Chapter 26
Study of Composing Web Service Based on SOA

Rongwang Yin

Abstract Automated service composition refers to automating the entire process of composing a workflow. This involves automating the discovery and selection of the service, ensuring semantic and data-type compatibility. We gave some basic notions and recent research of Web Service, and then classified Web Service Composition approaches according to the methodology they used. Furthermore, we analyzed every approach's motivation and its shortages, as well as outlined the essential problems of Web Service Composition. Finally, we concluded and discussed the developing trends.

Keywords SOA · Composition approaches · Web service

26.1 Introduction

In the application area, in order to be able to deal with the application environment that is complicated and changeable, most enterprises would build the IT system based on service-oriented architecture (SOA). They would adjust it and put it into application at a rapid pace according to the changes in the environment so as to adapt to the requirements of the business. From the perspective of the software techniques, the premise of service-oriented computing (SOC) is to "software as service". In addition, the core of SOC is to how to work in coordination with these software services in order to finish the facing application tasks. However, the traditional stacks of Web service do not equip with systems that offer common use

R. Yin (✉)
Foundation Teaching and Experiment Center, Hefei University, Hefei, 230601 Anhui, China
e-mail: yinrongwang@cssci.info

Y. Yang and M. Ma (eds.), *Proceedings of the 2nd International Conference on Green Communications and Networks 2012 (GCN 2012): Volume 2*,
Lecture Notes in Electrical Engineering 224, DOI: 10.1007/978-3-642-35567-7_26,
© Springer-Verlag Berlin Heidelberg 2013

to the services of discovery, composition, and execution. Therefore, they cannot support the dynamic interaction and composition between Web Services. With the appearance of Web Services, both of the academic and industrial fields, have made profound research of how to make use of Web Techniques, reuse existing, and rich Web Service resources by means of service composition, and produce new service resources that can meet the complicated applications at a rapid pace. They have brought out a series of methods for Web Service Composition. The definitions of Web Services and Web Services Composition have been given in Sect. 26.2 of this paper, basing on the current achievement of Web Service researches. In Sect. 26.3, a methodology used in Web Service Composition has been proposed to classify the Web Service Composition and their limitations have been pointed out. In Sect. 26.4, the research content and key problems on Web Service Composition have been brought out. Finally, the conclusion summarizes the whole paper and future jobs are being looked forward to.

26.2 Basic Concepts

Multilayer Structure of Web Service usually, a Web Service can be divided into five logic layers: Data Layer, Data Access Layer, Business Layer, Business Façade, and Listener. The nearest Web Service of Client tier is Listener while the furthest is Data Layer. Business Layer can be further divided into two sub-layers: Business logic and Business façade. Any physical data that is needed by Web Service is able to be saved in the Data Layer. The Data Access Layer is above Data Layer, which can offer data services to Business Layer. Data Access Layer can separate the business logic from the data storage in the bottom, which can protect the integrity of data. The Business Layer offers a simple links that can directly reflect the process of offering Web Service.

This system structure is very similar to the system structure of n layers application that is defined by Windows DNA. Listener in Web Service acts as the expression tier in the application of Windows DNA. If the service returns a response, Listener is responsible for making the response that is from the Business Layer into a message and then sends it back to the client tier.

Web Service Protocol The Web Service techniques can be divided into three key components: Description Stack, Discovery Stack, and Wire Stack. Description Stack deals with all kinds of techniques that describe Web Service in order to promote the common use of business process model and workflow structure in the relationship of B2B. Discovery Stack deals with techniques used for services of catalogues, discovery, and examination. Wire Stack is composed of techniques that offer information flow to operation engine of Web Service. Figure 26.1 shows the stack of Web Service.

Web Service Composition Web Service composition is the ability to offer value-added services through the composition of basic Web Services. Services composition is not only an important way for reusing service resources, but also a

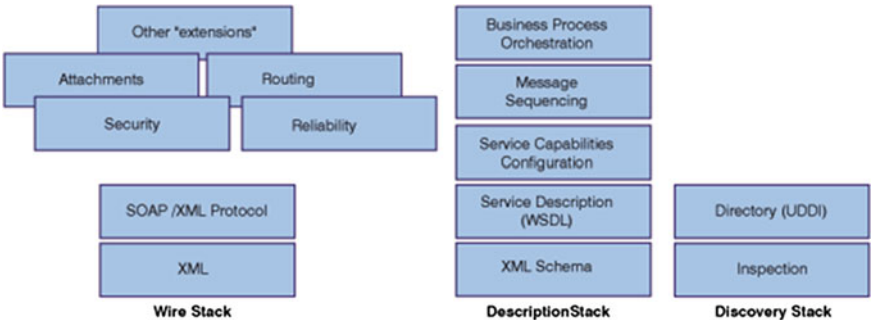

Fig. 26.1 Web service protocol

new channel to build a complicated application system and software based on Web. In the environment of distribution, different service components may offer the same function. If one of the components in certain service has changed in its functions, that service is required to do dynamic transfer in service components when asked, in order to make sure that the service is not being influenced. The dynamic transfer goes to another service component with the same functions.

