Preface

Selected papers on Integrated Formal Methods (iFM09)

This issue contains extended versions of selected papers from the 7th International Conference on Integrated Formal Methods, iFM09, held on 16–19 February 2009 in Düsseldorf, Germany. The Integrated Formal Methods conference series seeks to further research into the combination of different formal methods, both for modeling and analysis, covering all aspects from language design over verification techniques to tools and their integration into software engineering practice. For the 2009 conference, 21 papers were accepted by the program committee. Of these, six were selected for submission to the special issue, and after some extensive reviewing all six have finally been accepted.

The papers take a look at the area of integrated formal methods from different perspectives. The article by Steve Schneider and Helen Treharne on Changing System Interfaces Consistently: A New Refinement Strategy for CSP can be seen as a “classical” iFM paper, discussing the integration of the formal methods B and CSP, and the question of refinement when the granularity of operations changes.

An integration of predicate abstraction and deductive verification is introduced in the article by Benjamin Weiβ on Predicate Abstraction in a Program Logic Calculus. The aim of his work is the automatic derivation of loop invariants which are needed for deductive program verification and usually have to be supplied manually.

Nikola Benes, Lubos Brim, Barbora Buhnova, Ivana Cerna, Jiri Sochor and Pavлина Varekova describe the application of partial order reduction techniques to systems in which the requirements to be analyzed are given in a logic integrating atomic propositions on states and on events (Partial Order Reduction for State/Event LTL with Application to Component-Interaction Automata).

In the article A Semantics for Behavior Trees using CSP with Specification Commands, Robert Colvin and Ian Hayes provide a formal semantics to the language called “Behavior Trees”. This language is used to conveniently describe informal requirements. The formal semantics of Behavior Trees is given in terms of an extended version of the process algebra CSP with hierarchical states, which itself is formalized via an operational semantics.

Behavioral subtyping has been introduced to allow for modular verification of object-oriented programs, by restricting the use of inheritance. In the article Incremental Reasoning with Lazy Behavioral Subtyping for Multiple Inheritance, Johan Dovland, Einar Broch Johnsen, Olaf Owe, and Martin Steffen propose a more liberal approach, which is particularly well suited to multiple inheritance.

A language for specifying behavioral properties is introduced in the article Property Specifications for Workflow Modelling by Peter Wong and Jeremy Gibbons. This language is then translated into a fragment of linear temporal logic, which itself can be automatically translated to CSP for refinement checking. The approach is illustrated on a case study.

We are grateful to all the members of the iFM09 program committee, and in particular to the reviewers of these selected papers, some of which had not been involved in iFM09. For the preparation of this special issue, we would moreover like to thank Jan Bergstra, the editor in chief of Science of Computer Programming, and Bas van Vlijmen, the editorial manager of SCP, for their ever prompt help with all sorts of questions.

Guest editors
Program committee chairs iFM09
Michael Leuschel*
Universität Düsseldorf, Institut für Informatik,
Universitätsstr. 1, D-40225 Düsseldorf, Germany
E-mail address: leuschel@cs.uni-duesseldorf.de.

0167-6423/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.scico.2011.01.009
Heike Wehrheim
Universität Paderborn,
Fakultät für Elektrotechnik, Informatik und Mathematik,
Warburger Street 100, D-33098 Paderborn, Germany
E-mail address: wehrheim@mail.uni-paderborn.de.

Available online 4 March 2011

* Corresponding editor.