ACM Transactions on Embedded Computing Systems
Special Issue on Methods and Models for System Design

This special issue is dedicated to the research areas of the 12th ACM-IEEE International Conference on Methods and Models for System Design (MEMOCODE'14) held in Lausanne, Switzerland, in October 2014.

In the past, MEMOCODE emphasized co-design 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 necessitated a change in the title of the conference to cater to the needs of today's industry and research practices. MEMOCODE's main agenda is to bring together researchers in software design, hardware design, as well as hardware/software co-design, and exchange ideas, research results, lessons learned from each other and apply them to each other's areas. We want to emphasize the importance of models and methodologies in correct system design, and provide a platform for researchers and industry practitioners who work in any or all components of the system stack—hardware, firmware, middleware, software, architecture and applications.

Likewise, this special issue seeks outstanding research contributions addressing all aspects of methods and models for hardware and embedded software design. We are interested in formal foundations, informal engineering methodologies with sound basis, model driven approaches, design tools, design case studies and industry-scale experimental case-studies. Research areas of interest to MEMOCODE consist of (but not limited to) the following topics:

Programming Models, Languages, Methodologies and Tools - System Modeling Languages - Architecture Description Languages - Domain Specific Languages - Generic Programming.

Synchronous Programming Models and Languages - Reactive, Streaming, Concurrent Programming Languages - Program Synthesis Techniques, Tools, Methods - Correct-by-Construction Methodologies - Higher Level Hardware Description Languages.

Analysis, Verification and Test - Static and Dynamic Analysis - Symbolic Simulation - Model Checking - Type Theoretical Program Analysis - Abstract Interpretation - Test-cases Generation - Coverage Metrics and Techniques - Tools and Methodologies for Verification, Analysis - Performance Analysis - Formal Models such Petri-Nets, Timed Automata, Transition Systems, etc.

Refinement, Component, Platform-based Methodologies - Stepwise refinement methodologies - Refinement-based correct-by-construction Design - Component based design - Component composition languages and environments - Platform-based design - Refinement proof techniques, simulations and other relations - Reusability methodologies - Contract-based Component Design - Assume/Guarantee Reasoning - Reverse Engineering of Systems to build Platform

and Component Models - Separation of Concern based Design - Aspect-oriented Design Approaches.

Models of Time in System Design - Synchronous, polychronous, asynchronous concurrency models - Latency Insensitive Design - Globally Asynchronous and Locally Synchronous Design Methods - Locally Asynchronous and Globally Synchronous Design Methods - Real-Time Models, Scheduling, Proofs of Real-Time Guarantees.

Fault Tolerance, Fault Models, Reliability, and Resilience - Fault Tolerant System Design - Fault-tree and other techniques for reasoning about Faults - Defect Tolerant System Design - Reliability Models, Risk Models, Probabilistic Computation - Resilient System Design - Run-Time Adaptable Systems - Run-Time patching and maintenance.

Quantitative/Qualitative Reasoning - Power Models and Power/Performance/cost/latency trade-off methods - Reasoning techniques, data mining and other analytical methods for predicting power/performance etc - System Level Models for Quantitative exploration of Design Space.

Secure and Trustworthy Design - Methods and Models for Cyber Security - Reasoning techniques for Cyber Security, Vulnerability, Counter measures - Risk Modeling, Performance/Security Trade-offs - Security Specification languages, Formal tools, Testing.

Case Studies & Tools Paper - Design case studies based on sound methodological precepts - Industry Strength Case studies - Tools paper describing the architecture, design, capabilities of tools with experimental evaluation.

GUEST EDITORS

Paolo Ienne, EPFL
Jean-Pierre Talpin, INRIA, Associate Editor
Any questions on this special issue should be addressed to Jean-Pierre Talpin at jean-pierre.talpin@inria.fr

TIMETABLE

Submissions due March 11th, 2015
Preliminary notification on June 10th, 2015
Final papers due August 10th, 2015
Likely Publication in the fall 2015

Note: If a submitted paper is recommended by the guest editors for “major revision”, the paper will be put through the reviewing process when revised, but not guaranteed to be included in the special issue for which it was submitted. However, at the conclusion of the review process if the paper is accepted, it will then be included in a regular issue of the journal. This happens often due the time bounded nature of a special issue production process, and papers requiring multiple rounds of reviewing may not have a final decision within the stipulated time frame of the special issue.