26.3 Current Research Condition

Artificial Composition Methods What Document [1] brings up is typical artificial composition framework. In these frameworks, users need to produce a workflow by means of photos or text editor. In addition, they need to hand it to the engines of workflow process. Triana [1, 2] provides a user interface which is graphical. Users select the services they need from the tool box and put them into the layout manager. These services can be obtained from UDDI by means of key words. Moreover, in Triana, compositions that conduct services using local tools can be found.

BPWS4J [3] provides a plug-in unit to enable users to organize a map of workflow on the level of XML. This map of workflow and documents that relate to compositions and that serve the WSDL are handed to the execution engine. Self-Serve [4] allows users to build a workflow by means of service constructions. The service construction and UDDI can interact and find service in need. Use of an execution model based on P2P. Combination map (a state map that is labeled) is to be executed. One of the biggest features of this system is to introduce the concept of service container. The container has service sets with the same functions. In the process of construction, the service container will choose service that is actually in need according to the models of the members and an evaluation service.

There are certain defects for all of the above systems: In the first place, when the amount of the service offered increases, the discovery and choice of service methods are not able to be known timely. Therefore, it does not have scalability.

In the second place, they require the users to have knowledge about the basis. For example, in BPWS4J, users need to build a workflow on the level of XML. In the third place, if there are problems of certain services, the execution of the whole workflow will face failure. Even if in Self-Serve, service container may choose a service replacer.

Composition Methods of Semi-Automation For composition techniques using semi-automation like Documents [5], they make another pace in the process of composition and consider some characteristics of semantics in the choice of services. However, users still need to select services they are in need from appropriate service lists, and link these services according to the agreed subsequence. Sirin [6] brings up a system that is able to make consistent semantics in every stage that services have chosen. Cardoso and Sheth [5] have brought up a framework that is able to recommend services that meet the requirements of users. It mainly matches service template (ST) and service object (SO) that are appointed by users. While Chen [7] brings up a framework based on knowledge, which is able to offer suggestions when users are building the workflow. The system allows users to store workflows, which is beneficial to the reuse of workflow. The systems solve part of the problems of artificial composition, but they still do not have scalability when great amounts of services are accessed to users.

Automatic composition techniques of workflow such as [8] automate the whole composition process by means of artificial intelligence and relevant techniques. McIlraith and Son [9] is a service composition framework based on Agent. It guides compositions by using services of generic procedures and semantics. Agent acts as the gateway for Web services and is responsible for the choice of service. The framework assumes the existence of a generic procedure. If the generic procedure disappears, the composition will fail to work. In addition, if Agent is not able to match a service, execution will stop. SWORD [10] is to describe automatic compositions by using services based on rules. Users appoint initial and final state and then planners try to make a series of services a link that can meet the requirements. Here it requires that users appoint states. Moreover, there is no automatic service discovery mechanism in it. All the same, compositions also execute according to certain services. When certain services fail to work, it is very hard to ensure the successful execution of the whole process.

An important aspect in the composition workflow is the discovery of services. The researches on the aspect mainly focus on using DAML-S to describe services. Matchers compare the descriptions of DAML-S of the service requester and the service provider. Sycara [11] puts forward a framework based on DAML-S matcher and DAML-S virtual machine. The framework does not use workflow warehouse techniques. Therefore, when receiving a request, the workflow has to recalculate it every time. In addition, it does not distinguish the workflow in execution and out of execution. All kinds of workflows build on the available basis of service being executed. The result is that the workflow it built cannot be reused or shared, as nobody can ensure the service that is available now will still be available in the future. Sheshagiri [12] puts forward anther framework. It has two main characteristics: one is to describe the functions of services using DAML-S;

the other is to produce workflows using backward Chain Algorithm. One common defect of these frameworks is that it lacks Fault-tolerance Mechanism. In addition, there is no container to store workflows in these frameworks. Workflows that were built before can be reused through the container. It need not be recalculated every time. Finally, these frameworks do not distinguish abstract workflows with specific workflows, while the distinction is beneficial for sharing workflows. (Although they have been distinguished in Pegasus [13], it does not put forward a mechanism to advertise abstract workflow with specific workflow service.)

