Research on a Kind of QoS-Sensitive Semantic Web Services Composition Method Based on Genetic Algorithm

Cao Hongjiang, Nie Guihua, Chen Donglin
School of Economics, Wuhan University of Technology, Wuhan, P.R.China, 430070
(E-mail: chjnewfree@yahoo.com.cn)

Abstract In the process of QoS-sensitive business, the user not only cares about whether the function of the process will be finished successfully, but also pays great attention to the QoS of the whole process. Thus, how to make a choice among the candidate web services so that the selected ones can not only finish the assigned task and conform to the local restriction, but also can cooperate with other web services and optimize the QoS of the process, becomes a key problem. In order to solve this problem, an ontology model of the web service quality attributes and an approach based on the generic algorithm were proposed. The experimental results show the effectiveness of this method.
Key words Semantic web services; Ontology model; Qos; Generic algorithm

1 Introduction

With the continuous development of semantic web services, dynamic service composition which based on web services as an important method to achieving a flexible, rapid integration become a new hotspot. At present, as the number of service component growth rapidly, how to select the results to meet the needs of the user QoS requirements quickly in massive result is one of the main objectives of the development of web service composition. In the Web service selection, in addition to functional attributes, the user can also in some non-functional properties of set constraints to complete, namely, by constraint QoS (quality of service) property to choose. QoS contain the attributes are: price, response time, availability, and reliability, and so on[1,2]and the Web service providers will often set a range of QoS property value, so as to provide reference for potential users.

In the process of QoS-sensitive business, the user not only cares about whether the function of the process will be finished successfully, but also pays great attention to the QoS of the whole process. Thus, how to make a choice among the candidate web services so that the selected ones can not only finish the assigned task and conform to the local restriction, but also can cooperate with other web services and optimize the QoS of the process, becomes a key problem. We know that QoS query message format provided by the different service providers and service users is likely to be a variety of In order to provide QoS machine-readable information, the establishment of a QoS ontology model to standardize the description of the information QoS information is imperative. This paper presents an ontology model of web service quality attributes and genetic algorithms methods to select Web services effectively, and through experiments proved the effectiveness of the method. Compared with existing methods, this paper presents the method has a better universal.

2 Related Work

With the popularity of Web services in recent years, there has been a lot of research work in the area of use of Web service composition to obtain business processes. L Aversano[3] et al. proposed to use existing Web services, through the dynamic combination of them to be business processes
As the Internet environment and dynamic nature of the inherent randomness, people tend to demand through a combination of Web services have been in business processes to meet certain service level agreement SLA (service level agree-ment), which satisfy the QoS constraints. Therefore, how to solve the QoS-sensitive issue of Web service composition, the researchers made a lot of discussion.

One of typical model is used meta-process and meta-services concept, a meta-service representative a group of similar business functions of the Web services. Thus, concrete services that corresponding to the same meta- service can be replaced with each other. Meta-process is composed by the meta services to an abstract process model. In such paradigm, some researchers have proposed methods based on integer programming[4, 5]; this method assumes that the constraint conditions and objective function has a linear character. Zeng's research [4] focus on overhead response time, effectiveness and reliability QoS attributes. And in the course of calculation the objective function used the logarithmic strain. Although, Zeng think their model can extend the other similar behavior property, in reality, the process required to meet the constraints more complex, often do not have the linear characteristics. Such as service dependencies and user preferences. Therefore, Zeng proposed model does not have a good universal. Although Aggarwal [5] referred to the needs of such non-linear constraint, but did not give the appropriate solutions. Compared with such methods based on integer programming, the method proposed in this article based on genetic algorithms that can handle more forms of restraint, with a better universal. Such paradigm is used in another method of artificial intelligence query strategy. This methods from artificial intelligence constraint logic CLP (constraint logic programming), it has the advantage of strong modeling capabilities; Yu T and Berbner R's[6][7] article is the study of this approach. As the CLP has the characteristics of transmission constraints, therefore, these methods are more efficient query, but the cost of its own is relatively large. In the QoS-sensitive Web service composition process, but also due to fewer constraints in general are relatively simple. Therefore, this article considers that the adoption cost of a relatively small. Genetic algorithm is more suitable. As the genetic algorithm itself does not provide mechanisms to describe the binding, so a lot of research work to achieve through the penalty function method. For QoS- sensitive Web service composition problem[8, 9, 10].

