Form Methods Syst Des (2010) 37: 93–94
DOI 10.1007/s10703-011-0113-5

Special issue: 2nd World Congress on Formal Methods

Ana Cavalcanti · Dennis Dams

Published online: 20 January 2011

FM 2009, the 16th International Symposium on Formal Methods, marked the 10th anniversary of the First World Congress on Formal Methods that was held in 1999 in Toulouse, France. We celebrated this by organizing FM 2009 as the *2nd World Congress in the FM series*, aiming to once again bring together the formal methods communities from all over the world. The programme committee had members from 46 countries, authors from 38 countries submitted papers, and authors from 23 countries had papers accepted. The five invited speakers represented four continents.

The authors of a selection of the accepted papers were invited to submit extended versions of their papers to special anniversary issues of two reputable journals. This is one of them; the companion anniversary issue is in Formal Aspects of Computing.

The version of the paper *Pointfree Expression and Calculation: From Quantification to Temporal Logic* (by Raymond Boute) submitted to the congress got the Best Paper Award. It presents general techniques for reasoning about temporal logic formulas based on a functional temporal calculus resulting often resulting in shorter proofs. The calculational, algebraic reasoning style also helps to explore expressions in the logic without prior knowledge of the of outcome of the derivation, and without the need for semantic arguments. A number of examples ranging from quantified expressions to valid formulas in Temporal Logic of Actions (TLA) are considered.

Verification (and its automation) is the motivation for all the other papers.

In *Reasoning about Memory Layouts* (by Holger Gast), we have a method for mechanically verifying the layout of data structures and their components in a program's memory space. The data structures include local variables, block pointers, arrays, and C-like data

A. Cavalcanti (✉)
Department of Computer Science, University of York, York YO10 5GH, UK
e-mail: Ana.Cavalcanti@cs.york.ac.uk

D. Dams
Bell Labs Alcatel-Lucent, Copernicuslaan 50, D-6, 2018 Antwerp, Belgium
e-mail: dennis@research.bell-labs.com

structs. It is a new, flexible, and extensible verification framework. The approach centers around a general notion of unfolding data structures, and a novel way of reasoning using these unfoldings. Flexibility is offered by not requiring that memory layouts are fixed, but defined when needed. The well-known (and challenging) Schorr-Waite graph-marking algorithm is tackled in Isabelle/HOL to illustrate the approach.

Doomed Program Points (by Jochen Hoenicke, Rustan Leino, Andreas Podelski, Martin Schaef, and Thomas Wies) defines the notion of a doomed source code fragment to be one whose execution will fail regardless of the start state. Anautomated verification method is developed that can identify doomed program fragments and without reporting false positives. Experiments are then described using a prototype implementation.

In the paper *Scenario-Based Verification of Real-Time Systems Using UPPAAL* (by Kim Larsen, Shuhao Li,Brian Nielsen, and Saulius Pusinskas) an extension of Live Sequence Charts (LSCs) with real-time is proposed. This is used in the automated verification of real-time systems against requirements that are provided as LSCs. The verification is enabled by a reduction to real-time CTL model checking. Two approaches are implemented, one based on the UPPAAL tool and the other using a custom-built tool chain. A number of experiments with both approaches are reported.

Finally, *Model-Based Construction and Verification of Critical Systems Using Composition and Partial Refinement* (by Ralph D. Jeffords, Constance L. Heitmeyer, Myla M. Archer, and Elizabeth I. Leonard) is concerned with fault-tolerant systems. The process is simple, but effective: development starts without consideration of possible occurrence of faults and thus with an ideal model of requirements, then assumptions on faults are added and thus fault-tolerance redundancies are introduced and verified. A case study in avionics illustrates the approach.