Chapter 9
Validating Autonomic Services: Challenges and Approaches

Tariq M. King
Ultimate Software Group, Inc., USA

Mohammed Akour
North Dakota State University, USA

Peter J. Clarke
Florida International University, USA

Annaji S. Ganti
Microsoft Corporation, USA

ABSTRACT

Autonomic service-driven applications represent a new realm of software that can discover new capabilities, automatically integrate with other systems, and adapt to changing system environmental conditions. For the past many years, researchers and practitioners have been investigating, prototyping, and evaluating these self-configuring, self-healing, self-optimizing, and self-protecting systems. Although validation is expected to play a key role in the success of autonomic systems, there are few works that address this topic. Dynamic adaptation in autonomic software results in structural and behavioral runtime changes, which cannot be validated offline at design-time. Runtime testing has therefore emerged as a possible solution to validating dynamic adaptations in autonomic software. This chapter summarizes the state-of-the-art in runtime testing of autonomic systems, describes key challenges associated with runtime testing, and provides guidelines for integrating runtime testing approaches into autonomic software using self-testing architectures. Finally, directions for future research for validation of autonomic components are discussed.

DOI: 10.4018/978-1-4666-6178-3.ch009

INTRODUCTION

Service-driven computing provides a software development model in which user needs are represented as services that are integrated to provide a software solution. The trend towards service-oriented architectures, Web and Grid services, and Cloud computing suggests that the service-driven paradigm is leading the way for building the next-generation systems. The grand vision of autonomic computing portrays these next-generation systems as ones that can configure, heal, optimize, and protect themselves (Kephart & Chess, 2003). Researchers have been steadily moving towards that vision through the development and evaluation of approaches and prototypes for autonomic service-driven applications.

Autonomic systems continually seek to fulfill one or more goals, typically specified through a set of high-level policies. To achieve system goals, services can be added, removed, replaced, and composed at runtime - a process referred to as dynamic software adaptation (Zhang et al., 2004). *Dynamic software adaptation* enables the system to automatically evolve by adding new capabilities after the system has been deployed to production. However, dynamic adaptation also presents new software engineering research challenges (Salehie & Tahvildari, 2009).

As the vision of software systems that configure, heal, optimize, and protect themselves starts to become a reality, academic researchers and industry practitioners must consider the implications of autonomic computing on software quality. Incorporating self-management features into software increases its complexity, thereby making it more difficult to validate at development-time. Furthermore, since these systems can dynamically modify their own structure and behavior, runtime testing must be performed to avoid costly system failures (King et al., 2007; Costa et al., 2010; Tamura et al., 2013).

With software testing being the de-facto standard used for validating software in industry, it is expected to play a major role in the success of autonomic service-driven computing. In this chapter, we discuss issues and possible solutions for testing autonomic systems. More specifically, we focus on the use of runtime testing as an emerging solution for validating autonomic service-driven applications. The mission of this chapter includes the following objectives: summarize the current state-of-the art in runtime testing of dynamically adaptive autonomic systems; identify and describe the key challenges associated with validating autonomic service-driven applications at runtime; discuss proposed solutions that address the identified key challenges, and provide guidelines and recommendations to implement practical runtime testing solutions for autonomic software.

The rest of the chapter is organized as follows. The background section introduces the concept of software testing, specifically runtime testing of autonomic systems, and discusses related works. The next section describes and discusses the challenges in testing autonomic systems, which is followed by a presentation of approaches to runtime testing of autonomic and adaptive services. Finally, promising future directions for closing the gap in the current state-of-the-art for runtime testing of autonomic service-driven systems is presented and concluded.

BACKGROUND

This section contains background material on software testing that is necessary for understanding the chapter. It also provides a literature review of research on the validation and verification of autonomic and adaptive software systems.

Software Testing

Software validation seeks to ensure that a product meets the expectations of the customer. While there are several available validation techniques, testing continues to be the primary means of validation