Electronic Communications of the EASST
Volume 70 (2014)

Proceedings of the 14th International Workshop on Automated Verification of Critical Systems (AVoCS 2014)

The highs and lows of deploying Formal Methods in Industry

Guy H. Broadfoot

1 pages

Guest Editors: Marieke Huisman, Jaco van de Pol
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

The highs and lows of deploying Formal Methods in Industry

Guy H. Broadfoot

Silverdata Ltd.
United Kingdom

Abstract: I attended my first software conference in 1968; it was organised by NATO with the title "The Software Crisis." Many of the papers presented then could have been written yesterday; the problems of the software industry in producing reliable, correct software in the face of increasing complexity and shrinking time to market pressures have not fundamentally changed that much.

In the intervening years as a community we have developed various tactics for trying to minimise software errors. Advances in theorem proving and model checking are good examples of systematic efforts to improve software correctness. Nevertheless, it remains the case that such approaches are rarely if ever encountered in the industrial workplace, with the possible exception of some safety critical domains, such as the software controlling nuclear power plants.

In spite advances in formal methods and supporting tools, the tools available to programmers for verifying assertions about program execution are complex and require knowledge and skills that most practicing programmers do not have. Formal proofs remain difficult to construct, especially for anything but the simplest of programs. Merely constructing assertions to characterise program correctness is a difficult challenge.

In 1998, I conceived the idea of combing model checking, code generation and the specification approach of Sequence-based Specification together to form an integrated software design platform for developing software components whose design (implementation) would be formally verified for correctness with respect to its specification. Other general correctness properties such as freedom from deadlocks, nondeterminism, incomplete cases, etc. would also be verified. Verification would be performed by automatically translating Sequence-based specifications into semantically equivalent CSP process algebra and then applying the model-checking engine FDR2. After verification was completed, semantically equivalent source code would be generated in one of several supported high-level languages.

These ideas were developed further together with Philippa Hopcroft and in 2003 a company was founded to develop a commercial implementation of a development platform based on these ideas. In this talk, I will present an overview of the development platform and the technologies used. I will then discuss the experience gained during 10 years of trying to introduce this approach into industry and the lessons learned along the way.