Lecture Notes in Computer Science 6662

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Radu Calinescu Ethan Jackson (Eds.)

Foundations of Computer Software

Modeling, Development, and Verification of Adaptive Systems

16th Monterey Workshop 2010
Redmond, WA, USA, March 31 – April 2, 2010
Revised Selected Papers

Springer

Volume Editors

Radu Calinescu
Oxford University, Computing Laboratory
Wolfson Building, Parks Road, Oxford OX1 3QD, UK
E-mail: Radu.Calinescu@comlab.ox.ac.uk

Ethan Jackson
Microsoft Research
One Microsoft Way, Redmond, WA 98052-6399, USA
E-mail: ejackson@microsoft.com

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-21291-8 e-ISBN 978-3-642-21292-5
DOI 10.1007/978-3-642-21292-5
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: Applied for

CR Subject Classification (1998): D.2, H.4, H.3, C.2, H.5, D.2.2, C.2.4

LNCS Sublibrary: SL 2 – Programming and Software Engineering

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Message from the General Chairs

The 16^{th} Monterey Workshop was held at the Microsoft Headquarters during March 31 – April 2, 2010 in Redmond, Washington.

A decade into the new millennium, the field of software engineering faces more challenges than ever to evolve with the increasing demands placed on software, as technology expands and becomes even more integrated into every facet of our lives. This pervasiveness increases not only the range of challenges that software applications must face but also the adaptability, flexibility, and robustness they need in order to reliably function in the chaotic jungle of the real world.

Accordingly, the 16^{th} Monterey Workshop investigated an intriguing direction for potential innovation: mechanisms by which organisms cope with harsh, unfavorable, and variable conditions in the natural world. Distillation and formalization of common strategies of complex systems that persevere and function in severely stressful or unexpected situations can aid in the design of systems that must be able to weather similarly chaotic environments.

While these observed strategies provide a potential source of inspiration, unlike nature, software engineers are not limited to trial and error. We can formulate mathematical models and design principles to provide systematic avenues for realizing, analyzing, and improving software reliability. Specialization to meet the particular needs of software development and clever design based on insights gained from natural strategies may eventually surpass the robustness of biological systems. Multiple approaches are explored here, ranging from decentralization-based redundancy and improved verification of flexible systems with many configurations to self-adaptive, self-management, and self-correction capabilities.

You will find interest and inspiration in the work of this gathering of brilliant minds at Microsoft Headquarters. Touring Microsoft Research was a fantastic opportunity. The presentations on various frontier topics are extremely interesting. The papers in this volume represent the most important directions at the workshop, refined in response to workshop discussions and referee comments.

In particular, we note several presentations dealing with the notion of adaptation in software systems. It is amazing how this notion could be specialized and refined in various application domains such as autonomous space systems, reconfiguration of modular robots, adaptation to application design, managements of unpredictable changes in specifications, and certification of reconfiguration. This subtopic was covered in many different ways during the workshop, without prior coordination of the invitees. This outcome is not surprising if we consider that ubiquitous systems are rising, with increasing rates of new requirements, new operating environments, subsystem failures, and hostile activity.

In some contexts adaptation is a necessity, for a variety of reasons. For example, autonomic robots in some space missions have to be self-adaptive because

there is no way to get an answer from earth, due to a 40-minute delay. In the context of reconfigurable modular robots, runtime adaptation is needed because there is no way to pre-compute the potential moves, due to a combinatorial explosion of possible starting states. Variability in modeling languages requires adaptability because the application-specific extensions needed are discovered only when we know the application. Lightweight formal methods supporting reconfiguration in response to varying system loads are needed to derive a system change that will not violate any system constraints, to ensure success of the proposed adaptation before attempting any system modifications. In these and many other cases, details of the required adaptation depend on information that is not available at the time the system is designed.

We very much enjoyed sharing of the advancement of science and technology in the field of software engineering with the research community, following the culture and tradition of the Monterey Workshop, and would like to thank our fantastic Program Committee Chairs Radu Calinescu from Oxford and Ethan Jackson from Microsoft for assembling a fascinating workshop program.

