

# Editorial

Following this editorial is an obituary for John McCarthy, one of the most influential figures in the early development of artificial intelligence. Sadly, this is the second obituary published in our journal this year. His passing is another big loss to our community.

Next is the paper “Using Fold-in and Fold-out in the Architecture Recovery of Software Systems”, by Michele Risi, Giuseppe Scanniello, and Genoveffa Tortora, which presents an approach based on information retrieval and clustering techniques to automate the architecture recovery process of software systems. This paper accompanies those in volume 24, issue 2 of this journal, which was dedicated to the best papers from the *8th IEEE International Conference on Software Engineering and Formal Methods*. This conference was held in Pisa during 13th–18th September 2010, and the guest editors of the journal were José Luiz Fiadeiro, Stefania Gnesi and Tom Maibaum.

The current issue also contains three papers that were presented at *Formal Methods: Foundations and Applications*, which was the 12th Brazilian Symposium on Formal Methods, SBMF 2009. The symposium was held in Gramado in Brazil, during 19th–21st August 2009. These three papers were selected as the most outstanding to represent the symposium. They all concern principled, formal approaches to testing.

The first paper is “Testing interruptions in reactive systems” by Wilkerson Andrade and Patrícia Machado. Interrupt-driven systems are particularly common on portable devices, where processes have to share resources, such as the screen and hardware communications, as well as having to deal with the arrival of calls from network distributed services. The paper considers the problem of proper test-case selection for interruptions in reactive systems. The method is supported by a tool and the paper illustrates its use with a case study in mobile phone applications.

The second paper is “Concolic Testing of the Multi-sector Read Operation for Flash Memory File System” by Moonzoo Kim, Yunho Kim, and Yunja Choi. Concolic testing is a state-of-the-art testing method that combines concrete dynamic execution with symbolic execution to automatically generate test cases for full path coverage. The authors apply this technique in an empirical case study: the analysis of the multi-sector read operation in the interface to flash-memory management software. Their experiment provides insights into the advantages and weaknesses of the concolic testing approach.

The third paper is “Verifying compiled file system code” by Jan Tobias Mühlberg and Gerald Lüttgen. The authors discuss their retrospective verification of the Linux Virtual File System (VFS). Their objective is to discover violations of API usage rules and memory properties. VFS presents some difficult challenges for software model checkers: it manipulates dynamic data structures with pointers and it is programmed in C and in-lined assembly language. The authors’ solution lies in the use of a novel automated software verification tool, the SOCA Verifier, which symbolically executes and analyses compiled code. SOCA analyses memory accesses, pointer aliasing, and computed jumps, and the authors discuss its suitability as an effective and efficient bug-finding tool to assess operating system kernels.

The final paper in this issue is a regular paper, not part of the Brazilian collection, “Temporal-logic property preservation under Z refinement”, by John Derrick and Graeme Smith.

Jim Woodcock

Published online 26 April 2012

Correspondence and offprint requests to: J. Woodcock, E-mail: jim@cs.york.ac.uk