A Distributed Test Architecture for Adaptable and Distributed Real-Time Systems

Mariam Lahami, Moez Krichen and Mohamed Jmaiel

Research Unit of Development and Control of Distributed Applications
National School of Engineering of Sfax, University of Sfax
Sokra road km 4, PB 1173 Sfax, Tunisia
mariam.lahami@redcad.org,moez.krichen@redcad.org
mohamed.jmaiel@enis.rnu.tn,
http://www.redcad.org

Abstract. This work focuses on testing the consistency of distributed real-time systems when their configurations evolve dynamically, called also adaptable systems. In this context, runtime testing which is carried out on the final execution environment is emerging as a new solution for the validation of these systems. To reduce testing effort, cost and time, we apply the dependency analysis technique in order to identify affected parts of the system under test due to runtime reconfiguration. In addition, we propose a flexible and evolvable distributed test architecture made of two kinds of testers: *Single Component Testers* and *Component Composition Testers*. These testers execute unit tests (respectively integration tests) on the affected components (respectively component compositions) as soon as reconfiguration actions occur. An illustrative example describing interactions between the proposed testers when two reconfiguration scenarios happen is given.

1 Introduction

Distributed real-time systems become increasingly important in a wide range of application fields, e.g. in the system automation, aerospace, robotics and so on. A real-time system is defined in Schutz (1993) as a system which has to adhere not only to functional requirements but also to temporal requirements, often called "timing constraints" or "deadlines". If a computation is activated by a stimulus from the environment, it must be completed before the specified deadline. Therefore, the system correctness depends not only on the logical results of a computation but also on the time at which the results are produced.

Currently, real-time systems are being implemented as distributed systems. This issue is manifested by an execution environment composed of multiple nodes distributed among the network. This need is due to the increasing complexity of modern applications that require more computational resources.

To guarantee their high availability at runtime, these systems are becoming highly adaptive and reactive. They need to change their configuration dynamically in order to achieve new