Guest Editorial: Special Issue on Models and Methodologies for System Design

This special issue is based on innovative ideas presented and discussed during the 12th ACM/IEEE International Conference on Formal Methods and Models for System Design (MEMOCODE'14) held at EPFL in Lausanne, Switzerland, on October 19–21, 2014. Selected papers from the conference were invited for this special issue together with an open call soliciting novel contributions on the topics of this conference. Rigorous reviews of 12 submissions led to the selection of four articles for this special issue. In this editorial statement, we outline the premise and the context of this special issue, give a short introduction to the theme under consideration, and briefly introduce the articles selected.

MEMOCODE is a top venue for researchers working on formal methods and models for embedded hardware and software design (codesign). In past editions of the conference, MEMOCODE emphasized codesign as its primary focus, but over the last decade, the clear boundaries between system components implemented in hardware, firmware, software, middleware, or applications have blurred. This evolution in system design practices has triggered a change in the title of the conference to better reflect the current needs of industry and most relevant research directions. MEMOCODE's main goal is to bring together researchers in software design, hardware design, as well as hardware/software codesign, and exchange ideas, research results, and lessons learned and apply them to each other's areas. We want to emphasize the importance of models and methodologies in correct system design and provide an exchange platform for researchers and industry practitioners who work in any or all components of the system stack—hardware, firmware, middleware, software, architecture, and applications. Contributions to MEMOCODE address all aspects of methods and models for hardware and embedded software design: formal foundations, informal engineering methodologies with sound basis, model-driven approaches, design tools, design case studies, and industry-scale experimental case studies.

The first article of this special issue is entitled "On Memory Reuse Between Inputs and Outputs of Dataflow Actors." It is authored by Karol Desnos, Maxime Pelcat, Jean-François Nezan, and Slaheddine Aridhi. It introduces a new technique to minimize the memory footprints of digital signal processing (DSP) applications specified with synchronous dataflow (SDF) graphs and implemented on shared-memory multiprocessor systems-on-chips. In addition to the SDF specification, which captures data dependencies between coarse-grained tasks called actors, the proposed technique relies on two optional inputs abstracting the internal data dependencies of actors: annotations of the ports of actors and script-based specifications of merging opportunities between input and output buffers of actors. Experimental results on a set of applications show a reduction of the memory footprint by 48% compared to state-of-the-art minimization techniques.

The second article is entitled "Contract-Based Requirement Modularization via Synthesis of Correct Decompositions" and is authored by Thi Thieu Hoa Le, Roberto Passerone, Uli Fahrenberg, and Axel Legay. In the development of distributed systems,

1539-9087/2016/02-ART29 $15.00
DOI: http://dx.doi.org/10.1145/2885503

contract-based design can play a vital role in ensuring interoperability of components and adherence to specifications. The article proposes two strategies enabling designers to synthesize or refine a set of contracts so their composition satisfies an additional contract. Such a methodology is desirable to verify the satisfaction of an overall system property represented as a contract, given the satisfaction of component properties represented as contracts. If the verification result is negative, then the designer must face the issue of refining the subproperties and components. The article develops a generic algebraic method to that end and shows how it can be applied in different contract models to support top-down component-based development of distributed systems.

The third article is entitled “Designing Parameterizable Hardware IPs in a Model-Based Design Environment for High-Level Synthesis” and is presented by Shahzad Ahmad Butt, Mehdi Roozmeh, and Luciano Lavagno. This article proposes a method to model DSP algorithms as flexible intellectual property blocks within popular model-based design environments such as Simulink. These IP blocks are written in C and designed to support both functional simulation and hardware implementation. They allow architectural design space exploration and hardware implementation through high-level synthesis. An advantage of this modeling approach is to use the same bit-accurate model for both simulation and high-level synthesis. The proposed methodology is exemplified by the model of a fast Fourier transform algorithm that is synthesized for several DSP applications with different performance and cost requirements.

The fourth and last article of the special issue is entitled “A Rigorous Approach for System-level Performance Modeling and Analysis,” by Ayoub Nouri, Marius Bozga, Anca Molnos, Axel Legay, and Saddek Bensalem. It presents a systematic approach for building stochastic performance abstractions of system-level models using statistical inference and model calibration. The article proposes a scalable statistical model-checking technique to evaluate performance for such models.

Paolo Ienne
Jean-Pierre Talpin

Guest Editors