http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

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

Information Technology Journal 10 (8): 1645-1649, 2011
ISSN 1812-5638 / DOI: 10.3923/itj.2011.1645.1649

Research on Dynamic Web Services Composition Framework Based on Quality of Service

Shujun Pei and Deyun Chen
College of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China

Abstract: By using Business Process Execution Language (BPEL), a single Web service can be combined into a complex business process, but the BPEL language doesn't support modification operation in the course of business implementation process. Although the BPEL language has error handling and compensation mechanism, it will still affect the efficiency of the business if the business process fails to proceed. To solve this problem, a BPEL and Quality of Service (QoS) based dynamic Web service composition framework is put forward, taking advantages of the dynamic binding feature of QoS components and the function of dynamic discovery of Web service of Universal Description Discovery and Integration (UDDI), achieving the function of dynamically select Web service during business process running. Therefore, the efficiency of business process execution is improved greatly.

Key words: Web service, dynamic composition framework, UDDI, QoS

INTRODUCTION

With the growing maturity of Web service technology, more and more stable-to-use Web services are sharing on the network. A single Web service only provides limited functions, in order to make full use of the sharable Web services, it is necessary to combine them to offer stronger service function. BPEL (business process execution language) is a language used to describe business process implementation and a powerful tool of Web service composition. BPEL regulate how to describe the structure of a process, how to use a Web service, how to define and deliver the data in process and how to manage the life-circle. However, people gradually realized that BPEL can not solve all the problems the service composition faced under open Internet environment in practical use. Although during the design process of composite service, we can seek appropriate member to service via effective search mechanism, the BPEL composition service maintained a binding relationship with each service member, leading to a situation that difficult to dynamically adjust its structure during operation time. When the environment changes, the combination can only be re-established and re-described (Piccinelli *et al*., 2008). In this study, a dynamic Web service composition architecture is proposed. This architecture is based on BPEL language and UDDI protocol; taking advantages of extended QoS component to support business process dynamically bind the Web service during running time. And it can dynamically modify services under exceptional circumstances, so as to better improve execution efficiency of business process.

RELATED RESEARCH ON SERVICE COMPOSITION FRAMEWORK

On the aspect of dynamic Web service composition framework, many researchers domestic or foreign have put forward a number of frameworks. In MAIS framework, dynamic combination process is divided into two parts: The first is to replace what the abstract service faces by entity service; the second one, if the entity service there isn't available during execution process, it is to search for other abstract service integrity to replace and to find a corresponding entity service or even evaluate and replace it for a second time (Zhao and Lu, 2009). The disadvantages of MAIS framework lies in it blends the error handling into the framework, making the structure of the framework more complex to achieve (Shao *et al*., 2008). UNICS framework is a performance framework that combined capacity and state, part of it is service structure that cares service and service components, only concerning the function and call; another part is computing structure that cares the software components used in modeling the service, only concerning function and implementation. This method uses non-centered

Corresponding Author: Shujun Pei, College of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China

thought to arrange. Its fault is it is difficult to manage the framework and a non-centered arrangement is worse than a centered one in flexibility and robustness (Modafferi *et al*., 2009). USON framework is a service composition and service generation framework in a common computing environment. It calls all calculation entities Service Element (SE), such as Web service, computing resources, etc and calls combination methods among SE Service Template (ST). During service composition period, SE combines on the basis of every ST; on service generation period, new ST is generated through the combination history of SE and ST. The disadvantage of such a framework lies in that the service combination heavily depends on existed services template, human intervention is needed by template's generation and management. It is not obvious to generate a new template (Qian *et al*., 2006). In service model of FRESCO framework, each service contains two parts: capacity and scheduled logic. Capacity can be divided into five categories: interactive capabilities, distribution capacity, negotiation capability, contract management and billing capacities. Its composition model includes: service composition, service coordination and service aggregation three blocks (Takemoto *et al*., 2004). The actual composition process depends on composition, coordination and aggregation among the services. FRESCO framework is limited in: it is still a template to combine, coordinate and aggregate in framework and the number of service that this template involves is less than usual, so it can't meet the service composition with a large number of services (Yan-Ping and Zeng-Zhi, 2007).