On behalf of the Monterey Workshop Steering Committee, we would like to thank NSF, ONR, AFOSR, ARO, DARPA, and all of our European research sponsors for their support for the Monterey Workshops over the years, and ARO and Microsoft in particular for making this 16^{th} Monterey Workshop possible. Many of the Monterey Workshop topics have subsequently blossomed into major research initiatives and widespread applications with great benefit to all. Here are the topics of Monterey Workshops from the past two decades:

0^{th}: Research Review on Formal Methods in Software Engineering: Concurrent and Real-time Systems, Monterey, California, 1991
1^{st}: Computer-Aided Prototyping: CAPSTAG, Monterey, California, 1992
2^{nd}: Software Slicing, Merging and Integration, Monterey, California, 1993
3^{rd}: Software Evolution, Monterey, California, 1994
4^{th}: Specification-Based Software Architectures, Monterey, California, 1995
5^{th}: Requirements Targeting Software and Systems Engineering, Bernried, Germany, 1997
6^{th}: Engineering Automation for Computer-Based Systems, Monterey, California, 1998
7^{th}: Modeling Software and System Structure in a Fast-Moving, Scenario, Santa Margherita Ligure, Italy, 2000
8^{th}: Engineering Automation for Software-Intensive System Integration, Monterey, California, 2001
9^{th}: Radical Innovations of Software and Systems Engineering in the Future, Venice, Italy, 2002
10^{th}: Software Engineering for Embedded Systems: From Requirements to Implementation, Chicago, Illinois, 2003
11^{th}: Software Engineering Tools: Compatibility and Integration, Vienna, Austria, 2004
12^{th}: Realization of Reliable Systems on Top of Unreliable Networked Platforms, Irvine, California, 2005

13^{th}: Composition of Embedded Systems: Scientific and Industrial Issues, Paris, France, 2006
14^{th}: Innovations for Requirement Analysis: From Stakeholders' Needs to Formal Designs, Monterey, California, 2007
15^{th}: Foundations of Computer Software, Future Trends and Techniques for Development, Budapest, Hungary, 2008
16^{th}: Modeling, Development and Verification of Adaptive Systems, Redmond, Washington, 2010

March 2011

Luqi
Fabrice Kordon

Message from the Program Chairs

The 16^{th} Monterey Workshop was held at Microsoft Research in Redmond, WA, at an exciting turning point in consumer technology. Cloud computing was becoming mainstream facilitated by a combination of advances in virtualization technology and concerns about the costs and environmental impact of maintaining an in-house IT infrastructure. For the first time, sales of smart phones were predicted to overtake sales of laptops. These two trends have since synergized to provide powerful mobile computing on an unprecedented scale.

Similar technological advances have led to a continual increase in the adoption of IT-based solutions in industrial safety-critical and business-critical applications in recent years. We anticipate that cyber-physical systems — systems that integrate computing and physical processes — will become increasingly common over the next decade.

This evolution focuses our attention on a number of key research challenges. How can we ensure information privacy and security? Can data-centers, clouds, and other large-scale distributed systems be made reliable enough to truly depend on them? Can we certify that software performs its intended functions, and can it adapt to withstand unanticipated component failures? During the workshop we listened to presentations by experienced researchers in the modeling, development and verification of adaptive computer systems, and we discussed these challenges from many angles. This proceedings volume gives both an outline of these discussions and an extension of the works presented at the workshop.

We would to thank the participants for their insightful perspectives and lively discussion, which made this volume possible.

We would also like to thank Jim Larus of the Extreme Computing Group (XCG) for his overview of Microsoft's research in cloud computing. Similarly, we thank Desney Tan for presenting his group's research on next-generation user input devices, giving us a glimpse of what might come after the touch screen. Finally, we are grateful to Fabrice and Luqi for organizing the Monterey Workshop series.

March 2011

Radu Calinescu
Ethan Jackson

Table of Contents