Lecture Notes in Computer Science 9640

More information about this series at http://www.springer.com/series/7408

Rogério de Lemos · David Garlan
Carlo Ghezzi · Holger Giese (Eds.)

Software Engineering for Self-Adaptive Systems III

Assurances

International Seminar
Dagstuhl Castle, Germany, December 15–19, 2013
Revised Selected and Invited Papers

Springer

Editors
Rogério de Lemos
University of Kent
Canterbury
UK

Carlo Ghezzi
Politecnico di Milano
Milan
Italy

David Garlan
Carnegie Mellon University
Pittsburgh, PA
USA

Holger Giese
Hasso Plattner Institute for Software Systems Engineering
Potsdam
Germany

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-319-74182-6 ISBN 978-3-319-74183-3 (eBook)
https://doi.org/10.1007/978-3-319-74183-3

Library of Congress Control Number: 2018930988

LNCS Sublibrary: SL2 – Programming and Software Engineering

Cover illustration: Correspondences between the feedback loop and the MAPE-K Loop. Created by Marin Litoiu et al. Used with permission.

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

An important concern for modern software systems is to become more cost-effective, while being versatile, flexible, resilient, dependable, energy-efficient, customizable, configurable, and self-optimizing when reacting to run-time changes that may occur within the system itself, its environment, or its requirements. One of the most promising approaches to achieving such properties is to equip software systems with self-managing capabilities using self-adaptation mechanisms. Self-adaptive systems are able to adjust their behavior or structure at run-time in response to their perception of the environment and the system itself. Lately, the software engineering community has recognized its key role in enabling the development of self-adaptive systems that are able to adapt to internal faults, changing requirements, and evolving environments. Despite recent advances in this area, one key aspect that remains to be tackled in depth is assurances, i.e., the provision of evidence that the system satisfies its stated functional and non-functional requirements during its operation in the presence of self-adaptation. The provision of assurances for self-adaptive systems is challenging since run-time changes introduce a high degree of uncertainty during their operation.

This book is one of the outcomes of Dagstuhl Seminar 13511 on "Software Engineering for Self-Adaptive Systems: Assurances" held in December 2013. It is the third book of the series on *Software Engineering for Self-Adaptive Systems*, and the papers in this volume address the state of the art in the field of assurances for self-adaptive systems, describe a wide range of approaches coming from different strands of software engineering and control engineering, and posit future challenges facing this field of research. More specifically, this book comprises a research challenges paper, three topical papers dealing in depth with the identified challenges, and invited papers from recognized experts in the field. All the papers in this book have been peer-reviewed, with the exception of the research challenges paper, which was written in several iterations over the past two years by the participants of this Dagstuhl Seminar. The book consists of four parts: "Research Challenges," "Evaluation," "Integration and Coordination," and "Reference Architectures and Platforms."

Part 1 of the book, entitled "Research Challenges," contains four papers including the paper "Software Engineering for Self-Adaptive Systems: Research Challenges on Assurances," and three working group papers that elaborate on the key issues raised on the challenges paper.

The challenges paper summarizes the Dagstuhl Seminar discussions and provides insights on the provision of assurances for self-adaptive software systems. Instead of dealing with a wide range of topics associated with the field, this paper focuses on three fundamental topics of self-adaptation regarding the provision of assurances: perpetual assurances, decomposing assurances, and lessons on assurances from control theory.

The second paper in the first part of this book by Weyns, Bencomo, Calinescu, Cámara, Ghezzi, Grassi, Grunske, Inverardi, Jezequel, Malek, Mirandola, Mori, and Tamburrelli, entitled "Perpetual Assurances in Self-Adaptive Systems," provides a background framework and the foundation for perpetual assurances for self-adaptive systems. The paper also discusses concrete challenges of offering perpetual assurances, requirements for solutions, realization techniques, and mechanisms to make solutions suitable.

The third paper entitled "Challenges in Decomposing and Composing Assurances for Self-Adaptive Systems," authored by Schmerl, Andersson, Vogel, Cohen, Rubira, Brun, Gorla, Zambonelli, and Baresi, discusses how existing assurance techniques can be applied to composing and decomposing assurances for self-adaptive systems, highlights the challenges in applying them, summarizes existing research to address some of these challenges, and identifies gaps and opportunities to be addressed by future research.

The fourth paper "What Can Control Theory Teach Us About Assurances in Self-Adaptive Software Systems?," authored by Litoiu, Shaw, Tamura, Villegas, Müller, Giese, Rouvoy, and Rutten, describes the control theory approach for the provision of assurances, explains several control strategies illustrated with examples from both domains (classical control theory and self-adaptive systems), and shows how the issues addressed by these strategies can and should be considered for the assurance of self-adaptive software systems.

Part 2 of this book, entitled "Evaluation," consists of four papers describing verification and validation techniques that can be used as evidence for the provision of assurances.

The first paper by Sharifloo and Metzger, entitled "MCaaS: Model Checking in the Cloud for Assurances of Adaptive Systems," introduces a cloud-based framework that delivers model checking as a service (MCaaS). MCaaS offloads computationally intensive model checking tasks to the cloud, thereby offering verification capabilities on demand. Self-adaptive systems running on any kind of connected device may take advantage of model checking at run-time by invoking the MCaaS service. As a proof of concept, the authors have implemented and validated their proposed approach for the case of probabilistic model checking.