3 The Qos Ontology Model

Different service providers and service users with the QoS query message format is likely to be varied. In order to provide QOS machine-readable information, the establishment of a QoS ontology model to regulate the QOS information description is necessary [11].This paper presents an ontology model of the web service quality attributes, to a unified description of the various service quality attributes. Ontology is structured as follows:

Top-level ontology

Top-level ontology mainly definitions some of the advanced concepts related to QOS models .For each concept and the relationship between these concepts will be further explanation in this article, QosProfile is the most important concept, which is the main body of Qos model.

QosAttribute: used to indicate Qos parameter, which is the super class of all Qos parameters， In practical situations, it contains the description of the price of Qos parameters describing the response time of the Qos parameters, describing the reliability of the Qos parameters, etc. QosProfile through its hasQosAtrribute attributes can define all the Qos parameters of a service

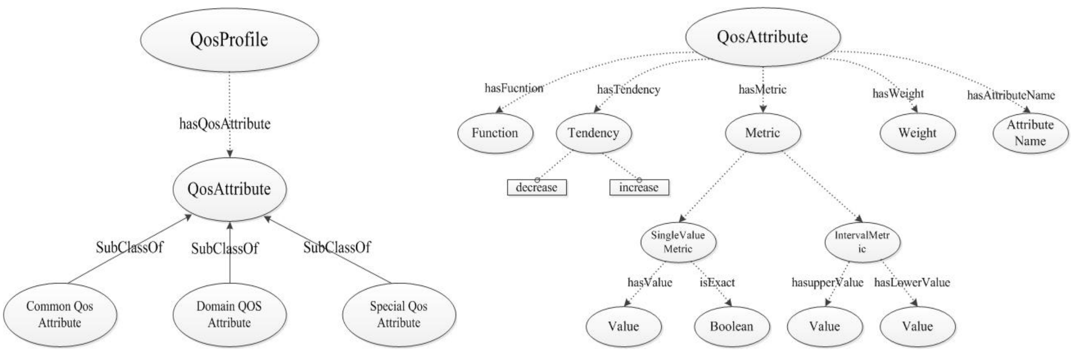

Figure 1 QoS Ontology Model

In order to meet the needs of Web services matching and choice, QosAirribute derived for the three subclasses: CommonQosAttribute, DomainQosAttribute, and SpecialQosAttribute.

CommonQosAttribute refers to all WebServiee need to possess Qos parameters, such as price, response time, reliability, reputation and so on.

DomainQosAttribute refers to the Qos parameters relevant to the domain, such as the packaging of the books bought is "hardcover" or "stripped-down" and so on.

SpecialQosAttribute used to describe service-specific QoS parameters. For service providers in the publication of the QoS information or Service requestor proposed QoS requirements.

Accordingly, QosProfile through the three subattributes of hasQosAttribute: hasCommonQos Attribute, hasDomainQosAttribute, hasspecialQosAttribute link with three subclass.QosAttribute is the super class used to describe all the parameters of QoS. Qos parameters of the class can b e expressed as metrics, monotonicity, and share of weight in the QosProfile and so on. If it is

composite Qos parameters, this model can also describe it with other Qos parameters.
QoS calculation of business processes

In order to complete the Web services choices according to QoS, we need to describe how from a single Web service QoS attribute value calculating business processes.

For a given specific Web services, business processes, Table 1 shows the calculation of the entire process of the QoS rules[12], and lists the calculation function corresponds to the variety of common processes structure and common QoS properties. Meanwhile, in the last line of Table 1 also gives the process the calculation of user-defined (such as the domain dependent properties) property rules. In this article, due to space limitations, only by time, cost, QoS, etc. these common attributes, to demonstrate, validate the method proposed in the text.

