A Web Service Composition Approach Based on Planning Graph and Propositional Logic

ShiYang Deng, Shandong University of Science and Technology, Qingdao, China
YuYue Du, Shandong University of Science and Technology, Qingdao, China
Liang Qi, New Jersey Institute of Technology, Newark, USA

ABSTRACT

To improve the computational efficiency of web service automatic composition, this article proposes a novel approach based on a planning graph and propositional logic. This approach has a forward searching stage and a backward combination stage: the former stage searches services in a service storage and performs the composition in a hierarchical architecture based on a planning graph, while the latter stage combines them via propositional logic operations. This method can obtain all composite services that contain no redundant services, and the computational complexity can be significantly reduced. Experiments are done to illustrate the effectiveness of the approach.

KEYWORDS

Logic Conjunctive Normal Form, Logic Disjunctive Normal Form, Minimum Cover Set, Planning Graph, Web Service Composition

INTRODUCTION

As self-describing and platform-independent computational elements, Web services are used to support service-oriented architectures, distributed computing and software reuse (Berardi, Cheikh, Giacomo, & Patrizi, 2008). A composite service is defined as a group of services that run in a certain order. It realizes functions that cannot be provided by a single service.

The rapid proliferation and development of services have drastically increased the choices for users, while searching services and composition become tedious and time-consuming. Many approaches and technologies have been proposed and applied to the composition and verification of services, such as Agent (Tong, Cao, Zhang & Li, 2011), workflows (Karakoc, Kardas & Senkul, 2006), Petri nets (Li, Fan, Sheng, Maamar & Zhu, 2011; Tan & Zhou, 2013), and semantic Web (Lecue & Mehandjiev, 2011). However, the aforementioned methods concentrate less on the organization and management of the function-same or similar services. Some researchers gather similar services into groups known service clusters (Nayak & Lee, 2007; Sun & Jiang, 2008; Ma, Chen, Hui & Wu, 2011; Sudha & Thamarai, 2006; Han, Wang & Cui, 2009) or service communities (Benslimane et al, 2007;
Maamar et al., 2008; Liu, Huang & Mei, 2007) to improve the efficiency of service discovery and composition. Logic Petri net is introduced to service clusters for modeling indeterminate parameters of services and formalizing deduction of service composition (Deng & Du, 2012; Hu, Du & Yu, 2014; Wu & Du, 2015). These methods aim at improving the efficiency of service composition, but do not consider removing redundant services of the composition.

As a powerful formal model, planning graph (Huang, Wei, Hu & Liu, 2009; Wei, Zhang, Huang, Chen, Hu & Liu, 2010; Deng, Huang, Wu, Yin & Li, 2013) has been used to construct service composition in many research studies. A planning graph is a directed acyclic hierarchical graph with nodes and arcs. There are two types of nodes in a planning graph, as shown in Figure 1: one type is service nodes, which are used to describe services, e.g., s_{11}-s_{23}, and another type is called parameter nodes, which are used to describe input or output parameters, e.g., p_{11}, p_{33}. In a planning graph, a set of service nodes can form a layer called a service node layer, while a set of parameter nodes can form a layer called the parameter node layer. Arcs are drawn from service nodes to parameter nodes or from parameter nodes to service nodes. Service composition based on a planning graph has a forward searching stage and a backward combination stage. The former is used to search services in a service storage to construct a planning graph that has parameter node layers and service node layers that appear alternately, as shown in Figure 1, where the nodes in the first parameter layer, called the initial layer, are the input parameters given by the users, and the nodes in the last parameter layer, called the target layer, contain all of the output parameters required by the user. The latter stage is a reverse searching process from the target layer to the initial layer. It is used to find services in the planning graph and combine them as a composite service that satisfies a user’s request.

There are two types of problems for Web service composition based on a planning graph. The first problem is that obtaining all of the composite services will cost an exponential amount of time. The second problem is called a redundant service problem, i.e., a composite service still satisfies the user’s request after a service, which is called a redundant service, and it is deleted from the composite service set. For example, in Figure 1, s_{22} is a redundant service, because the target parameters p_{21}-p_{23} can be obtained from the output of s_{21} and s_{23}.

Figure 1. A service composition modeled by a planning graph

![Planning Graph Diagram](image-url)
The Value of TAM Antecedents in Global IS Development and Research
www.igi-global.com/article/value-tam-antecedents-global-development/49657?camid=4v1a