

## **Behavior of Composed Concurrent Systems: Logic and Reasoning**

### **Preface**

---

This special issue contains the extended versions of the selected papers presented at the first IPM International Workshop on Foundations of Software Engineering (Theory and Practice), Tehran, Iran, October 1-3, 2005. This event, FSEN05, was organized by the School of Computer Science at the Institute for Studies in Fundamental Sciences (IPM) in Iran, in cooperation with the ACM SIGSOFT.

The general theme of this special issue is composition of concurrent systems, and how to reason about them.

Mark-Oliver Stehr develops a UNITY-style temporal logic for labeled transition systems in the calculus of inductive constructions and presents a general proof rule for compositional verification of liveness assertions in tightly coupled systems. This approach is closely related to classical work on interference-free proofs for parallel programs.

Einar Broch Johnsen, Olaf Owe, and Arild B. Torjusen validate components with respect to their behavioral interfaces. Behavioral interfaces specify semantic requirements on the observable behavior of components, expressed in an assume-guarantee style. Reasoning in this approach is supported by a rewriting logic model that is transparently extended with the history of all observable communications.

Dave Clarke proposes a semantics for the coordination language Reo in the presence of dynamic reconfiguration. He presents a basic logic together with its model checking algorithm for reasoning about connector reconfiguration.

Erika Abraham, Frank S. de Boer, Willem-Paul de Roever, and Martin Steffen introduce an assertional proof method to reason about safety-properties for a small concurrent sub-language of Java. This subset of Java covers concurrency and particularly exception handling. They show the soundness and the relative completeness of their proof method.

Shahram Esmailsabzali, Nancy A. Day, and Farhad Mavaddat extend interface automata with complex actions. As atomic constructs, complex actions encapsulate and limit interleaving to allow disciplined description of system behavior. They provide proofs of various properties of their extended version of interface automata and discuss its application for modeling of Web services.

*Farhad Arbab — Leiden University, The Netherlands*

*Marjan Sirjani — Tehran University, Iran*

Special Issue Editors