Special Issue on Formal Aspects of Component Software (Selected Papers from FACS'12)

Corina Pasareanu, Gwen Salaün

Preface: Special Issue on Formal Aspects of Component Software (Selected Papers from FACS'12)

This issue contains extended versions of selected papers from the 9th International Symposium on Formal Aspects of Component Software (FACS'12).

The FACS symposium series addresses formal methods in the context of component-based and service-oriented software development. Formal methods provide a foundation for component-based software by successfully addressing challenging issues such as mathematical models for components, composition and adaptation, or rigorous approaches to verification, deployment, testing, and certification.

FACS'12 was the 9th symposium in this series, and was held in Mountain View (USA) on September 12-14, 2012. For FACS'12, we received 40 submissions. After the review process, the international Program Committee decided to select 16 papers for presentation during the symposium and inclusion in the FACS'12 proceedings. From these 16 papers, the authors of six best papers were invited to submit an extended version to this special issue. These extended papers went through a rigorous peer review process. The revised versions of five papers were finally accepted and are included in this special issue. The papers included here provide key insights on different formal aspects of component software, covering topics ranging from real-time and communicating systems, interface theories, probabilistic verification assume-guarantee reasoning.

The first article in this special issue, *"Formal Patterns for Multirate Distributed Real-Time Systems"*, by K. Bae *et al.*, proposes multirate PALS (Physically Asynchronous, Logically Synchronous) as a formalized mathematical model providing a formal pattern that can drastically reduce the complexity of designing, verifying, and implementing multirate Distributed Real-Time Systems that must achieve virtual synchrony in an asynchronous setting. In particular, the authors prove that the entire DRTS design as a concurrent system of asynchronous components communicating in a network is bisimilar to a simpler synchronous multirate ensemble of state machines. This bisimilarity induces a significant reduction on the number of states, making model checking possible in many cases where it is unfeasible in the original DRTS setting. Multirate PALS is supported by Real-Time Maude for specification and model checking purposes, and is illustrated with a multirate hierarchical control system.

The second article, *"Avoiding Diamonds in Desynchronisation"*, by H. Beohar *et al.*, presents sufficient and necessary conditions under which a synchronous design is equivalent to an asynchronous one and formally proves that the so-called diamond property is no longer needed for desynchronisation when half-duplex queues are used as a communication buffer. These theoretical results are illustrated for desynchronising the synchronous systems that are synthesised using supervisory control theory.

The third article in this special issue, *"A Meta-Theory for Component Interfaces with Contracts and Ports"*, by S. Bauer *et al.*, presents a generic framework to construct a theory of component interfaces with port contracts on top of any

arbitrary labeled interface theory. The authors study reliable component interfaces and provide methodological guidelines how to design reliable interfaces and how to adapt them to changing environments. The approach was illustrated with two instantiations. First, the authors consider modal component interfaces such that component behaviors and the assume and guarantee behaviors of ports are given in terms of modal I/O (input-output)-transition systems with weak notions of refinement and compatibility. The second instance uses I/O-predicates as interface specifications.

The fourth article, *"Symbolic Counterexample Generation for Large Discrete-Time Markov Chains"*, by N. Jansen *et al.*, presents several symbolic counterexample generation algorithms for Discrete-Time Markov Chains (DTMCs) violating properties written in PCTL. A counterexample is a symbolic representation of a sub-DTMC that is incrementally generated. First, the authors extend bounded model checking and develop a simple heuristic to generate highly probable paths first. Second, they complement the SAT-based approach by a symbolic BDD-based technique. The experimental results show a substantially better scalability than existing explicit techniques. In particular, the BDD-based approach using a method called fragment search allows for counterexample generation for DTMCs with billions of states.

The fifth article, *"Compositional Assume-Guarantee Reasoning for Input/Output Component Theories"*, by C. Chilton *et al.*, introduces a sound and complete assume-guarantee framework for reasoning compositionally about components modelled as a variant of interface automata. A component is specified by finite traces and expresses both safety and progress properties of input and output interactions with the environment. The framework supports dynamic reasoning about components and specifications, and includes rules for parallel composition, logical conjunction and disjunction corresponding to independent development, and quotient for incremental synthesis. The framework is illustrated through a link layer protocol case study.

Many people have contributed to this special issue, without whose effort this special issue would not have been possible. Besides the authors of the papers, we would like to thank both the members of the Program Committee of the symposium and the additional reviewers who kindly agreed to help us with the reviewing of the papers in this special issue. All carried out an excellent job during this demanding process: Erika Abraham (RWTH Aachen University, Germany), Farhad Arbab (CWI and Leiden University, The Netherlands), Christian Attiogbé (University of Nantes, France), Christel Baier (Technical University of Dresden, Germany), Luís Barbosa (University of Minho, Portugal), Frank de Boer (CWI, The Netherlands), Roberto Bruni (University of Pisa, Italy), Carlos Canal (University of Málaga, Spain), José Luiz Fiadeiro (University of Leicester, UK), Carlo Ghezzi (Politecnico di Milano, Italy), Rolf Hennicker (Ludwig-Maximilians-Universität Munich, Germany), Sascha Klüppelholz (Dresden University of Technology, Germany), Stefan Leue (University of Konstanz, Germany), Michael Lienhardt (University of Paris Diderot, France), Zhiming Liu (Birmingham City University, UK), Markus Lumpe (Swinburne University of Technology, Australia), Eric Madelaine (Inria, Centre Sophia Antipolis, France),

Sun Meng (Peking University, China), John Mullins (Polytechnical School of Montreal, Canada), Peter Olveczky (University of Oslo, Norway), Meriem Ouederni (Toulouse INP, France), Frantisek Plasil (Charles University, Czech Republic), Pascal Poizat (Université Paris Ouest Nanterre La Défense, France), José Proença (KU Leuven, Belgium), Shaz Qadeer (Microsoft, USA), John Rushby (SRI International, USA), Bernhard Schätz (fortiss GmbH, Germany), Nishant Sinha (NEC Labs, Princeton, USA), Marjan Sirjani (Reykjavik University, Iceland), Volker Stolz (University of Oslo, Norway), Carolyn Talcott (SRI International, USA), Oksana Tkachuk (NASA Ames, USA), Sebastian Uchitel (University of Buenos Aires, Argentina), James Worrell (University of Oxford, UK), Lina Ye (Inria, France), Gianluigi Zavattaro (University of Bologna, Italy).

Corina Pasareanu
NASA Ames, USA

Gwen Salaün
Grenoble INP, Inria, France

Science of Computer Programming Guest Editors