Preface

The first International Symposium on Formal Methods for Components and Objects (FMCO) was held in Leiden, The Netherlands, from November 5 to 8, 2002. The program consisted of invited presentations given by leading experts in the fields of Theoretical Computer Science and Software Engineering.

After the symposium, the speakers were invited to contribute to a combined special issue of the two sister journals *Theoretical Computer Science* and *Science of Computer Programming* that provides an in-depth overview of the state of the art of formal methods for components and objects.

This special issue focuses on the pragmatic aspects of the application of formal methods to component-based and object-oriented systems, whereas the other special issue focuses on the theoretical foundations.

The organization of FMCO has been carried out in the context of the NWO/DFG bilateral project Mobi-J and of the European IST project Omega (2001-33522). In particular, we acknowledge the NWO funding of Mobi-J, without which this symposium could not have been organized. We appreciate the cooperation with Willem-Paul de Roever and Susanne Graf in the organization of the FMCO symposium.

This special issue opens with an article by F. Arbab introducing the notion of Abstract Behavior Type as a model for both components and their composition. He shows the applicability of Abstract Behavior Types through a number of examples expressed in REO, a new exogenous coordination language for compositional construction of component connectors based on a calculus of channels.

B. Jacobs and his coauthors C.-B. Breunessen, N. Cataño, and M. Huisman present a case study in formal specification and verification of a smart card application. The application is an electronic purse implementation annotated with specifications using the Java Modeling Language.

W. Damm and his coauthors B. Josko, A. Pnueli, and A. Votintseva in their article characterize a subset of UML suitable for modeling Safety-Critical Applications, and provide an operational semantics that can be used for the formal verification of temporal properties with existing model-checking tools.

This verification problem is addressed in the next article by W. Damm and B. Westphal. As specification language, the authors propose an extension of Live Sequence Charts with dynamically bound instance lines, and as verification techniques they use model checking.
in combination with abstract interpretation to construct a finite-state approximation of the original model.

J. Sifakis and his coauthor G. Gössler propose a framework for component-based modeling using an abstract layered model for components. They define an associative and commutative composition operator on components encompassing heterogeneous interaction, that lead to a concept of “flexible” composition.

The Java Modeling Language has been introduced by G. Leavens, who discusses in this issue, together with his coauthors Y. Cheon, C. Clifton, C. Ruby, and D. Cok, its use as a common notation for both formal verification and runtime assertion checking.

R. Leino and his coauthors T. Millstein and J. Saxe in their article describe a new method for reconstructing, from the output of a theorem prover, error traces that lead to the program errors discovered by the theorem prover.

E.-R. Olderog and H. Wehrheim explain CSP-OZ, a combination of Communicating Sequential Processes (CSP) and Object-Z (OZ) that enables the specification of systems having both a state-based and a behavior-oriented view using the object-oriented concepts of classes, instantiation and inheritance.

In the final article, K. Sere and her coauthors J. Plosila and M. Waldén propose a method to synthesize a set of components from a high-level specification of the intended behavior of the target system. They use Action Systems as a formal approach to system design. Their methodology is inspired by hardware-oriented approaches, but they succeed in showing that their approach is more general.

We gratefully acknowledge the referees for the high quality of their reviews.