

Introduction to the Software Engineering and Formal Methods 2013 special issue

Mario Bravetti^{1,2} · Robert M. Hierons³ · Mercedes G. Merayo⁴

Received: 30 March 2015 / Accepted: 12 April 2015 / Published online: 7 May 2015
© Springer-Verlag Berlin Heidelberg 2015

1 Introduction

We are in the world in which society is increasingly dependent on software, and so, the quality of this software is more important than ever. Unfortunately, the development of high-quality software is becoming increasingly challenging as complexity grows and systems are often concurrent and distributed. The *Software Engineering* and *Formal Methods* communities have developed a range of approaches that help address this problem, but initially there was relatively little interaction between these areas and some saw them as rivals. Thankfully, these attitudes have gradually changed, with the communities accepting that each makes a useful contribution in tackling an important problem.

It is arguable that several factors have helped bring together the Software Engineering and Formal Methods communities. For example, there has been increasing interest in topics such as model-based testing that fall within both areas, and so, there is much more overlap between the communities. Recent years have seen increases in computation power and improvements in solution mechanisms such as SAT/SMT

solvers and model checkers. This has led to verification and static analysis techniques, developed in the formal methods community, scaling to much larger systems and being used by many more software engineers. However, events that bring these communities together have also played a crucial role.

The first edition of the International Conference on Software Engineering and Formal Methods (SEFM) was held in Brisbane, Australia, in September 2003. The purpose of the SEFM conference is to bring together practitioners and researchers from academia, industry and government to advance the state of the art in formal methods, to facilitate their uptake in the software industry and to encourage their integration with practical engineering methods. SEFM is now an important, established annual event.

This special issue contains extended versions of papers from the 11th International Conference on Software Engineering and Formal Methods (SEFM 2013). The conference received 58 submissions. After a careful reviewing process, the Programme Committee accepted 21 regular papers. Authors of seven of these papers were invited to extend their papers, with one declining. After the standard review process, we were able to accept the two papers that appear in this special issue.

✉ Mario Bravetti
bravetti@cs.unibo.it

Robert M. Hierons
rob.hierons@brunel.ac.uk

Mercedes G. Merayo
mgmerayo@fdi.ucm.es

¹ University of Bologna, Bologna, Italy

² Focus INRIA, Sophia-Antipolis, France

³ Department of Computer Science, Brunel University London, London UB8 3PH, UK

⁴ Departamento de Sistemas Informáticos y Computación, Universidad Complutense de Madrid, Madrid, Spain

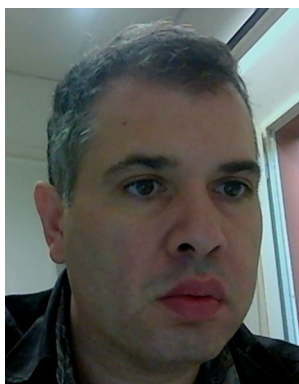
2 In this issue

This issue contains two papers. In “Constraint-based test generation for automotive operating systems”, Yunja Choi and Taejoon Byun propose a methodology for safety checking automotive operating systems based on a novel constraint specification language, called OSEK_CSL. With OSEK_CSL, they specify usage constraints using a predefined set of constraint types identified from the international standard OSEK/VDX. Test sequences with varying degrees

of constraint satisfaction are, then, automatically generated from the set of usage constraints. The authors report the results of a series of experiments that show that their methodology detects safety issues more effectively than conventional testing-based approaches.

The importance of type checking is widely accepted, but typically this does not solve the problem of verifying the correct use of units and dimensions in a model of a software system. In “Inferring Physical Units in Formal Models”, Sebastian Krings and Michael Leuschel present an extension to the B, Event-B and TLA + formal specification languages to support the analysis and partial inference of units of measure for physical quantities. The proposal is mainly based on abstract interpretation combined with classical animation, constraint solving and model checking. This contrasts with the type inference approach of some programming language proposals. The technique has been applied in several case studies and has been implemented as a plug-in for the ProB tool.

Acknowledgements Many people contributed to the success of this special issue. The special issue could not have occurred without the hard work of the authors and the reviewers. We received support from the SEFM Steering Committee in organizing the conference and, in particular, its Chair, Professor Antonio Cerone. We are also grateful to the SOSYM editors for their guidance and patience.



Mario Bravetti (PhD, 2002) is a Associate Professor of Computer Science at the University of Bologna since 2011 and permanent member of the Inria research team FOCUS. He is co-founder and member of the steering committee of the international workshop on Web Services and Formal Methods (WS-FM), currently at its 12th edition. He has been organiser/program committee chair of several international conferences and events,

such as the 20th International conference on Concurrency Theory (CONCUR 2009) and the meeting on the 25th anniversary of Process Algebra. His current research interests include behavioural contracts for service composition.



Robert M. Hierons received a BA in Mathematics (Trinity College, Cambridge) and a Ph.D. in Computer Science (Brunel University). He then joined the Department of Mathematical and Computing Sciences at Goldsmiths College, University of London, before returning to Brunel University in 2000. He was promoted to full Professor in 2003. He is joint editor of the Journal of Software Testing, Verification, and Reliability (STVR) and is on the editorial board of

both The Computer Journal and Formal Aspects of Computing. He has organised or been on the steering committee of several international conferences and workshops. His research largely concerns the automated generation of efficient, systematic test suites on the basis of program code, models or specifications.



Mercedes G. Merayo received her Ph.D. in Computer Science from Universidad Complutense de Madrid, Spain, in 2009. She holds an Associate Professor position in the Computer Systems and Computation Department at the same University. She has published more than 50 papers in refereed journals and international venues. She regularly serves in the Program Committee of conferences such as SEFM, ICTSS or QSR. Dr. Merayo has co-chaired A-QSIC

2011, SEFM 2013 and ICTSS 2014, among others. Her current research interests include model-based testing, distributed testing, asynchronous testing, mutation testing and timed extensions in formal testing.

Software & Systems Modeling is a copyright of Springer, 2017. All Rights Reserved.