual metric is evaluated using a term from an ordered finite collection of terms. Different values may be assigned to a metric at the same time with different probability. A graded metric is also evaluated using a term from an ordered finite collection of terms, but a metric must be exactly assigned with a single term at a time. Fuzzy and different expressions must be normalized before the overall QoS evaluation takes place. In our research we use triangle fuzzy number for fuzzy expression standardization.

3. Dynamic

Service QoS metrics are not static-valued as the current QoS model presumes. It may float with service context.

We classify QoS metrics into four categories according to the degree of their dependency on time and service context: Fixed means the metric value is predefined in the service description and remains unchanged until redefined, the service unit price is an example. Statistical means the metric value is recomputed once a service-offering is ended, for example, the service availability. It changes with time. Computational means the metric value depends on service contexts according to a predefined dependent function, for example the service total price is composed of service unit price, service time and a discount. Set means the metric's dependence on service contexts is hard to be modeled, and its estimation must be carried out manually or by some software tools of the server provider. Delivery time after ordering a customerized automobile is an example of a QoS metric that needs to be manually set. Web service provider is responsible for setting these QoS metric values on each service request.

4. Configurable

Service customers have different QoS preferences. When making a choice they are concerned with different subset of metrics and have different requirements. The Web service QoS metrics need to be configured according to the user's preference before being evaluated, which is not supported by the current QoS model and service choice-making process.

In our research Web service QoS model is a configurable evaluation index system. User can select a set of metrics and attach weight on them. So the evaluation of overall service QoS is not fixed, but is flexibly based on user's preference.

Extended Web service QoS description model

Based on the analysis above we set up an extended Web service QoS model as shown in Fig.1.

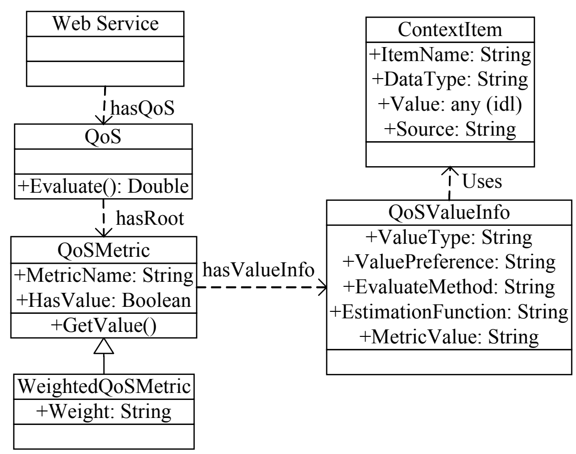

Fig.1 The extended Web service QoS model

Web service QoS is a hierarchical synthetic metric system QoS::=<QoSMetricSet, QoSRoot, RSet>, where QoSMetricSet is a set of QoSMetrics, QoSRoot is the root of the hierarchical structure that stands for the overall service QoS, and Rset is a set of isPartof relationship between QoSMetrics. Relationship isPartof(QoSMetric1, QoSMetric2) means that QoSMetric1 is a sub metric of QoSMetric2.

A QoS metric QoSMetric::=<MetricName, HasValue, ValueInfo>, here the HasValue indicates whether the QoSMetric has a meaningful metric value, and ValueInfo describes the metric's value type and evaluation method (if the metric has a meaningful value). In the QoS hierarchical synthetic index system there are some none-atomic metrics that do not have meaningful values. They serve as abstract metrics that group up sub metrics. All atomic metrics must have meaningful metric values.

ValueInfo describes a QoSMetric's value type and how to get the metric value. ValueInfo::=<Value-Type, ValuePreference, EvaluateMethod, EstimationFunction, MetricValue>, here the ValueType denotes the value expression type, a legal ValueType can be "numerical", "regional", "lingual" or "graded". ValuePreference indicates preferred value in metric value normalization. Available ValuePreference includes "BENEFIT" and "COST". EvaluateMethod denotes how to get a QoSMetric's metric value; a legal EvaluateMethod can be "fixed", "statistical", "computational" or "set". EstimationFunction is a formula for estimating computational metric. The MetricValue is a metric's value, expressed using expression type defined by ValueType.

