

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

E-WsFrame: A Framework Support QoS Driven Web Services Composition

Chen Yan-Ping and Li Zeng-Zhi

Department of Computer Science and Technology, Xi'an Jiaotong University, 710049, Xi'an China

Abstract: Using pre-exist Web services to Provide composed Web services according to the QoS requirements of clients is still an urgent problem to be solved. In order to solve this problem, firstly, we enhanced the current WSDL to describe the QoS properties of Web services and then gave a way to choose the proper pre-exist services based on their QoS properties. Then, an evolved Web Service frame is proposed by adding a new role to the current Web Service frame to realize the QoS driven composition. Finally, we design and implement a prototype, which can flexible select and compose Web services at run time.

Key words: Service management, Web service composition, QoS, WSDL

INTRODUCTION

Web Service is a hotspot in the research of SOA as the best realization of SOA. The primary goal of Web Service is to meet clients' requirements by integrating the pre-existing resources distributed in Internet, therefore, composing the existed service to provide an add-value complex service is a good way when the existing single service cannot meet clients' requirements (functional and/or non-functional). As Web services become more commercial, the service should not only satisfy the functional requirements of clients but also non-functional requirements. So, the main purpose of this study is to dynamically provide composed Web services according to clients' QoS requirements.

There are several problems in Web Service frame needed to be studied such as service composition, data integration of services, security of services, etc. However, all these existing problems are aroused by the composition (Kun *et al.*, 2004). 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 (Benatallah *et al.*, 2002) 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 (Yuan *et al.*, 2005), 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 (Esfandiari and Tomic, 2004; Tomic *et al.*, 2002, 2004) provide WSOL (Web Service Offering Language) that can change services at run time by dynamic switching in different service

constrains. DAML-S (Chakraborty *et al.*, 2002) 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.

WEB SERVICE DESCRIPTION MODEL

QoS property of Web services: There are many failed projects because of ignoring non-functional properties of software (Finkelstein and Dowell, 1996). Definitions of non-functional properties of software are various and have no unified definition (Fang-Chun and Xaing-Ming, 2004). 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.

If a service provider only takes functional requirement of clients into account without considering non-functional requirements when providing the service, the provided services will not be accepted by clients at runtime. For example, a loanApprove service, which can provide a function of approving the loan,

cannot guarantee the reliability and security then, obviously no client will use this service. 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. 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. A Web Service Offering Language is proposed (Esfandiari and Tosic, 2002; Tosic *et al.*, 2002, 2004) (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 current WSDL, 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 by Run (2003), it gives some definitions of QoS properties, but these properties are not from the view of managing composed Web services. In this study we explore an enhanced Web Service description language according to the need of managing the QoS properties of composed Web Service.

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 $S = \{\text{Function: QoS}\}$; where, Function 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 service and composed service). Web services that participate in the composition are all called element services. In this study, there are no essences differences between the element service and the composed service except for the granularity.

- Response time 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;
- Expire time represents the expire time of a service and the reliability of a service can be ensured before the expire time,
- 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,
- Security level represents the security level of a service.

For a composed service ,

$$CS = \text{Compose}(\text{Service}_i)_{i=1}^m$$

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, response time of CS is not the sum of response time of every element services when existing concurrent process in CS. Followings give a way to calculate the QoS parameters of CS.

- Response Time, considering that there exists concurrent service in composed Web Service, response time of composed service is not the sum of response time of all the element services, it should be the sum of critical route in execution process. CPA (Liangzhao *et al.*, 2003) is an algorithm to find the critical route.

Response time = CPA (Service₁, Service₂, ..., Service_m),

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

- Concurrency = min (concurrency₁, concurrency₂,..., concurrency_m),
- Expire time = min (expireTime₁, expireTime₂,..., expireTime_m),
- Price = $\prod_{i=1}^m \text{price}_i$,
- Fine = A*Price, the client and the provider negotiate to get A,
- SecurityLevel = min (securityLevel₁, securityLevel₂,..., 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.

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 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.

Definition 3 candidate relation: Let Service_a, Service_b are two services, if

- $\prod_{i=1}^l \text{funtion}(\text{Service}_a) \supseteq \prod_{i=1}^k \text{funtion}(\text{Service}_b)$
l, k ∈ N, l ≥ k or
- $\prod_{i=1}^k \text{funtion}(\text{Service}_b) \supseteq \prod_{i=1}^l \text{funtion}(\text{Service}_a)$
l, k ∈ N, k ≥ l

There exists a Candidate Relation between Service_a and Service_b, remembered as

$$\text{Service}_a \xleftrightarrow{\wedge} \text{Service}_b$$

Candidate Relation is an equivalence relationship.

