

## Guest editorial to the special section on SEFM 2009

Padmanabhan Krishnan · Dang Van Hung ·  
Antonio Cerone

Published online: 3 March 2012  
© Springer-Verlag 2012

### 1 Introduction

Software engineering is about developing large systems using systematic techniques. Formal methods in software engineering advocates the use of precisely defined techniques to develop software. In theory, requirements can be captured as an abstract specification and the final system obtained via suitable transformations. In practice this is not really feasible. Hence, only a small part of the actual system development process uses formal methods. Thus, there are numerous open research problems in the area of formal methods as applied to software engineering.

Formal methods in software engineering can be viewed as the use of techniques that are based on mathematics (i.e., the formal aspects) in the context of practical principles applied to building software (i.e., the engineering aspects). Topics that are of interest include the use of appropriate languages/notations for the specification of systems, the refinement of specifications to code, the semantic foundations that support formal methods, the rigorous analysis of specifications and/or code that can be undertaken.

The aim of this special section is to provide a collection of papers that illustrate the application of formal methods to

software engineering. The selected papers are an indication of the breadth of current research interests and remaining challenges in this field.

Formal methods and software engineering is now part of many curricula especially at the postgraduate level. We hope that this special section appeals to both established researchers as well as postgraduate students.

### 2 Scope and selection of papers

The Seventh IEEE International Conference on Software Engineering and Formal Methods was held on 23–27 November 2009, in Hanoi, Vietnam. It was sponsored by the IEEE Computer Society and supported by UNU-IIST, Macao, The Vietnam National University and the Japan Advanced Institute of Science and Technology. The aim of the conference was to bring practitioners and researchers together to encourage the integration of software engineering techniques with formal methods.

From the 27 papers that were published at the Conference, we invited 6 papers to submit an extended version for this section. This was based on the reviewers' scores and comments, the evaluation of their presentation at the conference as well as the relevance to the main theme. All submissions underwent the usual peer review process. All papers were reviewed by at least three experts in the field. The papers also underwent two rounds of reviews before they were finally accepted.

At the end of the review process the following four papers were selected.

The paper titled “Event-B Patterns and Their Tool Support” by Thai Son Hoang, Andreas Fürst and Jean-Raymond Abrial adapts the idea of design patterns to the Event-B specification language. This enables the reuse of existing models as well as their correctness proofs. Case studies along with

---

P. Krishnan (✉)  
School of Information Technology, Bond University, Gold Coast,  
QLD 4229, Australia  
e-mail: pkrishna@staff.bond.edu.au

D. V. Hung  
University of Engineering and Technology  
144 Xuan Thuy, Cau Glay, Hanoi, Vietnam  
e-mail: dvh@vnu.edu.vn

A. Cerone  
International Institute for Software Technology,  
United Nations University, P. O. Box 3058, Macau SAR, China  
e-mail: antonio@iist.unu.edu

tool support are used to demonstrate the practical applicability of this approach.

The paper titled “Metric Propositional Neighborhood Logics on Natural Numbers” by Guido Sciavicco, Davide Bresolin, Dario Della Monica, Angelo Montanari and Valentin Goranko represents a theoretical paper that develops a temporal logic over intervals that is decidable. This logic is more expressive than other relevant variants and is motivated by various standard applications in the literature.

The paper titled “Behavioral Specification of Reactive Systems Using Stream-Based I/O Tables” by Benjamin Hummel and Judith Thyssen presents a technique to specify reactive systems using signal patterns. The authors argue, using a large case study, that specifying the patterns as a table simplifies the creation of a formal specification from informal requirements. They also show that many properties can be automatically checked.

The paper titled “Relational interprocedural verification of concurrent programs” by Bertrand Jeannet develops a technique to analyse reachability in the context of recursive procedure calls and concurrency. The author develops the necessary theory to model the concurrent stack and suitable abstraction techniques. In addition to the theory, the

author describes a number of case studies that demonstrates the applicability of the theory.

These four papers show the wide spectrum of interest in formal methods as applied to software engineering. The special section covers foundational work, specification and verification techniques and tool support with emphasis on the application of the theory in practice.

We would like to thank the contributing authors for their efforts in extending the conference papers. We are grateful to the reviewers for their high quality work in reviewing the papers and providing advice that improved both the content and presentation of all the submissions. The reviewers who assisted us are: Jonathan Bowen, Antonio Cau, Wei-Ngan Chin, Dimitar Guelev, Bertrand Jeannet, Aditya Kanade, Uday Khedker, Bakh Khossainov, Xuandong Li, Dominique Mery, Mizuhito Ogawa, Martin Steffen, Phil Stocks, Hoang Truong, R. Venkatesh, Kostyantyn Vorobyov and Xu Wang.

Sincere thanks also to Martin Schindler and the editorial team at SoSyM for their fantastic and sustained support and encouragement.