Each computational metric has an estimation function for dynamically estimating the metric value with service contexts. That function is defined by Web service provider in service design process. Here we define context as follows:

Service context is a collection of ContextItems. ContextItem::=<ItemName, DataType, Value, Source>. Here the Source indicates what kind of information it describes: a context item can describe consumer information, for example <IS_IYHF_Member, Boolean, True, CONSUMER> indicates whether the consumer is a member of IYHF organization, or server information, for example <RoomAvailable, Integer, 17, PROVIDER>, or common information, for example <CurrentDate, Date, 2005-7-4, COMMON>.

A computational metric's estimation function can be defined as a formula containing several ContextItems, nested directly into the QoS definition. It also can be defined as an external function or another Web service taking one or more ContextItems as parameters, so the QoS definition only contains the invoking entrance.

Web service QoS is a part of Web service description. It can be encoded into an XML segment and nested into the OWL-S service description profile. Fig.2 shows the QoS part of a hotel booking service description (simplified for space).

```
<QoSMetric MetricName="GeneralMetrics"  HasValue="False">
   <QoSMetric MetricName="Cost" HasValue="True">
      <ValueInfo>
        <ValueType>Float</ValueType>
        <ValuePreference>COST</ValuePreference>
        <EvaluateMethod>CALCULATE</EvaluateMethod>
        <EstimateFunction>QoS.UnitPrice * QoS.Discount
        </EstimateFunction>
      </ValueInfo>
      <QoSMetric MetricName="UnitPrice"  HasValue="True">
        <ValueInfo>
           <ValueType>Float</ValueType>
           <ValuePreference>COST</ValuePreference>
           <EvaluateMethod>SET</EvaluateMethod>
           <MetricValue>400</MetricValue>
        </ValueInfo>
      </QoSMetric>
      <QoSMetric MetricName="Discount"  HasValue="True">
        <ValueInfo>
           <ValueType>Float</ValueType>
           <ValuePreference>BENEFIT</ValuePreference>
           <EvaluateMethod>CALCULATE
           </EvaluateMethod>
           <EstimateFunction>GetDiscount (Customer-
           Info.CustomerID, CustomerRequire-
           ment.RoomNumber, Requirement.FromDate, Re-
           quirement.ToDate)</EstimateFunction>
        </ValueInfo>
      </QoSMetric>
   </QoSMetric>
</QoSMetric>
```

Fig.2 A Web service QoS description (part)

The Web service QoS model shown above is used to define detailed information about all metrics that form the QoS metrics system of a Web service. When evaluating Web service QoS a weight must be attached to each metric. In our service evaluation model, weight is presented with 5 decreasing grades: "absolutely important", "very important", "important", "general" and "considerable". User can select one of them to define a metric's weight. Weights are processed as graded fuzzy expressions in QoS evaluation computation.

So there are three major steps to be followed when evaluating a Web service QoS based on this QoS model:

(1) Configure the QoS evaluation index system. In this step a set of metrics of interested are chosen from the QoS model and form a QoS evaluation index system, with a weight attached to each metric. Hierarchical relationship must be kept among selected metric.

(2) Obtain metric values. In this step item values

of selected metrics are collected through different ways defined by the item's EvaluateMethod. During evaluation computation (on the server side, broker side or customer side), metric values set by the service provider and the service context from both sides should be gathered.

(3) Fuzzy evaluation. QoS metric values and weights expressed in different fuzzy expressions are converted into a uniform expression before the computation. In our research we used triangle fuzzy number for fuzzy expression standardization. The multifaceted QoS system evaluation is computed recursively. Each none-atomic node's evaluation value is the weighted sum of those of its children's, with the top node's evaluation value standing for the overall QoS evaluation under that context and configuration. A detailed explanation of triangle fuzzy number based fuzzy synthetic evaluation method is included in (Liu, 2004).