The second paper, entitled "Analyzing Self-adaptation via Model Checking of Stochastic Games," by Cámara, Garlan, Moreno and Schmerl describes an approach based on model checking of stochastic multiplayer games that enables developers to approximate the behavioral envelope of a self-adaptive system by analyzing best- and worst-case scenarios of alternative designs for self-adaptation mechanisms. Compared with other sources of evidence, such as simulations or prototypes, the proposed approach is purely declarative and hence has the potential of providing developers with a preliminary understanding of adaptation behaviour with less effort, and without the need to have any specific adaptation algorithms or infrastructure in place.

The third paper by Eberhardinger, Anders, Seebach, Siefert, Knapp and Reif, entitled "An Approach for Isolated Testing of Self-organization Algorithms," describes a systematic approach for testing self-organization algorithms. A key feature of the proposed algorithm testing framework is automation since it is rarely possible to cope with the ramified state space manually. The test automation adopts a model-based testing approach, where probabilistic environment profiles are used to derive test cases that are performed and evaluated on isolated self-organization algorithms.

The last paper of this part by Calinescu, Gerasimou, Johnson and Paterson, entitled "Runtime Quantitative Verification — Advances, Applications and Research Challenges," surveys recent advances in the development of efficient run-time quantitative verification techniques, the application of these techniques within multiple domains, and some outstanding research challenges.

Part 3 of the book covers "Integration and Coordination," and includes three papers on approaches for coordinating interactions in self-adaptive software systems.

The first paper in this part is by Křikava, Collet, Rouvoy, and Seinturier, and entitled "Contracts-Based Control Integration into Software Systems." The described approach relies on the principles of design-by-contract to ensure the correctness and robustness of a self-adaptive software system built using feedback control loops. The proposed solution raises the level of abstraction upon which the loops are specified by allowing one to define and automatically verify system-level properties organized in contracts. These contracts are complemented by support for systematic fault handling.

The second paper, entitled "Synthesis of Distributed and Adaptable Coordinators to Enable Goal-Driven Choreography Evolution," by Autili, Inverardi, Perucci, and Tivoli, proposes a method for the automatic synthesis of evolving choreographies in which the coordination software is synthesized in order to enable proxies and control in the interaction between participant services. The ability to evolve the coordination logic in a modular way enables choreography evolution in response to possible changes. A running example in the domain of intelligent transportation systems illustrates the proposed method.

The last paper in this section, entitled "Models for the Consistent Interaction of Adaptations in Self-Adaptive Systems," by Cardozo, Mens, and Clarke, describes existing approaches that allow for the development of self-adaptive systems and management of the behavioral inconsistencies that may appear due to the interaction of adaptations at run-time. Each of these approaches is evaluated with respect to the assurances they provide for the run-time consistency of the system, in the light of dynamic behavior adaptations.

Part 4 of the book contains four papers covering a wide range of issues related to "Reference Architectures and Platforms."

The first paper in this part is by Rutten, Marchand, and Simon, and is entitled "Feedback Control as MAPE-K Loop in Autonomic Computing." This paper surveys the MAPE-K loop from the point of view of control theory techniques, and then discusses continuous and discrete (supervisory) control techniques, and their application, to the feedback control of computing systems. It also proposes detailed interpretations of feedback control loops as MAPE-K loops, and illustrates these interpretations using a variety of case studies.

The second paper, entitled "An Extended Description of MORPH: A Reference Architecture for Configuration and Behavior Self-Adaptation," and authored by Braberman, D'Ippolito, Kramer, Sykes, and Uchitel provides an extended description of a reference architecture that allows for coordinated, yet transparent and independent, adaptation of system configuration and behavior, thus accommodating complex self-adaptation scenarios.

The final paper of this part, entitled "MOSES: A Platform for Experimenting with QoS-driven Self-Adaptation Policies for Service-Oriented Systems," by Cardellini, Casalicchio, Grassi, Iannucci, Lo Presti, and Mirandola describes a software platform supporting QoS-driven adaptation of service-oriented systems. The platform integrates within a unified framework different adaptation mechanisms, enabling a greater flexibility in facing various operating environments, and the possibly conflicting QoS requirements of several concurrent users.

We would like to thank all the authors of the book chapters for their contributions, the participants of the Dagstuhl Seminar 13511 on "Software Engineering for Self-Adaptive Systems: Assurances" for their inspiring participation in moving this field forward, and Alfred Hofmann and his team at Springer for believing in this important project and helping us to publish this book. Last but not least, we deeply appreciate the great efforts of the following expert reviewers who helped us ensure that the contributions are of high quality: J. Andersson, M. Autili, N. Bencomo, R. Calinescu, J. Camara, N. Cardozo, S. Clarke, P. Collet, V. Cortelessa, N. D'Ippolito, C. E. da Silva, A. Filieri, A. Gorla, V. Grassi, K. Johnson, N. Khakpour, F. Krikava, A. Leva, M. Litoiu, S. Mocanu, K. Mens, R. Mirandola, H. Muller, J. Mylopoulos, R. Rouvoy, E. Rutten, B. Schmerl, V. Souza, M. Tivoli, S. Uchitel, N. Villegas, D. Weyns, and several anonymous reviewers.

We hope that this book will prove valuable for both practitioners and researchers involved in the development and deployment of self-adaptive software systems.

January 2017

Rogério de Lemos
David Garlan
Carlo Ghezzi
Holger Giese

Contents

Research Challenges

Evaluation