Chapter 8
Issues on the Compatibility of Web Service Contracts

Surya Nepal
CSIRO ICT Centre, Australia

John Zic
CSIRO ICT Centre, Australia

ABSTRACT

In the Service Oriented Architecture (SOA) model, a service is characterized by its exchange of asynchronous messages, and a service contract is a desirable composition of a variety of messages. Though this model is simple, implementing large-scale, cross-organizational distributed applications may be difficult to achieve in general, as there is no guarantee that service composition will be possible because of incompatibilities of Web service contracts. We categorize compatibility issues in Web service contracts into two broad categories: (a) between contracts of different services (which we define as a composability problem), and (b) a service contract and its implementation (which we define as a conformance problem). This chapter examines and addresses these problems, first by identifying and specifying contract compatibility conditions, and second, through the use of compatibility checking tools that enable application developers to perform checks at design time.

INTRODUCTION

The Service Oriented Architecture (SOA) model is being promoted for use in the development of the next generation of large scale distributed applications. These applications are comprised of a collection of independent, autonomous, abstract services that are provided by business partners and third party service providers. Though the model is simple, the SOA vision of large-scale, cross-organizational distributed applications may be difficult to achieve due to the issues of incompatibility of service contracts and their respective implementations.

Some of the most significant issues arise because the SOA model is based on an abstraction...
paradigm, with internal operations of the service deliberately hidden. The service is defined solely through the sequence of messages communicated. Hoare (Hoare, 1995) and Milner (Milner, 1989) recognized this in their seminal works in process algebra. Such system descriptions that abstract away internal operations may result in composed behaviors that differ from each other despite apparently offering the same message sequence inputs and outputs. Potentially, these differences lead to undesirable behaviors such as deadlocks or starvation. Avoiding these undesirable service behaviors may be achieved by composing this type of system with an “adaptor” that essentially prohibits those undesirable states ever being reached, while still preserving the external communication. This approach is an active area of research within the business process community (Benatallah, Casati, Grigori, Nezhad & Touami, 2005). However, these adaptors are typically written by the service providers themselves. Because they are written by and are a part of the provider’s service, the provider has had the opportunity to be aware of some of the internal operations of the service. They have “looked inside the box” and realized the operations that lead to undesirable behaviors. In this book chapter, we present an alternative methodology and support tools that allow the rigorous development of service contracts and their implementations. This methodology is able to assure completeness, acceptable consistent state and termination for the service. There is no requirement to “look inside the box” in order to achieve a composed system that behaves as expected, and there is no need to build specialized adaptors.

Problem Statement

Building applications using the SOA model is getting easier with the good support from Web services standards (BPEL4WS, 2002; SSDL, 2005; WS-BA, 2005) and support tools. Because of the availability of these standards and tools, architects are able to specify service-oriented applications, as well as allow developers the capacity to build service-oriented applications. Nonetheless, as with any distributed systems, issues of deadlocks, race conditions, failures and exceptions, concurrency and asynchrony remain, and need careful attention. Previously, similar issues were faced by the distributed transaction community with partial resolutions being based on the use of ACID based technologies (Elmagarmid, 1992; Gray & Reuter, 1992). However, as these solutions are for tightly coupled, short-lived applications, they are inappropriate in the autonomous, loosely coupled, cross-organizational SOA environment.

Analysis of service compatibility issues is highly amenable to model checking and the judicious application of formal methods. A growing body of work is examining the use and application of formal methods and model checking to establishing the compatibility of service-oriented applications. Compatibility issues are studied in two broad categories: (a) between a service contract and its implementation (Nakajima, 2002) and (b) between contracts of different services (Bultan, Fu, Hall & Su, 2003; Foster, Uchitel, Kramer, & Magee, 2004; Greenfield, Kuo, Nepal, & Fekete, 2005; Narayanan & McIlraith, 2002). We define the first category as a composability problem (Milanovic and Malek, 2004)) and the second category as a conformance problem.

Figure 1 shows the distinction between the conformance and composition relations between service contracts and service implementations.
Related Content

Automatic Construction of Service Network based on OpenCyc
[www.igi-global.com/article/automatic-construction-of-service-network-based-on-opencyc/124982?camid=4v1a](www.igi-global.com/article/automatic-construction-of-service-network-based-on-opencyc/124982?camid=4v1a)

Constructing Home Network Systems and Integrated Services Using Legacy Home Appliances and Web Services
[www.igi-global.com/article/constructing-home-network-systems-integrated/3116?camid=4v1a](www.igi-global.com/article/constructing-home-network-systems-integrated/3116?camid=4v1a)

A Logic Programming Based Framework for Intelligent Web Service Composition
[www.igi-global.com/chapter/logic-programming-based-framework-intelligent/26080?camid=4v1a](www.igi-global.com/chapter/logic-programming-based-framework-intelligent/26080?camid=4v1a)

The Development, Testing, and Deployment of a Web Services Infrastructure for Distributed Healthcare Delivery, Research, and Training
[www.igi-global.com/chapter/development-testing-deployment-web-services/26072?camid=4v1a](www.igi-global.com/chapter/development-testing-deployment-web-services/26072?camid=4v1a)