WEB SERVICE QOS REQUIREMENT MODEL

Consumer's requirements for different metrics may be various and flexible. Current service requirement description model does not contain a specific QoS part for systematically defining QoS requirements. User can only define constraints on certain metrics, or specify a metric for matched services to be ranked with. Users may prefer to consider the overall performance of many metrics, and impose different restriction level on those metrics. We have designed a Web service QoS requirement model (shown in Fig.3), in which customer requirements on target Web service QoS are classified into QoSConstraints and QoSPreferences.

Constraint is a "hard" restriction on target service that must be satisfied. QoSConstraint::=<MetricName, Operator, Threshold>. An example is a hotel room booking service QoS requirement. (HotelLevel, ≥, 3) means the hotel level should be at least 3 stars. A service QoS requirement may contain several "AND" related QoS constraints all of which a qualified Web service must follow. In service selection QoS constraints are used as filter on functionally matched (or by other matchmaking methods) services.

QoS preference is used to select the "best" service from those that passed functional matchmaking and QoS constraint checking. The service selection handler (a UDDI server for example) ranks those services according to the QoS preference for future user reference. There are two ranking methods. One ranks services according to a QoS metrics sequence, that is, service is ranked by the first metric and if the first metric will not do then the subsequent metric is used. Services may also be ranked according to synthetic QoS evaluation, in which the customer may use some interesting metrics with weights to form a QoS evaluation system for target services to be ranked with. It is clear that synthetic QoS evaluation based ranking is more flexible than QoS metric sequence based ranking.

Service QoS requirement is a part of Web service requirement definition. It can be encoded as an XML segment and nested into the target Web service OWL-S description profile. Fig.4 shows the QoS-Requirement part of a hotel booking service requirement description (simplified for space).

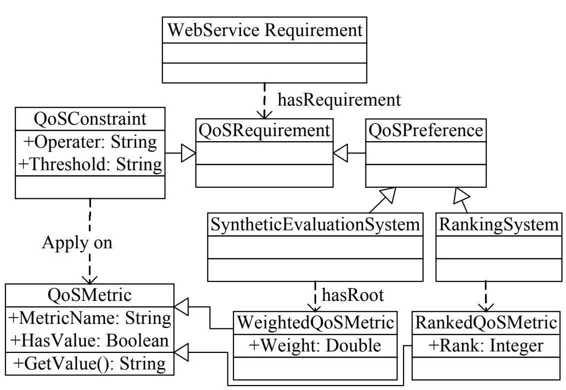

Fig.3 Web service QoS requirement model

```
<QoSRequirement>
    <QoSConstrain ItemName="HotelLevel" Operator=">="
      Threshold="3"/>
    <QoSEvaluationSystem>
        <WeightedItem ItemName="Cost" Weight
          ="VeryImportant"/>
        <WeightedItem ItemName="DistanceToTownCenter"
          Weight="VeryImportant"/>
        <WeightedItem ItemName="HotelLevel" Weight
          ="Important"/>
    </QoSEvaluationSystem>
</QoSRequirement>
```

Fig.4 A Web service QoS requirement definition

QOS BASED INTERACTIVE SERCVICE CHOICE-MAKING

An improved UDDI process model

Current Web service description, discovery and invoking process follows a UDDI protocol. The service provider describes a service properly and registers it on a UDDI server, then waits for invocation. A service consumer expresses service requirement to the UDDI server. The UDDI server checks for suitable services and returns relevant information to the consumer. Then the consumer can connect the provider for service invocation (UDDI, 2004). The process is shown in Fig.5a. We can see that the provider does not participate in the service choosing process, it just waits for invocation after registration; and the UDDI server's participation terminates once a service is selected, it does not care whether the collaboration is successful or not.

When we take dynamic QoS information as an important factor in service choice-making, the above process should be changed. We recognize that more and more Web services now are not simply providing information, but involve complicated business process, long execution time and expensive charges. The service selection and contract making may be the result of a series of cautious interaction and negotiation between the provider and the consumer. Fig.5a shows an improved UDDI process model.