For a Service Composer, if exist

$$\text{Service}_a \xleftrightarrow{\wedge} \text{Service}_b$$

the Composer should order the proper service according to QoS requirements to clients. We define a new duality relation ≥ meaning at least not worse than and deduced another two relations φ and ≈ based on ≥:

- Service_i^m ≈ Service_iⁿ iff Service_i^m ≥ Service_iⁿ and Service_iⁿ ≥ Service_i^m;
- Service_i^m φ Service_iⁿ iff Service_i^m ≥ Service_iⁿ and not Service_iⁿ ≥ Service_i^m.

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:

- Clients' SLA is the QoS of the composed Web services;
- Composed services and element services use the same glossary;
- The required QoS of clients can be pressed as a vector

$$(\text{ResponseTime}_c^*, \text{Availability}_c^*, \text{Concurrency}_c^*, \text{ExpireTime}_c^*, \text{Price}_c^*, \text{Fine}_c^*, \text{SecurityLevel}_c^*);$$

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, for example

$$(\text{ResponseTime}_c^* > \text{Availability}_c^* > \text{Avilability}_c^*, \text{Concurrency}_c^* < \text{Concurrency}_c^*, \text{ExpireTime}_c^* > \text{ExpireTime}_c^*, \text{Price}_c^* < \text{Price}_c^*, \text{Fine}_c^* < \text{Fine}_c^*, \text{SecurityLevel}_c^* > \text{SecurityLevel}_c^*);$$

- The composition logic is predefined and the aim is to simplify the composition and pay more attention to QoS.

Table 1: Programming matrix

Properties of composed web services					
Scheme	Response time (m sec)	Availability (%)	Concurncy	...	Security level
1	100	99	1	...	3
2	70	97	4	...	2
...					
M	80	97	3	...	2

Based on these assumptions, we use multi-object programming to model this problem and get the following programming matrix as shown in Table 1.

There are many ways to solve the multi-object programming (Ting, 1987; Hwang and Yoon, 1981; Yan-Ping *et al.*, 2006a-c).

DESIGN OF COMPOSITION WEB SERVICES FRAME

Evolved Web Service frame: In order to realize the service composition based on EWSDL, we proposed an evolved Web Service Frame (E-WsFrame) (Yan-Ping *et al.*, 2006a-c). Figure 1 gives the details of E-WsFrame.

There are three main roles in current Web Service framework: client, registry and service provider. In E-WsFrame, we proposed a new role named after Composer. Composer takes charge of composing pre-exist services to meet clients' requirements (functional and non-functional). In Fig. 1, dashed lines represent interactions

among roles in the concurrent Web Service framework and solid lines represent the augment interactions in E-WsFrame.

Noticed that, in E-WsFrame, we set Composer as a part of provider and some one may consider that Composer also can be looked as a part of registry. To simplify the problem, we consider Composer as a special provider. There is no need to extend the current registry because it cannot support EWSDL. How to evolve current registry to support EWSDL and how to lookup based on the EWSDL file are the next step of direction.

Service composition management language: Unlike other Web Service process description language, e.g., BPEL4WS, we defined a new language SCML (Service Composition Management Language). Using SCML, we can not only describe the process but also describe the QoS of services.

In SCML, the root element tService is composed of two sub-elements. One is service process which describe the process and embody the functional properties of services, the other is serviceConstrains which describe constrains and embody the non-functional properties of services. Commonly used Web Service process description languages e.g. BPEL4WS (Yan-Ping *et al.*, 2006a), they only contain part information of service process of SCML.

