An Automated Approach for Fault Injection Testing
of BPEL Orchestrations

Dessislava Petrova-Antonova\textsuperscript{1,2}, Sylvia Ilieva\textsuperscript{1,2}, Vera Stoyanova\textsuperscript{2},
Ilina Manova\textsuperscript{3}, and Valentin Pavlov\textsuperscript{3}

\textsuperscript{1}IICT-BAS, \textsuperscript{2}Sofia University, \textsuperscript{3}Rila Solutions
1113 Sofia, Bulgaria
d.petrova@fmi.uni-sofia.bg, sylvia@acad.bg,
vera.stoyanova@yahoo.com, {ilinam,vpavlov}@rila.bg

Abstract. Web service orchestrations are widely adopted solution for
development of loosely-coupled distributed applications. In addition to
traditional defects causing failures in the software systems, their quality is also
affected by additional problems such as network latency, interface
inconsistency or communication issues. The fault injection testing is useful for
validation the behavior of the web service orchestrations when such problems
occur. That is why we propose an approach based on fault injection technique
for generation and execution of fault tolerance test cases. The approach is
automated through implementation of two software tools for fault injection and
test case generation and execution Those tools are integrated in a common
testing framework, called TASSA, presented in this paper. They are validated
on a case study through simulation of different type of failures and fault
tolerance testing of a web service orchestration.

Keywords: BPEL orchestrations, fault injection testing, web services, test
automation.

1 Introduction

The wide adoption of Service-Oriented Architecture (SOA) and its web service based
implementation brings new research challenges related to development of complex
business processes and their subsequent testing [1]. Fault tolerance of such processes
is critical since the proper work of many client applications depends on them. The
business processes have to provide high-quality software services even if the
execution conditions are abnormal and crashes occur.

The widely used testing tools and frameworks are dealing with logic and
programming errors in the development process causing defects in the software
systems. However, the software quality is affected also by additional external factors
such as hardware crashes, communication interruption, unexpected user behavior and
so on. A well known testing technique that can be applied in such situations is the
fault injection. Its main objective is to test the fault tolerance through injecting faults
and verify whether the software system can handle the faults and recover from them.
There are several challenges regarding business process testing, which affect the usage of fault injection technique:

- The complexity of the SOA architecture and therefore of its implementations – the typical service-based systems and in particular business processes integrate a large number of services, that are developed with various software technologies and provided by different providers. The usage context and the quality requirements of these services are not known at the time of development. That is why the testing process requires simulation of large number situations with unexpected condition that is possible through usage the fault injection technique.

- The latency of faults – Some faults can exist in the business process without causing failures. For example, the business process may invoke certain service returning invalid responses in a very few situations, which might not happen in a long time. Such situations are useful to be simulated through fault injection.

- The dynamic nature of SOA – The business processes based on SOA orchestrate services that are changed dynamically. The service providers can produce new versions of their services, temporally stop or even undeploy them. Additionally some of the consumed services can be replaced with new ones providing the same functionality. This in turn could cause unexpected behavior of the business process that can be simulated through fault injection.

This paper presents an approach for fault injection testing of business processes described with Business Process Execution Language for Web Services (WS-BPEL). It is implemented in two software tools, namely Fault Injection Tool (FIT) and Test Case Generation and Execution Tool (TCGET) that are parts of a common testing framework for BPEL orchestrations, called Testing as a Service Software Architecture (TASSA) [2]. The goal of the framework is to provide a set of tools and common platform for end-to-end testing of WS-BPEL orchestrations. It can be used in different phases of software process, namely implementation, integration, verification and validation, and maintenance phase.

The rest of the paper is organized as follows. Section 2 describes the FIT and TCGET. Section 3 presents a case study showing fault injection testing of a sample business process. Section 4 outlines the related work. Section 5 concludes the paper and gives directions for future work.

2 Tool Support for Fault Injection Testing of TASSA Framework

The TASSA framework provides a methodology and set of tools for functional as well as non-functional testing business processes described with WS-BPEL. The functional testing includes path coverage according to functional requirements, full branch coverage with shortest path length of the business process, and path coverage according to new functionality. The non-functional testing is achieved through robustness and scalability testing. The robustness testing is conducted by simulations of possible failures of unreliable service(s) and tests compositions of both reliable and unreliable services. The scalability testing covers many configuration changes.