There are three main differences between processes in Figs.5a and 5b:

(1) On a service requirement the UDDI server may connect some service providers to get QoS data. This is because some metrics must be dynamically evaluated using provider side contexts or set by the provider. The UDDI server must send consumer contexts and QoS requirements as parameters to get QoS data from the providers through special interfaces.

(2) When a service is chosen, there will be a service contract set up between the consumer and the provider, in which QoS agreements are explicitly defined. Consumer may verify the QoS agreements during and after service process, pay the predefined charges if agreements are followed or else suffer penal actions.

(3) At the end of service the consumer is expected to feedback an evaluation of it to the UDDI server. The UDDI server traces and collects the most up-to-date service QoS information for future use.

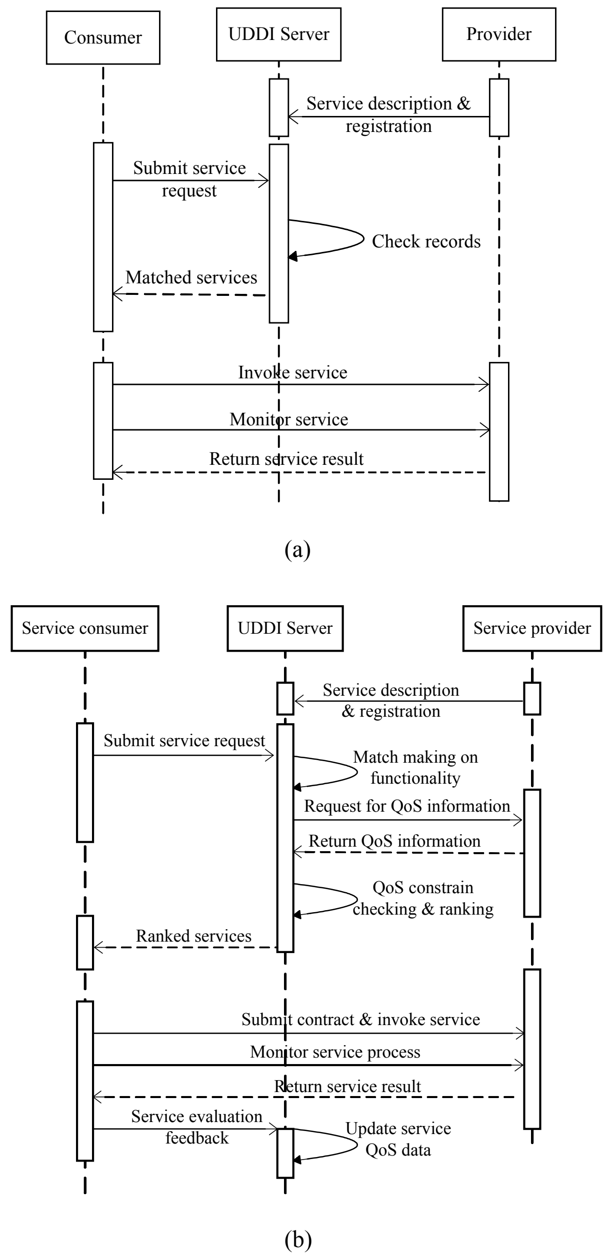

Fig.5 (a) Standard UDDI process; (b) Improved UDDI process

The improved UDDI model places more functional requirements on both UDDI server and service provider. Both of them must participate actively, in supporting more interfaces for QoS related message processing. We regard this as the natural result of service (including UDDI service) competition.

Compared with standard UDDI model, the improved UDDI process model involves much more interactions like message exchange and negotiation among the service consumer, provider and broker. Interaction is an important characteristic of our Web service choice-making process model, which is especially suitable for very discrete pivotal service choice-making.

Contract-net based interaction

QoS requirements definition, metric value acquisition, synthetic QoS evaluation and QoS based service selection form a complex interactive process involving participation of the service consumer, the UDDI server and the service provider. We use a contract-net based process to assure that such an interaction is carried out in good faith.

