Chapter 7.28
A Genetic Algorithm-Based QoS Analysis Tool for Reconfigurable Service-Oriented Systems

I-Ling Yen
University of Texas at Dallas, USA

Tong Gao
University of Texas at Dallas, USA

Hui Ma
University of Texas at Dallas, USA

ABSTRACT

Reconfigurability is an important requirement in many application systems. Many approaches have been proposed to achieve static/dynamic reconfigurability. Service-oriented architecture offers a certain degree of reconfigurability due to its support in dynamic composition. When system requirements change, new composition of services can be determined to satisfy the new requirements. However, analysis, especially QoS based analysis, is generally required to make appropriate service selections and service configurations. In this chapter, we discuss the development of QoS-based composition analysis techniques and propose a QoS specification model. The specification model facilitates QoS-based specification of the properties of the Web services and the requirements of the application systems. The composition analysis techniques can be used to analyze QoS tradeoffs and determine the best selections and configurations of the Web services. We develop a composition analysis framework and use the genetic algorithm in the framework for composition decision making. The framework currently supports SOA performance

analysis. The details of the genetic algorithm for the framework and the performance analysis techniques are discussed in this chapter.

INTRODUCTION

Reconfigurability is an important feature that is required in many modern application systems (Aksit & Choukair, 2003). For example, avionics systems, intelligent vehicle control systems, and remote monitoring systems frequently require dynamic adaptability so that the system can adapt to changes due to the failure of system components, reduced power level, unexpected operating conditions, and so forth. Also, some systems are multiple-mission and mission-specific, that is, they have to handle a class of missions that have similar system requirements but they also require adaptations to satisfy mission-specific needs.

Service-oriented architecture (SOA) is an ideal vehicle for achieving reconfigurability (Tsai, Song, Paul, Cao, & Huang, 2004). Desired functionalities can be achieved in SOA by dynamically composing appropriate services. Many research works focus on Web service composition (BEA et al., 2002; Hamadi & Benatallah, 2003; Sirin, Parsia, & Hendler, 2004). Most of these investigate various language issues. Business process execution language (BPEL) is an XML-based language for specifying the business processes and interaction protocols. Web services are then composed together to meet the specified functional requirements. The concept of semantic Webs extends the current Web with well-defined meanings for each Web service to facilitate their compositions. Based on semantic Webs, Sirin et al. (2004) have proposed a service composition tool including two main modules, namely, a composer and an inference engine. A user can interact with the composer to generate the composition and filter the results. The inference engine is a Web ontology language (OWL; W3C, 2004a) reasoner. The OWL reasoner searches the related services for the generated composition. Petri net-based algebra has also been used to help the composition by modeling the control flow. These composition methods can be used for adapting a system to new functional requirements (by composing a new set of services to achieve the modified functionalities).

Service composition generally focuses on composition of Web services to achieve some desired functionality. To achieve reconfiguration, new compositions can be derived for the modified functionalities. Actually, many adaptive systems require reconfigurability in terms of quality of service (QoS) behaviors. For example, some unmanned systems allow degraded QoS when some system failures occur. Many systems tradeoff the quality of their outputs versus the execution time to cope with periods of heavy loads or contentions for resources. In SOA-based systems, QoS reconfiguration can be achieved by, for example, selecting different services among those providing the same functionalities but, perhaps, having different QoS behaviors. Thus, systems can be dynamically assembled to fit the changing functional as well as QoS requirements.

Though SOA provides a convenient framework for reconfigurability, advanced techniques are still required to achieve actual adaptation (dynamic or static). For example, when the functional requirements of a system are modified, it is necessary to decompose the new requirements and then find the matching services in order to compose the new system. Similarly, when the QoS requirements of a system are modified due to changes in the execution environment, it is necessary to reconfigure the current services or choose new services to satisfy the new QoS requirements. To determine the correct service selections and configurations, it is necessary to analyze the QoS behaviors of the composed system.

The focus of this chapter is on QoS-based system reconfiguration. The goal is to develop analysis techniques and tools to facilitate static and dynamic system QoS behavior analysis such