Dynamically Reconfigurable Runtime Architectures: Challenges and Service-driven Approaches
(Chapter Proposal to “Emerging Architectural Trends in Service-Driven Computing”)

Marek Rychly <rychly@fit.vutbr.cz>
Department of Information Systems, Faculty of Information Technology, Brno University of Technology, Bozetechova 2, 612 66 Brno, Czech Republic

Abstract
Dynamic aspects of behaviour of software systems with dynamically re-configurable runtime architectures can result into significant architectural violations during their runtime. In such cases, a system's architecture evolves during the runtime according to the actual state of the system's environment and consequent runtime re-configurations 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 a system or a lack of its coherence and clarity of its form. This chapter will describe and compare possible measures to prevent the architectural violations, as they are used in dynamic service and component models. The aim of this chapter is to evaluate applicability of those measures in combination with advanced features of re-configurable runtime architectures such as ad-hoc re-configuration, service or component mobility, composition hierarchy preservation, and architectural aspects.

Keywords
Software architecture, Architectural violation, Dynamic architecture, Runtime re-configuration

Introduction
Current information systems tend to be designed as component-based systems and often utilise service-oriented architecture (SOA) and Web service technology. The service orientation allows to decompose 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, allows runtime re-configurations of the systems' architectures, i.e., to create, destroy, and update the services and to establish and destroy their interconnections dynamically at the runtime, on demand, and according to various aspects; to move the services into different contexts and to different providers (i.e., service mobility); etc. Eventually, a series of re-configurations, i.e., 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.

This problem of evolving architectures is known as the problem of architectural drift and architectural erosion [1]. The architectural drift is defined as an insensitivity about a system's architecture that, with increasing evolution, leads to its in-adaptability and a lack of coherence and clarity of form. The architectural erosion is defined as violations of a
system's architecture that lead to the significant problems in the system and contribute to its increasing brittleness. It may be caused by the unrestrained evolution of the architecture as well as by violation of the architecture that become obscured due to the architectural drift.

Current approaches to dynamic architecture address the problems of architectural violations and their prevention in different ways. Basically, the approaches prevent a system from the architectural violations by means of predefined design-time rules and specific runtime restrictions. Generally, these measures often result into limited re-configuration possibilities, which may interfere with advanced features of dynamic architecture.

**Goal and Objectives**

This chapter will describe and compare 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 re-configuration, service or component mobility, composition hierarchy preservation, and architectural aspects.

Specific objectives include an introduction to the problems of dynamically re-configurable 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 of architectural violations prevention, an evaluation of compatibility of the advanced features of dynamic architectures with the methods of architectural violations prevention, and a discussion of utilisation issues of the previously evaluated methods of architectural violations prevention in combination with the advanced features of dynamic architectures.

**Organisation**

The chapter will be organised as follows. It will start with an introduction to dynamically re-configurable runtime architectures and the problem of evolving architectures (in the similar way as it is described in the Introduction of this Chapter Proposal). The second section will describe the approaches to prevent the architectural violations in general and also their applications in the current state-of-the-art related works. Then, next section will deal with the evaluation of the previously described approaches in combination with the advanced features of dynamically re-configurable runtime architectures to check their compatibility. The fourth section will discuss possibilities of utilisation of the advanced features of dynamically re-configurable runtime architectures including previously described methods of the architectural violations prevention in implementations of service-oriented architectures, as the described features and methods are primarily related to component-based architectures in general and specific adaptations to service-oriented architectures may be needed. Finally, the last section will sum up the evaluation's results, provide a conclusion, and outline possibilities of future research in this field.

**References**