Contract-net protocol is designed by FIPA for standardizing multi-agent interaction (FIPA, 2002). Now it is used widely in intelligent agent negotiation, cross-organization process collaboration, workflow task distribution and automatic resource workload balancing (Ermolayev *et al*., 2004; Zhang, 2004). The contract-net based Web service choice-making interaction process (Fig.6) consists of the following phases:

(1) Defining requirements. Service consumer defines and submits target service requirements to the UDDI server. QoS requirement is defined according to the service QoS definition stored at the UDDI server side, so interaction is needed to transfer QoS model of the required service type to the consumer.

(2) Service requirement processing at the UDDI server, who uses target service description to find functionally (or by other matchmaking mechanisms) matched services. Then for each of the matched services the UDDI server checks whether all metrics involved in the QoS requirement can be evaluated locally. If provider participation is needed, a CallForProposal message is formed and sent to the service provider. The message contains requested QoS metrics, customer contexts and a time deadline. The UDDI server must support a CFP interface to form and send such a message and the service provider must support a CFPHandler interface to receive and process the message. An example CallForProposal message is shown in Fig.7.

(3) Service configuration and QoS data submission. CFPHandler at the service provider side parses

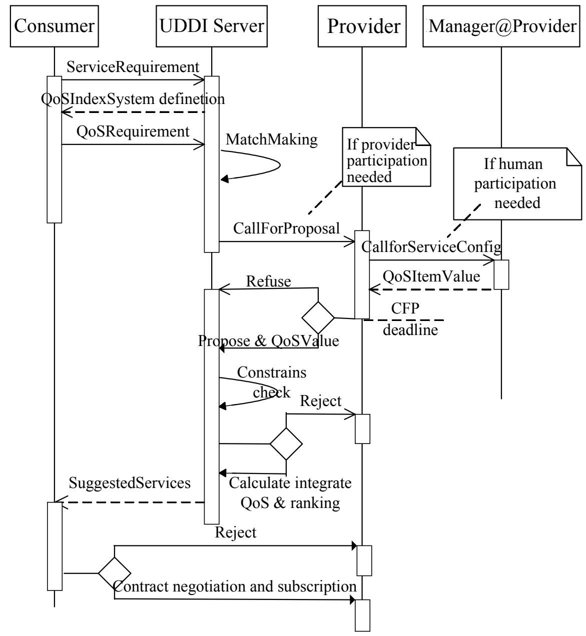

Fig.6 Contract-net based Web service choice-making interaction process

```
<CallForProposal>
  <CFPID>www.aUDDIServer.com_CFP_2003-2-28_#00203
  </CFPID>
  <From>www.aUDDIServer.com</From>
  <To>www.aSite.com/CFPHandler.wsdl</To>
  <TimeStamp>20:35:04 2005-2-28</TimeStamp>
  <DeadLine>24:00:00 2005-3-1</DeadLine>
    <WSDL>www.aSite.com/WebServices/BookRoom/aServi
      ce.WSDL</WSDL>
  <CustomerInfo>
    <IsSomeAllianceMember>Yes</IsSomeAllianceMember>
    <Name>aTravelAgency</Name>
  </CustomerInfo>
  <CustomerRequirement>
    <RoomType>Standard</RoomType>
    <NumberOfRoom>1</NumberOfRoom>
    <FromDate>2005-5-1</FromDate>
    <ToDate>2005-5-7</ToDate>
  </CustomerRequirement>
  <Constrains>
    <Constrain QoSIndexName="Hotellevel" Operator=">="
      Threshold="3"/>
  </Constrains>
  <Preference>
    <ConsernedQoSItem Index="Cost" Weight
      ="VeryImportant"/>
    <ConsernedQoSItem Index="HotelLevel" Weight
      ="Important"/>
    <ConsernedQoSItem Index="DistanceToTownCenter"
      Weight="VeryImportant"/>
  </Preference>
</CallForProposal>
```

