Chapter 3
Multi-Level Web Service Clustering to Bootstrap the Web Service Discovery and Selection

Banage T. G. S. Kumara
Sabaragamuwa University of Sri Lanka, Sri Lanka

Incheon Paik
University of Aizu, Japan

Koswatte R. C. Koswatte
Sri Lanka Institute of Information Technology, Sri Lanka

ABSTRACT

Existing technologies for web services have been extended to give the value-added customized services to users through the service composition. Service composition consists of four major stages: planning, discovery, selection, and execution. However, with the proliferation of web services, service discovery and selection are becoming challenging and time-consuming tasks. Organizing services into similar clusters is a very efficient approach. Existing clustering approaches have problems that include discovering semantic characteristics, loss of semantic information, and a shortage of high-quality ontologies. Thus, the authors proposed hybrid term similarity-based clustering approach in their previous work. Further, the current clustering approaches do not consider the sub-clusters within a cluster. In this chapter, the authors propose a multi-level clustering approach to prune the search space further in discovery process. Empirical study of the prototyping system has proved the effectiveness of the proposed multi-level clustering approach.

DOI: 10.4018/978-1-5225-7268-8.ch003

INTRODUCTION

Service Oriented Architecture (SOA) has been a widely accepted paradigm to facilitate distributed application integration and interoperability (Endrei, Ang et al., 2004). Web services are the backbone technology that can be used to implement SOA. The technology allows users to publish software applications as universally accessible and programmable services. Users can use service composition techniques to combine the existing services as components to create a value-added services to solve complex problems (Paik, Chen et al., 2014). Service composition consists of four major stages: planning, discovery, selection, and execution. However, Web service discovery and selection are becoming a challenging and time-consuming task that require considerable efforts because of unnecessary similarity calculations in the matchmaking process within repositories such as Universal Description, Discovery and Integrations (UDDIs) and Web portals., especially with the continuously increase of Web services. Thus, researchers intensively studied to increase the performance of the service discovery and the selection by clustering Web services into similar groups to reduce the search space (Wu, Chen et al., 2014). Then, many unnecessary similarity calculations in the matching process can be avoided.

Current research studies on the Web service clustering is mainly focusing on two main approaches: functional based clustering and quality-of-service (QoS) based clustering. The functional based approaches are clustering services based on functional attributes such as *input*, *output*, *precondition* and *effect* (Dasgupta, Bhat et al., 2011; Nayak & Lee, 2007). The QoS based clustering approaches are based on QoS properties such as *cost* and *reliability* (Xia, Chen et al., 2011). The functional based clustering are mainly used to bootstrap the discovery process (Elgazzar, Hassan et al., 2010) and QoS based clustering approaches are used to reduce the search space in selection phase (Karthiban, 2014). Apart from these two categories, there are some other clustering approaches based on social properties of the services such as sociability (Chen, Paik et al., 2013) and users (Zheng, Xiong et al., 2013). In this paper, we mainly focused on the functional based clustering.

A principal issue for the clustering is computing the semantic similarity between services. First, the clustering method computes the similarity of features (*SoFs*) of the services. Then, the similarity of services (*SoS*) is computed as an aggregate of the individual *SoF* values. Several matrix-based methods have been used to compute the *SoF*s in current functional based clustering approaches, such as those using string-based cosine similarity (Platzer, Rosenberg et al., 2009), the one-to-one matching of features such as the *service name*, matching of service signatures such as the *messages* (Elgazzar, Hassan et al., 2010), the corpus-based normalized Google distance (NGD) (Liu & Wong 2009; Elgazzar, Hassan et al., 2010), knowledge-based ontology methods (Wagner, Ishikawa et al., 2011; Wen, Sheng et al., 2011; Xie,