

Study on QoS Driven Web Services Composition

Yan-ping Chen, Zeng-zhi Li, Qin-xue Jin, and Chuang Wang

Department of Computer Science and Technology,
Xi'an Jiaotong University, 710049, Xi'an China
yanping@tom.com, lzz@mail.xjtu.edu.cn
wangchuang1103@sina.com

Abstract. Providing composed Web Services based on the QoS requirements of clients is still an urgent problem to be solved. In this paper, we try to solve this problem. Firstly, we enhanced the current WSDL to describe the QoS of services, and then gave a way to choose the proper pre-exist services based on their QoS.

1 Introduction

Web Service is a hotspot in the research of SOA as the best realization of SOA [1]. There are several problems in Web Service frame needed to be studied such as service composition, data integration of services, security of services¹, etc. However, all these existing problems are aroused by the composition [2]. The task of composition is to combine and link existing Web services to create new Web services. Lots of researchers have paid their attentions to the service composition. SELF-SERV [3] is a platform that can provide composed service, but SELF-SERV emphasizes only on functional composition and ignores the QoS requirements of clients. A way to compose components based on QoS is proposed in [4], but it gives no details about how to describe the QoS of a service. In distributed environment, different service components may possess the same functions, and references [5-7] provide WSOL (Web Service Offering Language) that can change services at run time by dynamic switching in different service constrains. DAML-S [8] aims to define ontologies for service description that will allow software agents to reason about the properties of services.

The above approaches solve some issues in composed Web Services from different views, but none of them can give a whole and realizable way to provide composed Web Services based on clients' QoS. To describe QoS requirements of clients, we proposed a new Web Service description language-EWSDL (Enhanced Web Service Description Language), and optimized concurrent Web Service frame with an evolved role of provider to meet non-functional requirements of clients to realize compositions.

The remainder of this paper is organized as follows. In section 2 we give an enhanced Web Service description language-EWSDL and the composition selection algorithm. The paper is concluded in section 3.

2 Web Service Description Model

2.1 QoS Property of Web Services

There are many failed projects because of ignoring non-functional properties of software [9]. Definitions of non-functional properties of software are various and have no unified definition [10]. However, there are still some broadly accepted views about the non-functional properties. Generally speaking, the non-functional properties of software should include performance, reusability, maintainability, security, reliability, and availability. In Internet, we call the non-functional property as QoS property [13].

If a service provider only takes functional requirement of clients into account without considering non-functional requirements when providing the service, the provided service will not be accepted by clients at runtime. How to assure QoS of services is a stringent problem for service providers. Now there are two different ways to solve this problem. One is syntax-based approach extending the current WSDL with more elements, and the other is to develop a new language based on semantic, such as XL, OWL-S. Both ways intend to add more information of the service when describe the service. But the language based on semantic is more complex. So, in this paper we extend WSDL to support more QoS descriptions.

WSDL2.0 describes three functional properties of Web Service: what to do, where to do, and how to do. However, WSDL cannot describe non-functional properties of services [11]. So, WSDL cannot be used to automatic service lookup and large-scale reuse and composition. Because of these defections, WSDL should be extended to include more information. In addition, only when including the non-functional properties it is integrity.

Essentially, how to describe and quantitative analyze non-functional properties of software is a complex problem and is still needed to solve, so how to describe the non-functional properties of Web Service has no existing way. Many researchers have paid their attentions to this problem. V.Tosic proposed a Web Service Offering Language[5-7](WSOL) to extend WSDL by adding new mark, such as price, time, etc. In essence, WSOL only provides some disperse, predefined, and limited property plate, so it lacks of flexibility. Some non-functional properties are given in [12], such as availability, channels, charging style, settlement, payment obligations, etc. The author also indicates that the non-functional properties of Web Service are actually the QoS properties actually. A model for Web Service discovery with QoS is given in [13], it gives some definitions of QoS properties, but these properties are not from the view of managing composed Web Services. In this paper we explore an enhanced Web Service description language according to the need of managing the QoS properties of composed Web Service.

2.2 Description Model of Web Service

Definition 1. Description model of Web Service. Let S be the description model of Web Service which can be expressed as $S = \{Func, QoS\}$, where, $Func$ denotes the functional properties of S , and QoS denotes the non-functional properties of S . The functions of Web Service is described by the portTypes of WSDL, in order to describe the QoS properties, a $tOperationInst$ element is added to $tport$ element of WSDL.

