A Semi-Automatic Approach to Composite Web Services Discovery, Description and Invocation

Qianhui Liang, University of Florida, USA
Lakshmi N. Chakarapani, University of Florida, USA
Stanley Y. W. Su, University of Florida, USA
Raman N. Chikkamagalur, University of Florida, USA
Herman Lam, University of Florida, USA

ABSTRACT

With the popularity of the Web services technology, more and more software systems’ functionalities become available by being published and registered as Web services. Registered Web services need to be dynamically combined to form “composite services” when individual “simple services” fail to meet service requestors’ complex service needs. In this article, we propose a semi-automatic approach to composite Web services discovery, description and invocation. We present an intelligent registry with constraint matching capabilities to support composite service discovery and description. It provides a user interface to interactively compose a service request. It then uses a semi-automatic mechanism and a search algorithm to construct a composite service template that satisfies the request. The operations of the template are bound to registered service operations by constraint matching subsequently. The resulting composite service is specified in the Web Services Flow Language. A composite service processor is designed to execute composite services by invoking the component service operations of various service providers.

Keywords: composite service description and invocation; dynamic composite service discovery; search algorithm for composite service discovery; semi-automatic Web services composition; service constraint processing for composite service discovery

INTRODUCTION

Sharing data and software resources among collaborating organizations is the key to achieving their common goals. The Web services technology (Booth, 2003) being developed by the IT industry is a general framework that enables the sharing of heterogeneous data and software resources. It allows software/application systems’ functionalities to be defined and published as Web services and makes them programmatically accessible over the Internet. Sharable data can then be accessed through Web
services. Furthermore, the Web services technology enables the development of new application systems by making use of the existing applications that have been published as Web services. Its support for standard-based resource sharing while remaining platform-neutral has encouraged many organizations to apply the technology in various domains such as supply chain management, virtual enterprise, homeland defense, e-government, and e-business. The technology has drawn a lot of attention from software vendors as well as business and government organizations.

Although Web services technology provides a way of describing, registering, finding, and invoking different software applications running on a variety of platforms, there are two serious limitations, which motivated our R&D effort. First, the existing implementations of the service registry (e.g., IBM’s UDDI (UDDI, 2000) Registry v.2.0) are not able to discover, either automatically or semi-automatically, composite services that meet the service needs of requestors based on the services and service operations registered with the registry. A requestor has to use some service flow language (e.g., IBM’s Web Services Flow Language (Leymann, 2001), IBM and Microsoft’s Business Process Execution Language (Andrews, 2003) and Microsoft’s XLANG (Thatte, 2001)) to manually model a composite service as a structure of activities that bind to component Web services. The model is then instantiated and processed by a process execution engine. Second, the Web Service Definition Language (Christensen, 2001) used to define Web services does not provide constructs for specifying constraints on the input and output data of a service operation, on a service operation (e.g., the amount of time it takes to perform the operation), and/or on the service itself (e.g., the cost and qualify of the service). Due to this limitation, service providers cannot specify constraints associated with the services they provide. Nor can requestors explicitly specify their requirements on the services they are requesting. Without constraint specifications in service descriptions and service requests, service providers and service requestors are not able to adequately specify the services they provide or request. Without constraint processing capability, a service registry will not be able to accurately find the services that suit requestors’ needs.

In this article, we address the problems and approaches to discover, describe and invoke composite Web services. We present an intelligent registry, which consists of (a) a constraint-based broker, which accepts the registrations of simple and composite services with constraint specifications and performs the discovery of Web services by constraint satisfaction processing, (b) a query composer, which assists a requestor to select suitable service categories, services and service operations and interactively compose his/her service request, (c) a service dependency graph generator, which constructs a graph to capture the input and output dependencies of all the registered service operations under the requestor-selected service categories, (d) a service composer, which applies an AND-OR graph search algorithm on the service dependency graph to find a composition of service operations that can be bound to some registered service operations and also satisfies the service request, and (e) a composite service specification generator, which generates a specification document in Web Services Flow Language (WSFL) to describe the discovered composite service. The WSFL document is then given to a composite service processor (i.e., a WSFL execution engine) for processing.
Big Data Techniques for Supporting Official Statistics: The Use of Web Scraping for Collecting Price Data
www.igi-global.com/chapter/big-data-techniques-for-supporting-official-statistics/217860?camid=4v1a

Development of Distance Measures for Process Mining, Discovery and Integration
www.igi-global.com/article/development-distance-measures-process-mining/3107?camid=4v1a