Chapter 5
A Model-Based Approach for Diagnosing Fault in Web Service Processes

Yuhong Yan
Concordia University, Canada

Philippe Dague
University Paris-Sud 11, France

Yannick Pencolé
LAAS-CNRS, France

Marie-Odile Cordier
IRISA, France

ABSTRACT

Web services based on a service-oriented architecture framework provide a suitable technical foundation for business process management and integration. A business process can be composed of a set of Web services that belong to different companies and interact with each other by sending messages. Web service orchestration languages are defined by standard organizations to describe business processes composed of Web services. A business process can fail for many reasons, such as faulty Web services or mismatching messages. It is important to find out which Web services are responsible for a failed business process because we could penalize these Web services and exclude them from the business process in the future. In this paper, we propose a model-based approach to diagnose the faults in a Web service-composed business process. We convert a Web service orchestration language, more specifically BPEL4WS, into synchronized automata, so that we have a formal description of the topology and variable dependency of the business process. After an exception is thrown, the diagnoser can calculate the business process execution trajectory based on the formal model and the observed evolution of the business process. The faulty Web services are deduced from the variable dependency on the execution trajectory. We demonstrate our diagnosis technique with an example.

DOI: 10.4018/978-1-61350-104-7.ch005
1 INTRODUCTION

Web services not only function as middleware for application invocation and integration, but also function as a modeling and management tool for business processes. In a Service Oriented Architecture paradigm, a business process can be composed of Web services distributed over the Internet. This kind of business processes can be flexible and optimal by using the best services from multiple companies. Various Web service process description languages are designed by standard bodies and companies. Among them, Business Process Execution Language for Web Service (BPEL4WS, denoted as BPEL after) (Andrews, Curbera, Dholakia, Goland, & et.al., 2003) is the de facto standard used to describe an executable Web service process. In this paper, we study the behavior of a business process described in BPEL.

As any other systems, a business process can fail. For a Web service process, the symptom of a failure is that exceptions are thrown and the process halts. As the process is composed of multiple Web services, it is important to find out which Web services are responsible for the failure. If we could diagnose the faulty Web services, we could penalize these Web services and exclude them from the business process in the future. The current throw-and-catch mechanism is very preliminary for diagnosing faults. It relies on the developer associating the faults with exceptions at design time. When an exception is thrown, we say certain faults occur. But this mechanism does not guarantee the soundness and the completeness of diagnosis.

In this paper, we propose a model-based approach to diagnose faults in Web service processes. We convert the basic BPEL activities and constructs into synchronized automata whose states are presented by the values of the variables. The process changes from one state to another by executing an action, e.g. assigning variables, receiving or emitting messages in BPEL. The emitting messages can be a triggering event for another service to take an action. The diagnosing mechanism is triggered when exceptions are thrown. Using the formal model and the runtime observations from the execution of the process, we can reconstruct the unobservable trajectories of the Web service process. Then the faulty Web services are deduced based on the variable dependency on the trajectories. Studying the fault diagnosis in Web service processes serves the ultimate goal of building self-manageable and self-healing business processes.

This paper is organized as follows: section 2 analyzes the fault management tasks in Web service processes and motivates the use of Model-based Diagnosis (MBD) for Web services monitoring and diagnosis; section 3 presents the principles for MBD; section 4 formally defines the way to generate an automaton model from a BPEL description; section 5 extends the existing MBD techniques for Web service monitoring and diagnosis; section 6 is the related work, and section 7 is the conclusion.

2 ADVANCED FAULT MANAGEMENT FOR WEB SERVICE PROCESSES

A Web service process can run down for many reasons. For example, a composed Web service may be faulty, an incoming message mismatches the interface, or the Internet is down. The symptom of a failed Web service process is that exceptions are thrown and the process is halted. The current fault handling mechanism is throw-and-catch, similar to programming languages. The exceptions are thrown at the places where the process cannot be executed. The catch clauses process the exceptions, normally to recover the failure effects by executing predefined actions.

The throw-and-catch mechanism is very preliminary for fault diagnosis. The exception reports where it happened and returns some fault information. The exceptions can be regarded as associated with certain faults. When an exception is thrown,
Related Content

Towards a Framework for Agent-Enabled Semantic Web Service Composition
[www.igi-global.com/article/towards-framework-agent-enabled-semantic/3045?camid=4v1a](www.igi-global.com/article/towards-framework-agent-enabled-semantic/3045?camid=4v1a)

Issues on the Compatibility of Web Service Contracts
[www.igi-global.com/chapter/issues-compatibility-web-service-contracts/59923?camid=4v1a](www.igi-global.com/chapter/issues-compatibility-web-service-contracts/59923?camid=4v1a)

Service Discovery and Composition Based on Contracts and Choreographic Descriptions
Mario Bravetti and Gianluigi Zavattaro (2013). *Adaptive Web Services for Modular and Reusable Software Development: Tactics and Solutions* (pp. 60-88).
[www.igi-global.com/chapter/service-discovery-composition-based-contracts/69470?camid=4v1a](www.igi-global.com/chapter/service-discovery-composition-based-contracts/69470?camid=4v1a)

Efficient Transport Bindings for Web Service Messages
[www.igi-global.com/chapter/efficient-transport-bindings-web-service/31211?camid=4v1a](www.igi-global.com/chapter/efficient-transport-bindings-web-service/31211?camid=4v1a)