Considering that the QoS properties of Web Service should be independent to the domain of services and they also should be quantifiable, we use a vector in seven dimensions to describe the QoS of a Web Service (both element service¹ and composed service)

responseTime represents the response time of a element service;

availability represents the probability of the service can be used correctly;

concurrency represents the maximum ability to support concurrent transactions;

expireTime represents the expire time of a service, and the reliability of a service can be ensured before the expireTime,

price represents the money the client should pay for this service,

fine represents service provider(client) should compensate client(provider) for breaking the contract between them. Commonly, the **fine** has a linear relation with the **price**,

securityLevel represents the security level of a service.

For a composed service CS it also possesses the same parameters, but in a CS, these QoS parameters cannot be calculated by a simple mathematical function such as sum. For example, responseTime of CS is not the sum of responseTime of every element services when existing concurrent process in CS. Followings give a way to calculate the QoS parameters of CS.

ResponseTime, considering that there exists concurrent service in composed Web Service, response time of composed service is not the sum of responseTime of all the element services, it should be the sum of critical route in execution process. CPA^[15] is an algorithm to find the critical route.

$$\text{ResponseTime} = \text{CPA} (\text{Service}_1, \text{Service}_2, \dots, \text{Service}_m),$$

$$\text{Availability} = \prod_{i=1}^m \text{availability}_i,$$

$$\text{Concurrency} = \min (\text{concurrency}_1, \text{concurrency}_2, \dots, \text{concurrency}_m),$$

$$\text{ExpireTime} = \min (\text{expireTime}_1, \text{expireTime}_2, \dots, \text{expireTime}_m),$$

$$\text{Price} = \bigcup_{i=1}^m \text{price}_i,$$

$$\text{Fine} = A * \text{Price}, \text{ the client and the provider negotiate to get } A,$$

$$\text{SecurityLevel} = \min (\text{securityLevel}_1, \text{securityLevel}_2, \dots, \text{securityLevel}_m).$$

These QoS properties are not independent, for example, a correlation between price and fine. This is why composition according to clients' QoS requirements is a difficult problem.

2.3 Composition Selection Algorithm

Definition 2. Service Class. Let A be a set of Web Services which have been registered into the Composer, and these services can be divided into several subsets according to

¹ Web Services that participate in the composition are all called element services. In this paper, there are no essences differences between the element service and the composed service except for the granularity.

the different functions, and each Web Service of A belongs to one subset at least, and the differences among services which belong to a same subset only are the QoS properties. Noticed, all the Service Classes form an overlay of A not a partition.

It is a hard work for the service composer to provide a composition Web Service to clients according to their QoS requirements based on the following two reasons: first, there are no standard measurement of all these QoS properties, second, QoS properties are not independent, and one QoS property may favor/feed back others.

So, the multi-dimensional of the QoS properties cannot be merged into one dimension, we proposed a Composition Selection Algorithm, which relies on the following assumptions:

- 1) Clients' SLA is the QoS of the composed Web Services;
- 2) Composed services and element services use the same glossary;
- 3) The required QoS of clients can be pressed as a vector (ResponseTime *_c ,

Availability *_c , Concurrency *_c , ExpireTime *_c , Price *_c , Fine *_c , Security Level *_c);The requirements of clients usually fall into a range not a certain value, so we must define the relation between composed service and the required QoS (requirement) of clients;

4) The composition logic is predefined, and the aim is to simplify the composition and pay more attention to QoS.

There are many ways to solve the multi-object programming [16,17,18], and in our prototype, we used the following method.

Step1, Construct programming matrix $A = (a_{ij})_{n \times m}$, and normalize A to $R = (r_{ij})_{n \times m}$ using proper ways;

Step2, Calculate $R = (r_{ij})_{n \times m}$ to get $\dot{R} = (\dot{r}_{ij})_{n \times m}$, where

$$\dot{r}_{ij} = \frac{r_{ij}}{\sum_{i=1}^n r_{ij}} , \quad i \in N, j \in M .$$

Step3, Get the information entropy E_j of every QoS,

$$E_j = -\frac{1}{\ln n} \sum_{i=1}^n \dot{r}_{ij} \ln \dot{r}_{ij} , j \in M ,$$

Step4, Get weight vector $\omega = (\omega_1, \omega_2, \dots, \omega_m)$, where

$$\omega_j = \frac{1 - E_j}{\sum_{k=1}^m (1 - E_k)} ,$$

Step5, Synthesis QoS of every scheme is defined as

$$z_i(\omega) = \sum_{j=1}^m r_{ij} \omega_j, \quad i \in N,$$

Step6, Sorting and selecting a scheme are according to $z_i(\omega) (i \in N)$.

