

Chapter 22

Dynamically Reconfigurable Architectures: An Evaluation of Approaches for Preventing Architectural Violations

Marek Rychly

Brno University of Technology, Czech Republic

ABSTRACT

Dynamic aspects of behavior of software systems in dynamically reconfigurable runtime architectures can result in significant architectural violations during runtime. In such cases, a system's architecture evolves during the runtime according to the actual state of the system's environment, and consequently, runtime reconfigurations may eventually lead to incorrect architecture configurations that were not considered during the system's design phases. These architectural violations are known as architectural erosion or architectural drift, and they contribute to an increasing brittleness of the system, or a lack of its coherence and clarity of its form. This chapter describes and compares possible measures to prevent architectural violations in dynamic service and component models. The aim of this chapter is to evaluate the applicability of those measures in combination with advanced features of reconfigurable runtime architectures such as ad hoc reconfiguration, service or component mobility, composition hierarchy preservation, and architectural aspects.

INTRODUCTION

Current information systems tend to be designed as component-based systems and often utilize Service Oriented Architecture (SOA) and Web service technology. The service orientation enables decomposition of a complex software system into a collection of cooperating and autonomous components known as services. These services cooperate with each other to provide a particular functionality of the implemented software system with defined quality.

Loose binding between the services, which represent individual components of a system, enables *runtime reconfigurations* of the system architectures. In other words, it enables creating, destroying, and

DOI: 10.4018/978-1-5225-3923-0.ch022

updating the services, and establishing and destroying their interconnections dynamically at runtime, on demand, and according to various aspects to move the services into different contexts and to different providers (i.e., service mobility). Eventually, a series of reconfigurations contributing to the evolution of the architecture, of a supposedly well-designed system may lead to incorrect architecture configurations that were not considered during the system's design phase. These incorrect configurations are commonly known as *architectural violations*.

This chapter describes and compares possible measures to prevent the architectural violations, as they are used in the current state-of-the-art approaches. The goal is to evaluate applicability of those measures in combination with the advanced features of dynamic architecture such as ad hoc reconfiguration, service or component mobility, composition hierarchy preservation, and architectural aspects.¹ Specific objectives include an introduction to the problems of dynamically reconfigurable runtime architectures, an analysis of the state-of-the-art approaches in this field with focus on the advanced features of dynamic architectures, and the methods to prevent architectural violations.

The chapter is organized as follows. The next section deals with software architecture in general and introduces component-based development and service-oriented architecture with concepts of dynamically reconfigurable runtime architectures. We also describe several important state-of-the-art works dealing with component-based development and component models supporting features of dynamic and mobile architectures. In the following section, we discuss existing problems relating to the support of dynamic and mobile architectures that cause architectural violations in component-based or service-oriented systems.

Then, we outline possible strategic improvements and introduce approaches to prevent the architectural violations in general, and also describe their applications in the current state-of-the-art related works. The next part of the chapter deals with the evaluation of the previously described approaches for preventing architectural violations. More specifically, we analyze compatibility of the approaches with the advanced features of dynamically reconfigurable runtime architectures. Finally, we discuss future research directions such as possibilities of utilization of the advanced features of dynamically reconfigurable runtime architectures including previously described methods of preventing architectural violations in implementations of service-oriented architectures.

BACKGROUND

According to IEEE (2000), software architecture is defined as the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution. Another definition by Bass et al. (2003) adds that the architecture describes only externally visible properties of components, i.e., it is an abstraction of a system that suppresses details of components, except for services published by interfaces, relationships to environment of the components, and their externally observable behavior.

Oquendo (2004) distinguished three types of software architectures according to their evolution which depends on changes to their environment: static architecture, dynamic architecture, and mobile architecture. The last one is also known as a fully dynamic architecture.

Architecture of a software system is the *static architecture* if there are no changes to the system's structure during runtime. After initialization of the system, there are no new connections between the system's components and existing connections are not destroyed.

16 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:
www.igi-global.com/chapter/dynamically-reconfigurable-architectures/192892

Related Content

Fuzzy Linear Multi-Objective Stochastic Programming Models

(2019). *Multi-Objective Stochastic Programming in Fuzzy Environments* (pp. 78-127).

www.irma-international.org/chapter/fuzzy-linear-multi-objective-stochastic-programming-models/223803

Fuzzy Multi-Objective Programming With Joint Probability Distribution

(2019). *Multi-Objective Stochastic Programming in Fuzzy Environments* (pp. 263-295).

www.irma-international.org/chapter/fuzzy-multi-objective-programming-with-joint-probability-distribution/223807

Exploring Cyber Security Vulnerabilities in the Age of IoT

Shruti Kohli (2018). *Cyber Security and Threats: Concepts, Methodologies, Tools, and Applications* (pp. 1609-1623).

www.irma-international.org/chapter/exploring-cyber-security-vulnerabilities-in-the-age-of-iot/203577

Spatial Data Mining, Spatial Data Warehousing, and Spatial OLAP

Amira M. Idrees, Mostafa Lamlom Ahmed Khaled and Amal Hassan Ali Talkhan (2018). *Emerging Trends in Open Source Geographic Information Systems* (pp. 97-132).

www.irma-international.org/chapter/spatial-data-mining-spatial-data-warehousing-and-spatial-olap/205158

A Methodology for Model-Based Reliability Estimation

Mohd Adham Isa and Dayang Norhayati Abang Jawawi (2018). *Computer Systems and Software Engineering: Concepts, Methodologies, Tools, and Applications* (pp. 461-484).

www.irma-international.org/chapter/a-methodology-for-model-based-reliability-estimation/192888