Fig.7 A CallForProposal message

the CallForProposal message, gets required context data and evaluates requested QoS metric. It may be a totally automatic data collection and calculation process, or need human participation if there are metrics to be manually set. QoS data must be submitted to the UDDI server through a proposal message (shown in Fig.8) before the time deadline. The provider may also send a Refuse message to the UDDI server or simply ignore the CallForProposal message if is not interested in this service request.


```
<Proposal>
   <InResponsefCFP>www.aUDDIServer.com_CFP_2003-2-
        28_#00203</InResponsefCFP>
   <From>www.aSite.com/CFPHandler.wsdl</From>
   <To>www.aUDDIServer.com</To>
   <TimeStamp>09:23:43 2005-3-1</TimeStamp>
   <WSDL>www.aSite.com/WebServices/BookRoom/aService.
        WSDL</WSDL>
   <QoSInfo>
     <QoSItem Name="Hotellevel" Value="3"/>
     <QoSItem Name="DistanceToTownCenter" Value="0,5"/>
     <QoSItem Name="Price" Value="400"/>
   </QoSInfo>
</Proposal>
```


Fig.8 A Proposal message

(4) QoS evaluation. If all CallForProposal messages are answered or the deadline is reached, the UDDI server uses QoS data (recalled from self recordings or retrieved from service providers) to process consumer QoS requirement. Firstly QoS constraints are checked to filter out unwanted services. Then the synthetic QoS of each remained service is calculated according to the QoS preference. UDDI server returns a set of suggested services to the customer, ranked with the overall QoS evaluation. The detailed metric information may also be included on customer request.

(5) Service selection and invocation. The consumer selects one from the suggested services; connects with the provider for service contract negotiation and subscription. For each rejected service the consumer just sends a reject message.

Fig.9 shows the final QoSAgreement on a QoSMetric, it is part of the QoSAgreement in a Web service contract. A complete service contract may cover contract metadata, general information about provider and consumer, invocation of interfaces and parameters, QoS agreements and actions on violation, digital signature and safety protocol, etc. A detailed discussion can be found in (Qian, 2005).


```
<QoSAgreement>
     <ItemName>HotelLevel</ItemName>
     <ResponsibleSide>PROVIDER</ResponsibleSide>
     <Operator>>=</Operator>
     <ThretholdValue>3</ThretholdValue>
     <CheckPolicy>ONSUBMITTING</CheckPolicy>
     <ActionOnViolation>
          <CancleExecution>YES</CancleExecution>
          <CanclePayMent>YES</CanclePayMent>
          <CreditLevelReduce>3</CreditLevelReduce>
     </ActionOnViolation>
  </QoSAgreement>
```


Fig.9 QoS agreement on a QoSMetric

In the above interaction the three parties need to share a common QoS definition on a Web service. We suggest this definition be kept at the UDDI server side. Any Web service description registered on that UDDI server and any service requirement submitted to it must follow that QoS definition.

The interaction process may be adjusted as follows: (1) To reduce QoS requirement definition efforts, the consumer can predefine QoS requirements into different profiles. The profiles may be kept either at the consumer side or UDDI server side, and used according to context, see (Liu *et al*., 2004) for a detailed discussion; (2) QoS can be evaluated either at the UDDI server side, or the provider side, as long as required contexts (consumer contexts and provider contexts) can be obtained; (3) If none of the functionally matched Web service can pass QoS constrain checking, the UDDI server may send a message to the consumer for requirement adjustments and begin next round selection.

CONCLUSION

More and more organizations are delivering business transactions as Web services to the Internet for dynamic B2B collaboration. Increasing service number will cause competition among functionally similar services, making QoS a most important factor in service selection. Systematical methodology on Web service QoS description and QoS based Web service choice-making is needed. We point out in this paper that existing standards and semantic models for

Web service description and selection are limited in QoS information expression and utilization. A Web service QoS model is set up. The evaluation algorithm is introduced. We also set up a QoS requirement model for flexibly expressing user's requirement on target service's QoS. An improved UDDI process is suggested for dynamic QoS based interactive service choice-making. Usage of QoS information for all participations through the whole service process is discussed briefly. The solution in this paper provides stronger QoS based supports to Web service based dynamic business collaboration.

