AND/OR Graph and Search Algorithm for Discovering Composite Web Services

Qianhui Althea Liang, Singapore Management University, Singapore
Stanley Y.W. Su, University of Florida, USA

ABSTRACT

This paper presents a formalization of the Web Service composition problem as a search problem in an AND/OR graph, and a search algorithm for searching the graph to identify composite service(s) that satisfies a Web Service request. Given a service request that can only be satisfied by a composition of Web Services, we identify the service categories that are relevant to the request and dynamically construct an AND/OR graph to capture the input/output dependencies among the Web Services of these service categories. The graph is modified, based on the information provided in the service request. The search algorithm is then used to search the modified AND/OR graph for a minimal and complete composite service template that satisfies the service request. The algorithm can be applied repeatedly to the graph to search for alternative templates until the result is approved by the service requester. We have evaluated the algorithm both analytically and experimentally, and the experiment results are presented.

Keywords: AND/OR graph search algorithm; service discovery; Web Service composition

INTRODUCTION

Web Services Technology allow heterogeneous software and application systems to interoperate and enables organizations to share data, software, and hardware resources over the Internet. The functionalities of heterogeneous software and application systems can be published uniformly as Web Services and registered with a service registry. Standard protocols are provided to Internet users to find and invoke registered services.

Registered Web Services will have limited use, unless they can be combined automatically or semi-automatically to form composite services that meet more complex service needs of users. With the rapid increase in the number of Web Services supporting e-business solutions, the demand for sharing and integrating these Web-based autonomous, heterogeneous services in an automatic or semi-automatic way becomes even greater.

A composite service is a Web Service that is composed of a structure of some regis-
tered simple and/or composite services. The function of a composite service is an integration of the functions of its component services. Service composition refers to the construction of new composite services from registered services.

The existing Web Services model allows the creation, registration, and discovery of distributed Web Services. In this model, service providers and their services are registered with a public service registry (UDDI) (Bellwood, 2002). The public interfaces and binding information about the registered services are defined clearly and described in a standard service description language — the Web Service Description Language (WSDL) (WSDL, 2001). By querying the service registry and accessing the description document of a registered service, a service requestor is able to find out what the service does and how it can be accessed. The service thus can be invoked remotely through a lightweight messaging protocol — the Simple Object Access Protocol (SOAP) (Gudgin, 2003).

The discovery aspect of UDDI in the current Web Services model is restricted to the discovery of simple services; UDDI implementations of the current model cannot automatically discover or dynamically compose composite services. Related research focuses on two different aspects of service composition: constructing and binding. Research on constructing (composite service models) focuses on dynamically building the flow structures of service models, while research on binding (Zeng, 2003), focuses on selecting good-quality service providers and instantiating the models. This paper addresses the issue of constructing or composing the service models.

Reported work on constructing composite Web Services can be categorized as manual composition, automatic composition, and semi-automatic composition. In manual composition, a composite service is modeled manually by a structure of sub-services using a service flow language such as the Web Services Flow Language (WSFL, 2001) or the Business Process Execution Language (Andrews, 2003). The structure defines an e-business process model, and an invocation of the composite service is treated as an instance of the process model. Examples of this approach can be found in eFlow (Casati, 2000a, 2000b) of HP, the scenario-based service composition of the NTT Lab in Tokyo (Kiwata, 2001), the pattern-based process modeling introduced by Tut et al. (2002), constraint-driven composition introduced by Aggarwal (2004) and WSOM (2002), and TSSuite (Fontoura, 2003) of IBM.

In automatic composition, a discovery agent generates a structure of service operations of some registered services, based on the information provided in a service request. The discovered composite service then be invoked by the requestor and can be registered as a Web Service with the service registry for future use. Recently, several efforts were made to automate the service composition process. Some use rule systems to deduce a requested composite service from the available services (Ponnekanti, 2002; Thakkar, 2003). Some try to solve this problem by agent planning (McIlraith, 2002; Srivastava, 2003; Wu, 2003) and reasoning (Berardi, 2002; Wu, 2003). Others take ontology-based approaches (Arpina, 2004).

Both manual and existing fully automated approaches have some problems. Manual approaches involve a lot of human effort, which is not desirable. Fully automated approaches usually make some unrealistic assumptions. For example, rule-system-based approaches assume that the requestor knows the exact input and output interface of a desired composite service. Planning and reasoning approaches, on the other hand, assume that the requestor knows all the operations that constitute the desired composite service. Besides, the existing fully automated approaches use all registered services as the search domain for a composite service instead of restricting the search to those services that are relevant. These problems have motivated our research on a semi-automatic approach to service composition.

In our approach, an Intelligent Registry provides a GUI, which requestors use to describe in their own terms the service they look
Ontology and Service Oriented Programming
www.igi-global.com/article/ontology-service-oriented-programming/3063?camid=4v1a

Self-Reconfiguration of Service-Based System for Service Level Agreements and Resource Optimization
Ying Li, Kewei Sun, Jie Qiu and Ying Chen (2006). *International Journal of Web Services Research* (pp. 81-100).
www.igi-global.com/article/self-reconfiguration-service-based-system/3080?camid=4v1a