NEW FRAMEWORK FOR DYNAMIC WEB SERVICE COMPOSITION

New framework: The core idea of this framework is to use the unified discovery services function of UDDI and the dynamic binding function of QoS component to offer desirable Web services for running business process. It can dynamically bind the Web service during running business process and dynamically modify service function under exceptional circumstances. To adopt error processing mechanism of BPEL language to handle it, keeping a simplified framework and a easy-managed property. The core of the framework is in QoS component (Yang and Li, 2010); it monitors and manages other parts of the framework. The framework does not rely on a fixed template; it only aims at dynamically binding the Web service BPEL process in BPEL process store and the dynamic replacement in it. Since there is no specific template restriction, the framework has no limitation to the composition number of service, either. The frame structure

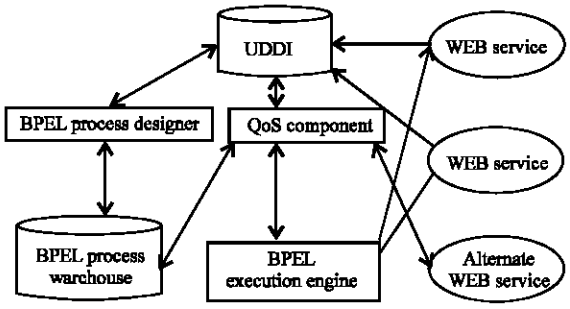

Fig. 1: Frame structure

is shown in Fig. 1. Function of each part of the framework is shown as follows:

BPEL process designer: it provides visual process modeling tool and automatically generate BPEL process documentation. When the process needs to be modified to import the original BPEL document generation flow chart of the process and make the appropriate changes.

BPEL process warehouse: Primarily, it stores the design-completed business process instance. The designer can search for process warehouse when designing a new process; a desirable process document can be used directly if it is desirable. If there is no suitable process document, the BPEL designer can be used to design and to store the qualified new-designed process document in to the BPEL process warehouse.

QoS component: Core of framework, its functions mainly are:

- Analyze BPEL business process template and ask UDDI for appropriate Web service
- Dynamically bind Web service according to selection
- Send BPEL business process to engine to execute
- Monitor BPEL execution engine and alternate Web service, acquiring QoS information, QoS managing the framework

BPEL execution engine: When receiving requests, it chooses the process document from QoS component according to description of the process to complete specific tasks by using corresponding Web service.

UDDI: Its functions are:

- Users publish Web service for the use of other users
- Users can inquire demanded service and acquire corresponding service information

Web service: End of business implementation, to realize specific business

Alternate web service: QoS component can find the Web service which is not dynamically tied together in UDDI.

Realization of dynamic composition: In designing period of business process, business process designers design via BPEL process designer according to the requirement of the current business process, can inquire the Web service information through UDDI. They can submit the completed business process to BPEL process warehouse and to store it. The process of business implementation is shown in Fig. 2.

Process of business implementation:

Step 1: Request select BPEL process and submit it to QoS component according to requirement
Step 2: QoS components analyze corresponding BPEL process template, and inquire UDDI for the Web service that BPEL process template demanded according to the analysis
Step 3: QoS components dynamically bind the Web service in BPEL process according to its choice
Step 4: QoS components add other qualified Web service into the list of alternative services
Step 5: QoS component send the business process to BPEL implementation to carry it out
Step 6: QoS components monitor BPEL implementation engine and alternative Web services

In the running period of business process, because of a Web service's abnormality, or too much time for business process for the reason of the bad network, the QoS component will use the replaceable Web service in alternative list of Web service to dynamically replace the abnormal one, making sure the business process can continue normally (Talib *et al*., 2004). Its operation process is as follows:

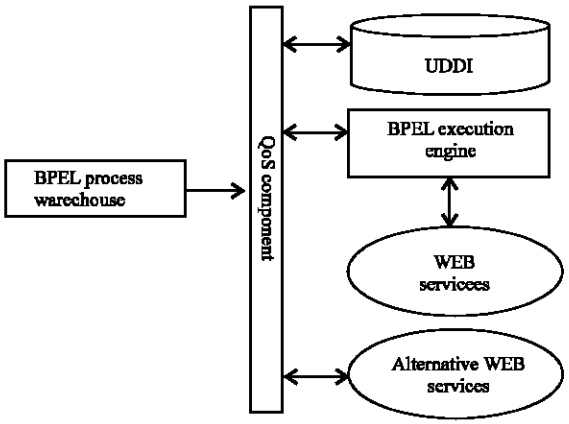

Fig. 2: Business process implementation

Step 1: If there is an abnormal Web service in business process, QoS component will inform BPEL execution engine to pause the execution of business process
Step 2: QoS component can choose a available Web service and send it to BPEL execution engine from alternative Web service according to replacement policy. BPEL execution engine calls new Web service and BPEL process continue to execute

Combination of QoS and dynamic service composition
Extension of BPEL element: We can find from the research to the combination of BPEL services: the reason why the traditional BPEL service composition is static binding lies in that the partnership (partner link) to connect the service member is statically bound, that is to say the process design have identified specific interactions among services (Supadulchai and Aagesen, 2005). Therefore, to realize dynamic binding, it is necessary to redefine the 'partner linker'. A effective method is not to make the meaning of partner link a specific value but to take it as a function: Partnerlink = f (n1, n2....) Function parameter can use the service name, method name, message type, etc in Web service. In the BPEL design process, a parameter can be assigned initial value or to say be identified a specific services. In the period of BPEL process execution, the result inquired from UDDI according to QoS components selects the Web service that meet the requirement of function Partnerlink = f (n1, n2....) for dynamic binding. If the service appears in the implementation process is not available, then according to the QoS algorithm in QoS components to select specific Web service to replace the failed service from the query results.

WSDL, SOAP and UDDI data structure extension: Since the QoS description capacity of WSDL document to Web service is limited, and without a specific definition and description to the QoS metrics of Web service, the service providers publish and register service to UDDI by SOAP information. But the SOAP information does not support the QoS description mechanism, and the UDDI service discovery mechanism does not find the function of Web service according to QoS property, so it is necessary to expand the description of WSDL file, header

of SOAP information, and data structure of UDDI. In description file of WSDL, we can add QoS child elements in < service > component to enable it with the description function to QoS property of Web service. SOAP message contains two child elements: an optional header and compulsory message body. To expand SOAP header, to add service quality parameters, such as price, execution time, reliability, availability, credibility, etc. Similarly, the registration information of Web service in UDDI needs added corresponding QoS property of Web service (Mansourian *et al*., 2008).

QoS selection method: For QoS attributes for Web services, it takes cost, time, reliability and availability as examples to illustrate a method for selecting web service. for a single Web service, its overall service quality can be defined as:

$$Q(S) = (q_{price}(s): q_{time}(s): q_{rel}(s): q_{avail}(s))$$

For a complex business process with multiple independent Web service composition it needs to find optimal Web service on the nodes of business process, and to use the quality parameter of multi-dimensional vector space to represent the service quality of the whole process (Hanna and Abu Ali, 2011). The definition is as follows:

In simple cases, when calculate service quality of business process, we can use the weighted average method to measure; in Specific application, to set corresponding algorithm according to the focused QoS attributes.

$$Q=\begin{Bmatrix} q_{price}(s_1) & q_{time}(s_1) & q_{rel}(s_1) & q_{avail}(s_1) \\ q_{price}(s_2) & q_{time}(s_2) & q_{rel}(s_2) & q_{avail}(s_2) \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ q_{price}(s_n) & q_{time}(s_n) & q_{rel}(s_n) & q_{avail}(s_n) \end{Bmatrix} =$$

$$\begin{Bmatrix} q_{1.1} & q_{1.2} & q_{1.3} & q_{1.4} \\ q_{2.1} & q_{2.2} & q_{2.3} & q_{2.4} \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ q_{n.1} & q_{n.2} & q_{n.3} & q_{n.4} \end{Bmatrix}$$