Table1 1 The Model for Computing the QoS of Services Compositon

Qos Attribute	*Sequence*	*Switch*	*Parallel*	*Loop*
Time	$\sum_{i=1}^{m} T(t_i)$	$\sum_{i=1}^{m} P_{ai}T(t_i)$	$\max\{T(t_i)_{i\in(1...p)}\}$	$kT(t)$
Cost	$\sum_{i=1}^{m} C(t_i)$	$\sum_{i=1}^{m} P_{ai}C(t_i)$	$\sum_{i=1}^{p} C(t_i)$	$kC(t)$
Availability	$\prod_{i=1}^{m} A(t_i)$	$\sum_{i=1}^{m} P_{ai}A(t_i)$	$\prod_{i=1}^{P} A(t_i)$	$A(T)^k$
Reliability	$\prod_{i=1}^{m} R(t_i)$	$\sum_{i=1}^{m} P_{ai}R(t_i)$	$\prod_{i=1}^{P} R(t_i)$	$R(t)^k$
User-defined	$f_s(F(t_i))$ i□$\{1...m\}$	$f_R((P_{ai},F(t_i)))$ i□$\{1...n\}$	$f_F(F(t_i))$ i□$\{1...p\}$	$f_L(k,F(t))$

4 A Method Based on Genetic Algorithm

Genetic algorithm is based on the mechanism of natural selection and natural genetic search algorithm, which is an effective way to solve the optimization problem. Using genetic algorithm for solving optimization problems, we first need to select some individuals in the feasible domain is encoded as a chromosome to form the first-generation groups, and calculate the fitness of each chromosome. Next, the algorithm uses the selection mechanism randomly selected from the group of chromosomes as the samples before reproductive process. The probability of a chromosome being selected is proportional to its fitness. In the next reproduction process, Genetic algorithm uses crossover and mutation operator to the selection of chromosomes after the reproduction processing.

Genetic algorithm does not require QoS combination operations (also including the objective function and constraints) have linear characteristics. This makes our approach more suitable for (even user-defined) QoS attributes, without the need for linearization. In the QoS-sensitive Web service composition problems, the chromosomes described as an integer array of $S = (S_1, S_2, ... S_n)$. The S_i array corresponds to each element of business processes a sub-task. Each element contains an index, the index pointing to the elements in a group with the same or similar functionality as Web services in the selected Web services.

4.1 Dynamic fitness function

According to characteristics of QoS-sensitive Web service composition problem, the fitness function of chromosomes requires some QoS attribute value to the maximum, while leaving other parts of the QoS property value of the minimum. Similarly, if the user-defined QoS properties of a specific area, then the fitness function would also like to make further adjustments under the circumstances. In addition, the fitness function should be able to suppress the chromosome does not meet the constraints to guide the direction of them to evolve to meet the constraints, as well as violations of the different QoS constraints given different weights:

$$Vl_i(g) \le 0, \ i = 1,2....n \tag{1}$$

Where represents the violation of section i of constraints on chromosomes brought about the penalty value of fitness.

D (g) is defined as: the impact of current chromosome breach of the various types of constraint

$$D(g) = \sum_{i=1}^{n} Vl_i(g) y_i \tag{2}$$

In this formula, if chromosome violates the constraint, then y_i is 1, otherwise y_i is 0.

Because different metrics of its range is usually not the same. Values are often differences in the direction, if the raw metric data directly to calculate, could make between the results are not comparable. Therefore, the need for data pre-processing, mapping them all to the same range, and a unified direction for all metrics values. For example, metrics M's range is [min, max], an instance of the raw measurement data is v, If the metrics values upward direction, that is, the greater the value of v, said the better the quality of service(such as Web service's credibility), then according to the formula (3) to be normalized it

$$V' = \begin{cases} \frac{v - min}{max - min}, & min \le v \le max \\ 1, & v \ge max \\ 0, & v \le min \end{cases} \tag{3}$$

Conversely, if the metrics values downward direction, that is, the smaller the value of v, said the better the quality of services(such as the Web service operation of the response time and delay time), then according to the formula (4) to be normalized it.

$$V' = \begin{cases} \frac{max - v}{max - min}, & min \le v \le max \\ 1, & v \le min \\ 0, & v \ge max \end{cases} \tag{4}$$

After standardized processing, with all metrics of the data fall into [0,1] interval, and the direction of their values are upward. These values, in the QOS ontology model presented in this paper have a corresponding classes and attributes to represent。

According to common QoS properties in the Table 1,the fitness function is defined as follows:

$$F(g) = \frac{\omega_1 A(g) + \omega_2 R(g)}{\omega_3 C(g) + \omega_4 T(g)} + \omega_5 D \tag{5}$$