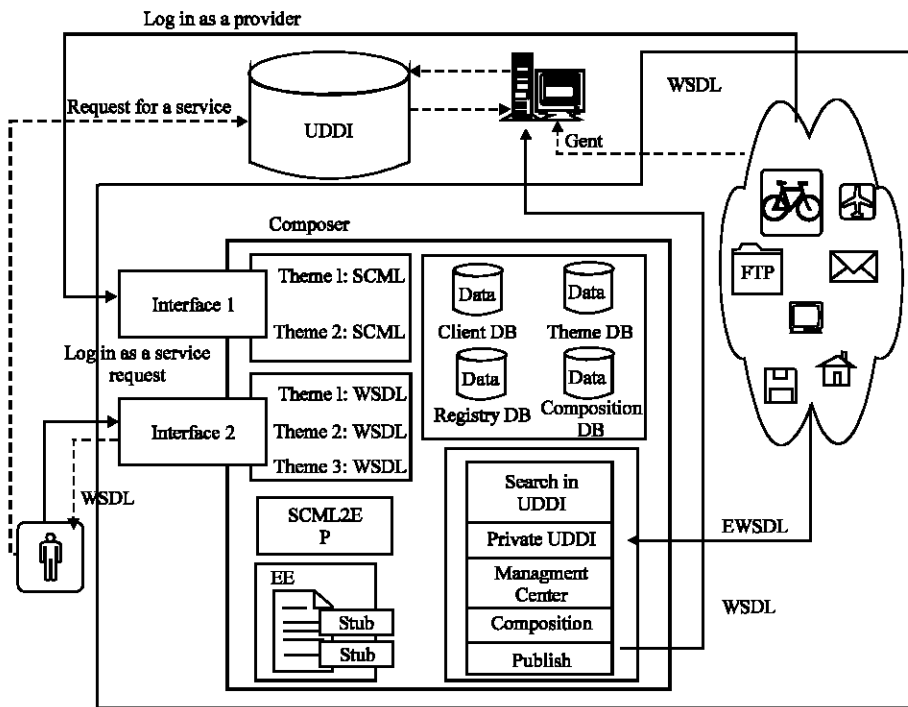


Fig. 1: Structure of E-WsFrame

IMPLEMENT PROTOTYPE OF E-WSFRAME

Main functional modules: There are five main functional modules:

- ManagementCenter, it provides the functions such as register, lookup and composition, etc.;
- Dbs, it contains theme DB, Client DB, Registry DB, composition DB. Theme DB stores the information of the themes in current system. Client DB stores three kinds of interface: one is for the client who requests for composed Web services, the second is for the element services providers and the third is for E-WsFrame management. Registry DB stores the information of the service providers. Composition DB stores the schemes of themes.
- Login interface, there are two kinds of interface one is for clients who requests for composed services, the other is for service providers.
- SCML2EP, Converts the SCML files into executable processes, there are following main steps:
 - Creates executable processes and hands it to EE,
 - Creates stubs,
 - Fills the portType table in the portType table of theme DB,
 - Fills the operation table in the operation table of theme DB,
 - Fills the sequence table in the sequence table of composition DB and
- EE, execution environment which can execute the process and realize the dynamic composition.

Experiments and results: In E-WsFrame, client interface is developed by JSP+Win2000+Tomcat. Management Center is developed by Borland Delphi. DBs is developed by MySQL and an Active-BPEL engine is disposed on execution server.

We choose the travel problem as the scenario to realize the composed service. A travelService is composed by several element services such as accommodation Service, sightseeingService, weather Service and pay Service, etc. For two travelers even if they choose the same sightseeing, ManagementCenter can provide clients the different travelService according to their different QoS requirements such as price, accommodation requirement, etc. and fills the portType IDs into the EWSDL of element services in EE.

In Fig. 2a each step in a composed service process has two element services of the Candidate Relation. While in Fig. 2b each step in a composed service process has