For the design of QoS management function of QoS composition, the traditional design method of QoS management framework can be adopted. It can make corresponding QoS model according to response time, reliability, throughout and other specific focuses. The QoS information collection can be based on the hander mechanism that provided by the JAX-RPC and JAX-WS specification (Yanping *et al*., 2006). In the QoS information storage, since there is few deleting and modifying operation, it can store in the way of adding up the ordinary files to index. Generally speaking, the BPEL process execution efficiency is the most important; therefore, QoS algorithm can be designed on the basis of response time and throughput. QoS management function can be used as a separate module to design; its changes will not affect the dynamic composition of Web service and the execution of BPEL process, making the QoS components be easily extended.

CONCLUSION

Dynamic Web service composition is an important tendency in the field of Web service. Combines with BPEL and UDDI, it put forward a new framework of Web service composition which is based on QoS components. This framework has the ability of dynamically bind and choose Web service during the period of business process, having good scalability and recoverability and enhanced the business composition ability of Web service. To enhance the usefulness of the framework, we will further study the security of business processes, service granularity division in business process templates and other issues. In addition, research on dynamic binding strategy and replacement strategy of Web service are the major focus in future work.

ACKNOWLEDGMENTS

This study is supported by high and new technology industry foundation of Harbin Grant by No. 2008FG02CG201.

REFERENCES

Hanna, S. and A. Abu Ali, 2011. Platform effect on web services robustness testing. J. Applied Sci., 11: 360-366.

Mansourian, A., F. Mahdi and T. Mohammad, 2008. Development of new generations of mobile GIS systems using web services technologies: A case study for emergency management. J. Applied Sci., 8: 2669-2677.

Modafferi, S., E. Mussi, A. Maurino and B. Pernici, 2009. A framework for provisioning of complex e-services. Proceedings of the International Conference on Services Computing, Sept. 15-18, Shanghai, China, pp: 81-90.

Piccinelli, G., C. Zirpins and W. Lamersdorf, 2008. The FRESCO framework: An overview. Proceedings of the Symposium on Applications and the Internet Workshops, Jan. 27-31, Florida, USA., pp: 120-124.

Qian, M., J. Yu and X. Ma, 2006. A BPEL support environment based on run-time architecture. Electronics, 34: 2361-2365.

Shao, L., L. Tian and J. Zhao, 2008. An extended QoS managing framework of web service. Computers, 31: 1458-1470.

Supadulchai, P. and F.A. Aagesen, 2005. A framework for dynamic service composition. Proceedings of the 6th International Symposium on a World of Wireless Mobile and Multimedia Networks, June 13-16, Taormina, Italy, pp: 527-531.

Takemoto, M., T. Oh-Ishi, T. Iwata, Y. Yamato and Y. Tanaka *et al*., 2004. A service-composition and service-emergence framework for ubiquitous-computing environments. Proceedings of the International Symposium on Applications and the Internet Workshops, Jan. 26-30, Tokyo, Japan, pp:313-313.

Talib, A.M., Y. Zongkai and Q.M. Ilyas, 2004. Modeling the flow in dynamic web services composition. Inform. Technol. J., 3: 184-187.

Yan-Ping, C. and L. Zeng-Zhi, 2007. E-WsFrame: A framework support QoS driven web services composition. Inform. Technol. J., 6: 390-395.

Yang, H. and Z. Li, 2010. Improving QoS of web service composition by dynamic configuration. Inform. Technol. J., 9: 422-429.

Yanping, C., L. Zengzhi, G. Zhisheng, J. Qinxue and W. Chuang, 2006. Service selection algorithm based on quality of service and its implementation for Web services composition. J. Xian Jiaotong Univ., 40: 897-900.

Zhao, J. and H. Lu, 2009. A composition method that supports field characteristic. Comput. Sci., 28: 731-738.