Our future research work will be focused on QoS based service process monitoring, QoS context based process management decision support and QoS based process improvement.

References

Busse, R., 2005. Cooperative Support Service—Quality of Service. Http://www.crossflow.org/public/wsfoils/QoS.ppt.

Cardoso, J., Sheth, A.P., Miller, J.A., Arnold, J., Kochut, K.J., 2004. Quality of service for workflows and Web service processes. *Web Semantics: Science, Services and Agents on the World Wide Web*, **1**(3):281-308. [doi:10.1016/j.websem.2004.03.001]

Chandrasekaran, S., Silver, G., Miller, J.A., Cardoso, J., Sheth, A.P., 2002. Web Service Technologies and their Synergy with Simulation. Proceedings of the 2002 Winter Simulation Conference (WSC'02), San Diego, California, p.606-615.

DAML-S (The DAML Services Coalition), 2005. DAML-S: Semantic Markup for Web Services. Http://www.semantic-Web.org/SWWS/program/full/paper57.pdf.

Ermolayev, V., Keberle, N., Plaksin, S., Kononenko, O., Terziyan, V., 2004. Towards a framework for agent-enabled semantic Web service composition. *International Journal of Web Service Research*, **1**(3):63-87.

FIPA (Foundation for Intelligent Physical Agents), 2002. FIPA Contract Net Interaction Protocol Specification. Http://www.fipa.org/specs/fipa00029/SC00029H.html.

Gonzalez-Castillo, J., Trastour, D., Bartolini, C., 2001. Description Logics for Matchmaking of Services. Proceedings of the KI-2001 Workshop on Applications of Description Logics.

Liu, Y.H., 2004. Research and Realization of Configurable Fuzzy Synthesized Evaluation System. M.D Thesis, Shanghai Jiao Tong University (in Chinese).

Liu, H., Han, Y.B., Li, G., Zhang, C., 2004. Achieving Context Sensitivity of Service-Oriented Applications with the Business-End Programming Language VINCA. Proceedings of the Third International Conference on Grid Computing (GCC 2004), p.89-96.

Mani, A., Nagarajan, A., 2002. Understanding Quality of Service for Web Services: Improving the Performance of your Web Services. Http://www-128.ibm.com/developerworks/Webservices/library/ws-quality.html.

OWL-S, 2004. Semantic Markup for Web Services. W3C Member Submission. Http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/.

Paolucci, M., Kawamura, T., Payne, T.R., Sycara, K., 2003. Semantic Matching of Web Services Capabilities. Proc. of the First International Semantic Web Conference.

Qian, Q., 2005. Research on Several Key Technologies of Resource Management in Grid Computing. M.D Thesis, Shanghai Jiao Tong University (in Chinese).

Trastou, D., Bartolini, C., Gonzalez-Castillo, J., 2001. A Semantic Web Approach to Service Description for Matchmaking of Services. Proceedings of the International Semantic Web Working Symposium (SWWS).

UDDI, 2004. UDDI Version 3 Specification. Http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm#uddiv3.

W3C, 2003. QoS for Web Services: Requirements and Possible Approaches. W3C Working Group Note. Http://www.w3c.or.kr/kr-office/TR/2003/NOTE-ws-qos-20031125/.

W3C, 2004. Web Services Description Language Version 2.0. Http://www.w3.org/TR/2004/WD-wsdl20-20040803.

Wombacher, A., Mahleko, B., Neuhold, E., 2004a. IPSI-PF: A Business Process Matchmaking Engine. Proc. of Conference on Electronic Commerce (CEC 04), San Diego, California, p.137-145.

Wombacher, A., Fankhauser, P., Mahleko, B., Neuhold, E., 2004b. Matchmaking for business processes based on choreographies. *International Journal of Web Services*, **1**(4):14-32.

Zhang, X.G., 2004. Research on Environment and Techniques for Project Process Management. Ph.D Thesis, Shanghai Jiao Tong University (in Chinese).