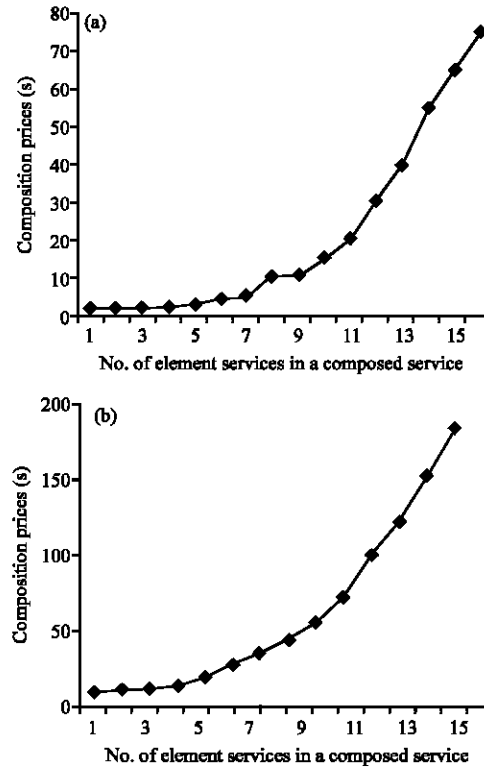


Fig. 2: Composition price of composer

five element services of the Candidate Relation. Examinations indicate that composition price will increase greatly with both the increasing numbers of candidate services in the composed Web Service and the number of element services in a composed service process. In order to deduce the cost, we should reduce the number of candidate services in composed service through ordering the candidate service by some algorithms and reduce the number of element services by increasing the granularity of services.

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 (Esfandiari and Tomic, 2002; Tomic *et al.*, 2002, 2004). The QoS properties are independent on above two aspects due to different targets. In this study we aim to manage the QoS of composed Web services and propose a new service description language based on current WSDL and then evolved the current Web Service architecture with a new role of Composer, finally give a prototype and do some examinations. Our experiments evaluate the correctness of the researches.

Short-term goal of our research is to add a monitor to E-WsFrame. The monitor can collect and analyze service data and make judgment when client or provider violates the agreement between them.

ACKNOWLEDGMENTS

This research was sponsored by the National Science Foundation of China NSFC under the grant No. 90304006 and 60673170.

REFERENCES

- Benatallah, B., M. Dumas, Q.Z. Sheng and A.H.H. Ngu, 2002. Declarative composition and peer-to-peer provisioning of dynamic Web services. Proceedings of the 18th International Conference on Data Engineering (ICDE.02).
- Chakraborty, D., A. Joshi, Y. Yesha and T. Finin, 2002. GSD: A novel group-based services discovery protocol for MANETS. In: Proc. of the 4th IEEE Conf. on Mobile and Wireless Communications Networks.
- Esfandiari, B. and V. Tasic, 2004. Requirements for Web Service composition management. In Proc. of the 11th HP-OVUA Workshop, Paris, France, June 21-23.
- Fang-Chun, Y. and L. Xiang-Ming, 2004. An overview on software non-functional properties research. *J. Beijing Univ. Posts and Telecommun.*, 27: 1-11.
- Finkelstein, A. and J. Dowell, 1996. A comedy of errors: The london ambulance service case study. Proceedings of the 8th International Workshop on Software Specification and Design, pp: 2-4.
- Hwang, C.L. and K. Yoon, 1981. Multiple Attribute Decision Making and Applications. New York: Springer-Verlag.
- Kun, Y., W. Xiao-Ling and Z. Ao-Ying, 2004. Underlying techniques for Web services: A survey. *J. Software*, 15: 428-442.
- Liangzhao, Z., B. Boualem and D. Marlon, 2003. Quality driven Web services composition. Proceeding of WWW May 20-24, Hungary.
- Ran, S., 2003. A model for Web services discovery with QoS. *ACM SIGecom Exchanges*, 4: 1-10.
- Ting, C., 1987. Programming Analysis. Science Press. Peking.
- Tasic, V., K. Patel and B. Pagurek, 2002. 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.
- Tasic, V., W. Ma, B. Pagurek and B. Esfandiari, 2004. 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, IEEE, 2004, pp: 817-830.
- Yan-Ping, C., Li Zeng-Zhi, J. Qin-Xue and W. Chuang, 2006a. Study on QoS Driven Web services Composition. The Eighth Asia Pacific Web Conference (APWEB'06), LNCS Volume 3841/2005, January 16th-18th, Harbin, China.
- Yan-Ping, C., Li Zeng-Zhi and G. Zhi-Sheng, 2006b. Service selection algorithm based on quality of service and its implementation for Web services composition. *J. Xi'an Jiaotong Univ.*, 40: 897-900.
- Yan-Ping, C., Li Zeng-Zhi and T. Ya-Zhe, 2006c. A method satisfying markov process of Web Service composition under incomplete constrains. *Chinese J. Comput.*, 2: 1076-1083.
- Yan-Ping, C., Li Zeng-Zhi and X. Hong, 2006d. Design and implement a web services composition framework based on the quality of service. *J. Beijing Univ. Posts and Telecommun.*, 29: 21-24.
- Yuan, L., T. Lei and Li Ming-Shu, 2005. A method of qos-aware service component composition. *Chinese J. Comput.*, 28: 627-634.