Softw Syst Model (2008) 7:255–256
DOI 10.1007/s10270-008-0091-2

GUEST EDITORIAL

Software engineering and formal methods

Bernhard Aichernig · Bernhard Beckert

Published online: 11 June 2008

Software engineering and formal methods

Formal methods, i.e., technologies for the formal description, construction, analysis, and validation of software—mostly based on logics and formal reasoning—have matured and can be expected to complement and partly replace traditional software engineering methods in the future. The field has outgrown the area of academic case studies, and industry is showing serious interest.

The challenge now is to scale up the application of formal methods in software industry and to encourage their integration with practical engineering methods. This challenge is met by the conference series Software Engineering and Formal Methods (SEFM) (http://sefm.iist.unu.edu), which aims to bring together practitioners and researchers from academia, industry and government to advance the state of the art in formal methods and their integration in the software development process.

SEFM originated from a workshop held in 2002 in Adelaide, Australia, and became an international conference in 2003 in Brisbane, also in Australia. After SEFM 2004 in Beijing, China, SEFM 2005 was held in Koblenz, Germany, SEFM 2006 took place in Pune, India, and SEFM 2007 in London, UK. SEFM 2008 is scheduled to be held in Cape Town, South Africa.

B. Aichernig (✉)
Institute for Software Technology,
TU Graz, Graz, Austria
e-mail: aichernig@ist.tugraz.at

B. Beckert
Institute for Computer Science, University of Koblenz,
Koblenz, Germany
e-mail: beckert@uni-koblenz.de

Articles in this special section

This special section of Software and Systems Modelling contains articles based on presentations at the Third IEEE International Conference on SEFM (2005). The best papers of the conference were invited to prepare revised and extended versions, and three contributions have been selected for publication:

Savi Maharaj, Thomas Wilson, and Robert Clark: flexible and configurable verification policies with omnibus. Integration of different verification technologies is a key issue for the practical use of verification. The authors describe their work towards the concerted use of verification technologies that are often regarded as alternatives rather than being complementary. They discuss the advantages and limitations of run-time assertion checking, extended static checking, and full formal verification, when applied to the development of object-oriented component-based systems, and they propose guidelines for the successful combination of these technologies. The earlier version of this paper that was presented at SEFM 2005 won the conference's best paper award.

Antonio Cerone, Simon Connelly, and Peter Lindsay: formal analysis of human operator behavioural patterns in interactive surveillance systems. This article proposes a new approach to understanding the causes of human error and to model their occurrence, which is important for human reliability assessment in interactive systems. The authors use the process algebra CSP to model human behaviour. The environment is also modelled as a CSP process, and model checking is used to analyse the overall model. As a case study, a formal model of an air–traffic-control system operator's task is presented and analysed.

James Welch, David Faitelson, Jim Davies: automatic maintenance of association invariants. Invariants are an important concept in the specification and analysis of software systems. The authors present an approach to maintaining association invariants in object-oriented program development based on Booster, a declarative object-modelling language. Association invariants cause difficulties in the development process because they concern more than one class at the time, thus breaking with class modularity. The authors describe how association invariants may be specified and used to extend a method's pre- and post-conditions in an automatic way, based on a set of model transformation rules.

Acknowledgments We thank the authors of the papers submitted to this special section, and we are grateful to the reviewers for helping to select the contributions. We also appreciate the support of the programme committee and the organisers of SEFM 2005. We thank the editors of SoSyM for publishing this special section.