Under consideration for publication in Formal Aspects of Computing

Editorial

Manfred Broy[1], Gerald Lüttgen[2] and Michael Mendler[3]

[1]Department of Informatics, Munich University of Technology, Germany
[2]Department of Computer Science, The University of York, UK
[3]Informatics Theory Group, The University of Bamberg, Germany

This is the second special issue of the Formal Aspects of Computing journal on the topic of *Semantic Foundations of Engineering Design Languages*, the first of which recently appeared as number 4 of volume 15. The response to the original call for contributions publicised in the summer of 2002 has been overwhelming both in number and in the high quality of the submissions received. This has not only testified to the importance and timeliness of this topic, but has also made it clear that the breadth of current scientific activities in this area warrants representation within a double special issue.

The six contributions included in this second special issue explore several interesting facets of integrating theoretical and practical aspects of formal methods, as they meet in the semantics of specification and design languages for embedded systems. The articles are naturally structured here into two tracks:

- **Semantic Models.** This first track presents a broad set of semantic models applied in the specification and analysis of state–of–the–art engineering design languages. M. Gallardo, P. Merino and E. Pimentel present a generalised transition system semantics of Spin's Promela language that is parameterised in abstract data domains with the aim to support model checking by abstraction and refinement. I. Krüger, W. Prenninger and R. Sandner investigate an extension of *Message Sequence Charts* by a mechanism of broadcasting messages based on semantic predicates over the domain of streams. Using an extended form of interface automata, E.A. Lee and Y. Xiong develop a behavioural type system for the static analysis of the compatibility of interaction protocols and scheduling schemes in the component–based programming framework of Ptolemy II.
- **Combining Languages.** This second track explores proposals of unifying different language paradigms. N. López, M. Núñez and F. Rubio present an integrated framework of a stochastic process algebra with a functional programming language exploiting synergies between them for both the implementation and performance analysis of asynchronously communicating stochastic processes. M. Nebut gives a systematic analysis of the role of clocks for the specification of signal presences and absences in several synchronous programming languages (Lustre, Lucid Synchrone, Esterel and Signal), and proposes a unifying first–order clock language that supports signals with values from non–Boolean data domains. S. Tini shows that the framework of Timed Concurrent Constraint Programming lets one embed, compositionally and elegantly, the synchronous languages Argos and Lustre, which brings out clearly the essential declarative nature of the synchrony hypothesis and the connections between synchronous and logic programming.

We are grateful to all the authors for their excellent contributions, to the many reviewers for their careful and responsible refereeing, and finally to the editor–in–chief Prof. C.B. Jones and to the associate editor Dr. D.J. Cooke for making this second special issue possible.

Correspondence and offprint requests to: gerald.luettgen@cs.york.ac.uk