2010 IEEE International Conference on Web Services

Compose Real Web Services with Context

Sen Luo, Bin Xu, Kewu Sun
Department of Computer Science and Technology, Tsinghua University
Tsinghua National Laboratory for Information Science and Technology
Beijing, 100084, China
{luos, xubin, kewusun}@keg.cs.tsinghua.edu.cn

***Abstract*—Web service composition is to integrate existing web services to provide a compound service which satisfies specified requirement. However, traditional web service compositions fail to provide different compound services under various scenarios. In this paper, we propose an approach to compose services with context. A context ontology is defined to describe the scenario for user. An abstract service description is defined to describe current three kinds of services including WSDL/Restful/Web API. The goal is to correlate context and service composition to improve the quality of the compound service from real services.**

I. INTRODUCTION

One of the thrilling uses of web service lies in service composition, which allows developers to develop rapidly new applications by composing existing services. Traditional web service compositions usually focus on one or several metrics [1-3], such as response time of compound service.

Assume a compound service which consists of three individual services. The first takes address as input and outputs latitude and longitude of the address. The second uses the outputs of the first service as inputs, and tells restaurant information around former position. And then the third service shows restaurant information on a map. A number of approaches have been proposed for web service composition, they usually focus on finding the optimal solution under some quality metric, such as finding the least response time solution [4]. Our goal is to combine service composition with the scenario in which the compound service is going to be used. In other words, we aim at getting different solutions for different scenarios and the solution for a scenario should be optimal. Consider the composition example above, solutions can be given according to the scenarios different in whether user likes spicy food or not. Also, by checking whether Google map service is accessible, different solutions for scenarios will be suggested.

Our work aims at taking an overall view of the scenario to guide composition so that we can get the best satisfying compound service. This is more flexible and reasonable compared with traditional service compositions. With different environments, the compound services can be different too. And to make our work interesting and convincing, the services used for composition is not constrained to WSDL/SOAP services. RESTful (Representational State Transfer) services and some web API (such as Google map API) are also included.

Our work includes building context ontology which is used to describe the scenario formally, establishing service description that can accommodate WSDL, RESTful and Web API and correlating context and service composition. Section 2 presents the context ontology. And Section 3 introduces service description. Section 4 tells how to correlate context and service composition. Section 5 discusses related works. And finally, Section 6 gives conclusion of our work.

II. CONTEXT ONTOLOGY

Context should contain everything that can be used to describe the scenario, which requires context ontology could be extent easily. And used to optimize service composition, context ontology is required to be defined very formally. In this regard, we choose OWL (Web Ontology Language) to model context ontology for its expressiveness and extensibility.

Our context ontology is responsible to capture all the characteristics of the scenario in which compound service is used, which is indeed a difficult task. As our domain can be divided into a collection of sub-domains such as desktop sub-domain, laptop sub-domain, mobile phone sub-domain, etc, we separate domains to reduce the burden of context capturing. Our context ontology is divided into upper ontology and domain specific ontology.

Upper ontology is a high level ontology which captures general knowledge and is divided into four categories, user context, computation context, physics context and time context. User context describes requirement or preference of the user. Computation context tells information about the devices. Physics context describes physical environment about the scenario. Time context tells the time.

Domain specific ontology means low level ontology which defines details of concepts. And we have developed desktop ontology, laptop ontology and mobile phone ontology.

After binding upper ontology and domain specific ontology, context ontology is formed. It could describe scenario that a person sitting at table in a 23 centigrade room. It also could describe scenario that a child walking outside in a windy condition.

III. SERVICE DESCRIPTION

WSDL/SOAP services, RESTful services and web API are quite different from each other. WSDL/SOAP services

978-0-7695-4128-0/10 $26.00 © 2010 IEEE
DOI 10.1109/ICWS.2010.63

IEEE computer society

are well defined by international standards. RESTful services are proposed according to the resource share view of the Internet. And there are no international standards for web API and sometimes they are very complicated.

We propose an object based description that every service belongs to an object. The service of an object is defined similar to WSDL with a few extends. And in order to build an application, we create some local services, which can be described by the service description.