In this formula, ω_1, ... ω_5 are positive real number, as a parameter. ω_1, ...ω_4 on behalf of the user for a specific QoS attributes to set the weights, ω5 is the overall weight of violating the constraints. Usually, if a penalty factor with a high weight, In the evolutionary process, could make chromosome which close to the optimal solution but violate the constraint discarded in the early evolution. To address this issue, this paper presents dynamic punishment strategies: the penalty weight increases along with the descendant number increases. To make such a chromosome could be retained, through several generations of evolution; the entire group will meet constraints. Therefore, dynamic fitness function can be defined as:

$$F(g) = \frac{\omega_1 A(g) + \omega_2 R(g)}{\omega_3 C(g) + \omega_4 T(g)} + \omega_5 D(g) \frac{gen}{gen_{max}} \tag{6}$$

Where gen represents Current evolutionary times, gen_{max} represents Maximum number of evolutionary

4.2 Termination conditions of genetic algorithms

Specify a maximum number of evolutions are a more efficient way. Therefore, in QoS-sensitive issue of Web service composition, using the termination condition as follows:

set the maximum number of offspring gen_{max} , iterate until the constraint (D (g) = 0) have been satisfied. If have already reached the maximum number of generations, but the constraints have not been met. Then through the algorithm to determine, we think that there is no corresponding solution. Or until the optimization of the chromosome in the offspring is not changed so far.

In this way, QoS-sensitive Web service composition problem can be described by means of chromosomes. And through repeated chromosome selection, crossover and mutation operators to continuously optimize the results; until the termination condition of the algorithm is true, we can get the optimal solution of Web service composition.

5 Experiment

This section will be through experiments analyze and validate the effectiveness of the method proposed in this paper. And through analysis of the offspring produced by genetic algorithm is to observe the changes in chromosome fitness. Thus Comparison of static and dynamic fitness functions.

We use elite genetic algorithm, in which each evolved to retain only the best two chromosomes. Crossover probability 0.7, mutation probability of 0.01, population size of 100, the selection mechanism used switch wheel method. The implementation of genetic algorithms for each experiment 50 times and record the average of the results. In order to observe the different fitness function, to require the evolution to achieve preset number of times, if the descendant has not improved, then stops the algorithm. Business processes used in the experiment included 25 service calls, of which there are 16 different tasks, For each task. Were observed when the number of optional web services are 5,10,15,20,25 situation at the time。Figure 2 shows, the constraint is to require the cost and service response time to reach the optimum, the figure shows the genetic algorithm can find solutions that satisfy the constraint.