26.4 Research Content and Key Problems of Web Service

Currently, the research aspects of Web Service home and aboard mainly focus on these two aspects. One is to build a descriptive language that can be understood between computers and that can fully describe the behaviors, functions, attributes, and restraints of Web Service; the other is to put forward a model or system structure that supports self-discovery, composition, and execution of services. The common objective is to use the content that can be stored both in the Web and the machine and create intelligence automatic service and business process infrastructures. Considering the combination of the above, implementing functional complementation is a natural choice.

In conclusion, build conceptual modeling on services using ontology domain model effectively in the Web Service, which can guide the design of Web service application. Combine Web Service and Semantic Web dynamically utilizing the information of semantics in the Web Service, which can improve the quality of Web service. The semantic Web is the combination of semantic Web and Web Service, which can offer effective support to the discovery, execution, explanation, and combination of Web Service. Among them, in the environment of distribution, the collection, expression of semantics, and the exploration of appropriate knowledge and inference method of Web Service are subjects that need further research.

26.5 Conclusions

Currently, for the research on Web service composition, the industrial field focuses on the research of business process; while the academic field makes research on automatic composition of Web Service with the core of semantics and inference by means of artificial intelligence planning. From the perspective of application, the techniques and methods put forward by the industrial field have good implementation ability, but its composition process is complicated and easy to have mistakes. The application systems that developed have relatively low flexibility and expandability. The methods put forward by the academy field have certain

automatic degree but it contains formalized descriptions. In conclusion, how to build a automatic, active and inferential Web Service and accomplish service check with semantic technologies and develop common and intelligent Web Service composition system are key problems that need to be considered during the process of Web Service working in the application field.

Acknowledgments Foundation Item: Education office in Anhui Province KJ2009B175Z.

References

1. Taylor I, Shields M, Wang I, Philp R (2003) Grid enabling applications using Triana, workshop on grid applications and programming tools, Seattle. In: conjunction with GGF8, vol 38(32), pp 749–755
2. Taylor I, Shields M, Wang I, Philp R (2003) Distributed P2P computing within Triana: a galaxy visualization test case. To be published in the IPDPS 2003 conference, vol 29(17), pp 84–93
3. IBMAlphaworks,BPWS4J. http://www.alphaworks.ibm.com/tech/bpws4j
4. Benatallah B, Sheng QZ, Dumas M (2003) The self-serv environment for web services composition. IEEE Internet Comput 7(1):40–48
5. Cardoso J, Sheth A (2002) Semantic e-workflow composition, vol 12(6). Technical report, LSDIS Lab, Computer Science, University of Georgia, pp 74–78
6. Sirin E, Hendler J, Parsia B (2003) Semi-automatic composition of web services using semantic descriptions. In: Web services: modeling, architecture and infrastructure workshop in conjunction with ICEIS, vol 35(4), pp 43–48
7. Chen L, Shadbolt NR, Goble C, Tao F, Cox SJ, Puleston C, Smart P (2003) Towards a knowledge-based approach to semantic service composition. In: 2nd international semantic web conference, vol 37(28), pp 88–95
8. Stevens RD, Robinson AJ, Goble CA (2003) MyGrid: personalised bioinformatics on the information grid. Bioinformatics 19(1):38–47 (Eleventh international conference on intelligent systems for molecular biology)
9. Blythe J, Deelman E, Gil Y, Kesselman C, Agarwal A, Mehta G, Vahi K (2003) The role of planning in grid computing. In: 13th international conference on automated planning and scheduling, vol 39(33), pp 75–83
10. Deelman E, Blythe J, Gil Y, Kesselman C (2002) Pegasus: planning for execution in grids, GriPhyN technical report 20(17):12–22
11. Deelman E, Blythe J, Gil Y, Kesselman C, Mehta G, Vahi K, Koranda S, Lazzarini A, Papa MA (2003) From metadata to execution on the grid pegasus and the pulsar search., GriPhyN technical report 20(15):54–63
12. Motta E, Domingue J, Cabral L, Gaspari M (2003) IRS-II: a framework and infrastructure for semantic web services. In: 2nd international semantic web conference (ISWC2003), vol 18(9). Sundial Resort, Sanibel Island, Florida, pp 20–23
13. Blythe J, Deelman E, Gil Y, Kesselman C (2003) Transparent grid computing: a knowledge-based approach. In: 15th innovative applications of artificial intelligence conference (IAAI 2003), vol 48(29), pp 748–755