IV. Correlate Context and Service Composition

Our goal is to correlate context and service composition so as to improve the quality of the compound service. To evaluate service quality, non-functional criteria of services are needs, like what traditional web service compositions do. Traditional evaluation is usually based on several quality metrics, such as the response time of services and the cost of services. The quality metrics in our system are added a new dimension describing the contexts that have connection with these metrics and are more diverse. For the example in introduction, traditional service compositions may have response time as quality metric. In our system, response time is connected with a context (user's requirement about response time), and the response time and the connection are together serving as quality metric.

We propose a constrain-based partial weight approach to let context exert an impact on service composition. It is constrain-based as the compound service must obey constrains defined by context, such as response time limitation or cost limitation. We call it partial because for a specified context only parts of services will response to it. This is different from traditional compositions, in which all services have the same quality metrics. And we use a weight based method to integrate all contexts to form a uniform evaluation system. Furthermore, as business process may have various flow structures, such as sequence flow, parallel flow, switch flow, etc. The calculation of weight gain should follow these flows reasonably. In this manner, we combine context and service composition to form an optimization problem. Upon a given scenario, to find the compound service with highest weight gain while satisfying constrain defined in the context.

Furthermore, we can monitor the scenario. And according to the context change captured, make the compound service adjust. In this way, we make service composition dynamic and really blended with context.

V. Related Work

We can divide service compositions into two categories. The first category defines the business process of the compound service. And we call a unit of the process a task. A task can be done by different services which may have various quality metrics. The composition is to select services for tasks to get the best satisfying compound service. The other is that the business process is not clear [5, 6]. This kind of composition relies heavily on help of semantic interoperability. Our problem falls in the first category.

For every task, there is a list of services with different quality metrics for choosing. We need to get a combination of services which satisfy specified constrains and has the largest weight gain.

Our work is different from traditional compositions in that unlike traditional ones, which aim to find parameter based optimal solution with respect to limited quality metrics [1-3]; our approach takes an overall view. We take into account all the information in order to find the best solution for the given scenario. And traditional compositions can be well fit in our approach. For example, to find the least response time solution can be fit in the scenario that user cares about response time most.

VI. Conclusion

Service composition is meaningful as it can produce service with higher usage using existing services. We devote our energy to correlate context and service composition. In this way we can make service composition more intelligent. The compound service can be different in various scenarios.

Until now we have completed the definition of context ontology and the context collecting system. And draft service description and service quality metrics with aspect to context.

In the future, we will first give a formal definition of this problem in the base of above work. Then we will analyses the complexity of this problem. It should be a NP-hard problem. And in the end we will propose an algorithm for finding almost optimal solution.

Furthermore, we can implement a monitor system to watch context. So that the compound service is not only context specified, but also can change according to context. This will make service composition even more intelligent

Acknowledgment

This work is supported by China National High-Tech Project (863) under grant No. 2009AA01Z120 and 2007AA010306.

References

[1] G. Canfora, M. D. Penta, R. Esposito, and M. L. Villani. An approach for qos-aware service composition based on genetic algorithms. In GECCO, pages 1069–1075, 2005.

[2] Yu T, Zhang Y, and Lin KJ. Efficient Algorithms for Web Services Selection with End-to-End QoS Constraints. In ACM Transactions on the Web, Vol. 1, No. 1, Article 6, May 2007.

[3] Freddy L´ecu´e, Nikolay Mehandjiev. Towards Scalability of Quality Driven SemanticWeb Service Composition. In ICWS, pages 469-476, 2009.

[4] Web Service Challenge. http://www.ws-challenge.org.

[5] Q. A. Liang and S. Y. W. Su. AND/OR graph and search algorithm for discovering composite web service. In International Journal of Web Service Research, vol. 2, no.4, pages 48–67, 2005.

[6] S.-C. Oh, D. Lee, and S. R. T. Kumara. Web service planner (WSPR): An effecttive and scalable web service composition algorithm. In International Journal of Web Service Research, vol. 4, no. 1, pages 1–23, 2007.