Chapter 100
Service Composition Verification and Validation

Manuel Palomo-Duarte
University of Cadiz, Spain

ABSTRACT

Web services are changing software development thanks to their loosely coupled nature and simple adoption. They can be easily composed to create new more powerful services, allowing for large programming systems. Verification and validation techniques try to find defects in a program to minimize losses that its malfunction could cause. Although many different approaches have been developed for "traditional" program testing, none of them have proven definitive. The problem is even more challenging for new paradigms like web services and web service compositions, because of their dynamic nature and uncommon web service-specific instructions. This chapter surveys the different approaches to web service and web service composition verification and validation, paying special attention to automation. When no tools are available for a given technique, academic efforts are discussed, and challenges are presented.

INTRODUCTION

According to the Software Engineering Body of Knowledge - SWEBOK (Abran & Moore, 2004),

verification and validation addresses software product quality directly and uses testing techniques which can locate defects so that they can be addressed.

Software *Verification and Validation* (*V&V*) has been a key problem in Computer Science since the so-called *Software Crisis*, causing monetary and human losses (Leveson, 1995). Many efforts have been made since then to develop computer programs that meet system requirements, but up to date, no solution has proven definitive (Myers, Badgett, Thomas, & Sandler, 2004). This way, a "good" quality can only be achieved by carefully combining different techniques.

Service Oriented Architectures (SOA), and Web Services (WS) are changing software development. They greatly ease system interoperability, and new WS can be built on them using service composition technologies. But they also include some peculiarities uncommon in other paradigms (like WS-specific instructions), so V&V techniques have to be adapted accordingly (Bozkurt, Harman, & Hassoun, 2010). This chapter surveys the different approaches to web service and web service composition V&V, paying special attention to the tools available to automate them.

The rest of the chapter is divided into five sections. The first one provides some background information to make the text self-contained. The second section, *Service Verification and Validation*, deals with V&V techniques for external WS, given that we don't have access to their code (i.e. *black-box testing*). The following one explains additional techniques to test the internal logic of a composition of different WS, that is *white-box testing*. Along these two sections we comment tools[1] to automate the different software testing techniques, or proposals showing how to implement them. In the fourth and fifth sections, we compile future research directions, and draw some conclusions about the state of the art of service and service composition V&V. Finally, we include the references, some additional readings and a glossary of key terms and definitions.

BACKGROUND

Service Composition

WS usually publish their available operations using WSDL files. WSDL is the XML-based language standardized by W3C® (WSDL Technical Committee, 2007) to describe WS. WSDL description files include all the information needed for the interaction between service invoker and provider: the URI where the WS waits for requests, the different kinds of messages accepted, etc. WSDL specifications use XML Schema data types to describe the messages exchanged.

Thanks to their platform and language independence, WS can be not only invoked in traditional programs, but they can also be easily composed to create new higher-level ones that suit costumer needs (Newcomer & Lomow, 2004). This is sometimes called *programming in the large*. There are two ways of composing services: choreography and orchestration (Papazoglou, 2006). When orchestrating there is a director process that implements all the logic of the composition, whereas in choreography that logic is distributed over the different partner services.

Orchestration is simpler, as it can use independent services, and is easier to monitor. This has led to a much wider industrial support: the WS-BPEL (Web Services Business Process Execution Language) orchestration language OASIS® standardization committee was formed by specialists from the leading IT companies: Oracle®, Microsoft®, IBM®, HP®, etc. (WS-BPEL Technical Committee, 2007). Additionally, BPMN (Business Process Modeling Notation) has backed WS-BPEL supporting automatic translation from BPMN graphical models to it (Business Process Management Initiative, 2010). In contrast, the W3C language for choreography, WS-CDL, remains as a Candidate Recommendation since 2005 (Web Services Choreography Working Group, 2005).

WS-BPEL is a language for static web service orchestration. It includes the usual instructions in imperative languages for data manipulation, assignments, loops, and other WS-specific ones for web service invocation, timeout event processing, compensation for faulty WS, etc. WS-BPEL compositions are deployed in a WS-BPEL engine, usually contained in an application server, that creates a new process when an invocation to a service is received.

There are other languages that allow dynamic discovery and composition of semantic WS (WS including semantic information in their description), namely OWL-S and WSMO. Nevertheless,

Related Content

Analysis and Comparison of Hash Functions in Terms of Security and Performance
Öner Özerand Gurkan Tuna (2023). *Cyber-Physical Systems and Supporting Technologies for Industrial Automation (pp. 64-89).*
www.irma-international.org/chapter/analysis-and-comparison-of-hash-functions-in-terms-of-security-and-performance/328493

A Quantitative Risk Assessment Model for the Management of Software Projects
Dan Shoemaker (2003). *Practicing Software Engineering in the 21st Century (pp. 97-115).*
www.irma-international.org/chapter/quantitative-risk-assessment-model-management/28113

Analysis of Cloud and Self-Web-Hosting Services Based on Security Parameters
Surbhi Khareand Abhishek Badholia (2022). *International Journal of Information System Modeling and Design (pp. 1-14).*
www.irma-international.org/article/analysis-of-cloud-and-self-web-hosting-services-based-on-security-parameters/297629

Applying AHP for Collaborative Modeling Evaluation: Experiences from a Modeling Experiment
Denis Ssebuggwawo, Stijn Hoppenbrouwersand Henderik A. Proper (2013). *International Journal of Information System Modeling and Design (pp. 1-24).*
www.irma-international.org/article/applying-ahp-collaborative-modeling-evaluation/75462

Software Cost Estimation and Capability Maturity Model in Context of Global Software Engineering
Ayub Muhammad Latif, Khalid Muhammad Khanand Anh Nguyen Duc (2022). *Research Anthology on Agile Software, Software Development, and Testing (pp. 910-928).*
www.irma-international.org/chapter/software-cost-estimation-and-capability-maturity-model-in-context-of-global-software-engineering/294501