3 Conclusion and Future Directions

The studies on Web Service management focus on two aspects, one is the self-management of the element Web Service; the other is the management of composed services [5-7]. The QoS properties are independent on above two aspects due to different targets. In this paper, we aim to manage the QoS of composed Web Services and propose a new service description language based on current WSDL, and then propose a composition selection algorithm.

Short-term goals of our research contain three parts: firstly, we will extend the current UDDI to support EWSDL; secondly, we will add a monitor to the prototype E-WsFrame. The monitor can collect and analyze service data and make judgment when client or provider violates the agreement between them. Finally, we will order the candidate services by some algorithms to reduce composition price.

Acknowledgements

This research was sponsored by the National Science Foundation of China NSFC under the grant No. 90304006 and Research Fund for Doctoral Program of Higher Education of China (No.2002698018).

References

- [1] IBM dW 2004 special, <http://www-128.ibm.com/developerworks/cn/>.
- [2] YUE Kun, WANG Xiao-ling, ZHOU Ao-ying. Underlying Techniques for Web Services: A Survey. *Journal of software*, 2004, 15(3):428-442.
- [3] Boualem Benatallah, Marlon Dumas, Quan Z. Sheng, Anne H.H. Ngu. Declarative Composition and Peer-to-Peer Provisioning of Dynamic Web Services. *Proceedings of the 18th International Conference on Data Engineering (ICDE.02)*.
- [4] LIAO Yuan, TANG Lei, LI Ming-Shu. A Method of QoS-Aware Service Component Composition. *Chinese Journal of computers*, 2005, 28(4):627-p634.
- [5] Esfandiari, B., Tosic, V. Requirements for Web Service Composition Management, in *Proc. of the 11th HP-OVUA Workshop*, Paris, France, June 21-23, 2004.
- [6] Tosic, V., Patel, K., Pagurek, B. WSOL - Web Service Offerings Language, in *Proceedings of the Workshop on Web Services, e-Business, and the Semantic Web - WES Bussler, C. et al.(eds.)*, Toronto, Canada, May 2002.
- [7] Tosic, V., Ma, W, Pagurek, B, Esfandiari, B. Web Services Offerings Infrastructure (WSOI) – A Management Infrastructure for XML Web Services. In *Proc. of NOMS (IEEE/IFIP Network Operations and Management Symposium) 2004*, Seoul, South Korea, April 19-23, 2004, IEEE, 2004, pp. 817-830.

- [8] Chakraborty D, Joshi A, Yesha Y, Finin T. GSD: A novel group-based services discovery protocol for MANETS. In: Proc. of the 4th IEEE Conf. on Mobile and Wireless Communications Networks. 2002.
- [9] Finkelstein A, Dow ell J. A Comedy of Errors: the London Ambulance Service case Study. Proceedings of the 8th International Workshop on Software Specification and Design. 1996. 2-4.
- [10] YANG Fang-chun, LONG Xiang-ming. An Overview on Software Non-Functional properties Research. Journal of Beijing University of Posts and Telecommunications, 2004,27(3):1-11.
- [11] WSDL 2.0. <http://www.w3.org/TR/2005/WD-wsdl20-primer-20050510/>.
- [12] JUSTIN O’SULLIVAN. What’s in a Service? Towards Accurate Description of Non-Functional Service Properties. Distributed and Parallel Databases, 2002, 12:117 - 133
- [13] Shuping Ran. A Model for Web Services Discovery with QoS. ACM SIGecom Exchanges, 2003, 4(1):1-10.
- [14] McCall J.A.,Richards P.K.,Walters G.F. Factors in software quality. RADC:Technical report RADC-TR-77-363,1977.
- [15] M. Pinedof. Scheduling: Theory, Algorithms, and Systems. Prentice Hall, 2001.
- [16] CHEN Ting, Programming Analysis. Science Press. Peking. 1987.
- [17] WANG Xiao-ping, CAO Li-ming. genetic algorithm- theory, application and realization. Xian Jiaotong University press. Xian, 2002.
- [18] Hwang C L, Yoon K. Multiple Attribute Decision Making and Applications. New York: Springer-Verlag, 1981.