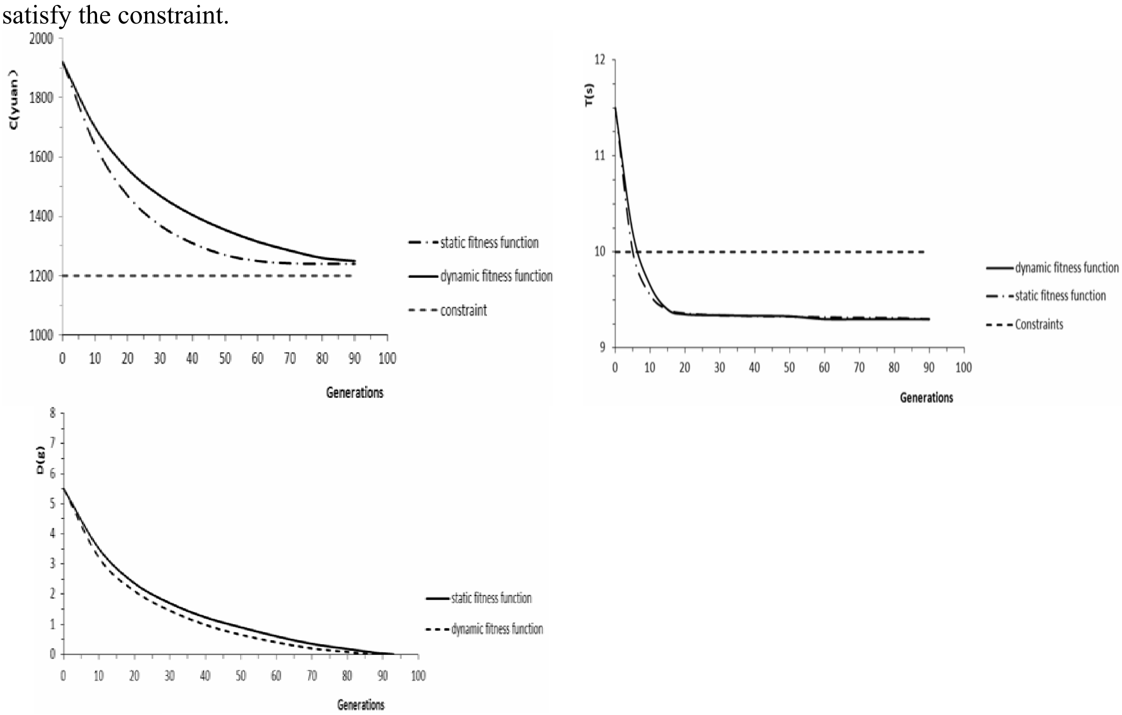

Figure 2 Comparison of Different Fitness Functions for Different Parameter

6 Conclusion

This paper presents a method based on genetic algorithms to solve the QoS-sensitive Web service composition problem. Through the use of genetic algorithms for business processes in the task select the appropriate Web services, and binding, So that business processes can not only satisfy the constraints also make specified QoS attribute to achieve optimal. Experimental data is to prove the effectiveness of this method. Meanwhile because of genetic algorithm is not subject to linear constraints, therefore has a better universality and the extendibility. Presented in this paper using dynamic fitness function is to reduce the influence of Constraints to fine chromosome, increasing the competitiveness of fine chromosomes. In addition, the proposed QOS ontology model enables QOS attributes to get a unified description. In future work, will further optimize the fitness function; and consider the adoption of genetic algorithm to solve the automatic generation of meta-process in semantic web service composition process.

References

[1] ISO. UNE-EN-ISO, ISO 8402 (Part of the ISO 9000 2002): Quality Vocabulary[S]

[2] IT Recommendation. E800 Quality of Service and Dependability Vocabulary[S]

[3] Aversano L, Canfora G, Ciampi A. An Algorithm for Web Service Discovery Through Their Composition[C]. Proceedings of the 2004 IEEE International Conference on Web services (ICWS). San Diego, California, USA: IEEE Computer Society, 2004:332-339

[4] Zeng LZ, Benatallah B, Ngu AHH. QoS-Aware Middleware for Web Services Composition[J]. IEEE Trans. on Software Engineering, 2004, 30(5):311－327

[5] Aggarwal R, Verma K, Miller J, et al. Constraint Driven Web Service Composition in METEOR-S[C]. 2004 IEEE International Conference on ServicesComputing, (SCC). Shanghai, China: IEEE Computer Society, 2004:23-30

[6] Yu T, Lin KJ. Service Selection Algorithms for Composing Complex Services with Multiple QoS Constraints[C]. ICSOC, Heidelberg: Springer-Verlag, 2005:130－143

[7] Berbner R, Spahn M, Repp N, Heckmann O, Steinmetz R. Heuristics for QoS Aware Web Service Composition[C]. ICWS, Chicago: IEEE Computer Society, 2006:72－82

[8] Liu SL, Liu YX, Zhang F, Tang GF, Jing N. A Dynamic Web Services Selection Algorithm with QoS Global Optimal Web Services Composition[J]. Journal of Software, 2007, 18(3):646−656 (In Chinese)

[9] Canfora G, Penta MD, Esposito R, Villani ML. An Approach for QoS-aware Service Composition Based on Genetic Algorithms[C]. GECCO, Washington: IEEE Computer Society, 2005:1069－1075

[10] Zhang CW, Su S, Chen JL. Genetic Algorithm on Web Services Selection Supporting QoS[J]. Chinese Journal of Computers, 2006, 29(7):1029－1037 (In Chinese)

[11] Li Z, Shuang K, Yang FC, Su S. A Decision Making Approach for Semantic Web Services Selection with Heterogeneous QoS Values[C]. CIICT, 2007: 783−790

[12] Cardoso J, ShethA P, Miller JA, Arnold J, Kochut KJ. Modeling Quality of Service for Workflows and Web Service Processes[J]. Web Semantics Journals Science, Services and Agents on theWorld Wide Web Journal, 2004, 1(3): 281-30