Preface

In the last decade *software architecture (SA)* has emerged as a central notion in the software development of large complex systems. SA main feature resides in being a manageable but meaningful abstraction of the system under development. In the software life cycle, SA is the first system representation that encompasses a description of both the system structure and the system behavior. The static, structural view describes how the system is made up of interconnected subsystems called components. The dynamic, behavioral view specifies how these components interact at execution time to achieve the system's goals. SA abstractions allow for the early validation of basic design choices with respect to both functional and nonfunctional requirements. Moreover SA plays a significant role throughout the software development life cycle, from requirements analysis and validation, to design, and down to code and execution level. Due to the importance of the SA level of abstraction, in the past years there has been much research activity in the use of formal methods in SA for analysis and validation purposes.

This volume collects the set of papers accompanying the lectures given at the third *International School on Formal Methods for the Design of Computer, Communication and Software Systems (SFM).* The school addresses the use of formal methods in computer science as a prominent approach to the rigorous design of computer, communication and software systems. The main aim of the SFM series is to offer a good spectrum of current research in foundations as well as applications of formal methods, which can be of help for graduate students and young researchers intending to approach the field.

SFM 2003:SA is devoted to software architectures and covers architectural description languages and tools as well as architecture-level analysis and synthesis techniques for an architecture-centric software development process that encompasses system requirements, design, testing, and evolution. The school also includes lectures concerned with mobility, performance, and dependability at the architectural level of design. The ordering of the papers in this volume follows the order of the lectures and reflects the software life cycle.

The opening paper by David Garlan introduces the problem of formally describing SA, which is preliminary to any SA investigation-based activity. Axel van Lamsweerde addresses the problem of bridging requirements and SA and proposes a goal-oriented approach to architectural design based on a framework for modeling, specifying, and analyzing requirements. Jeff Kramer, Jeff Magee, and Sebastian Uchitel's paper outlines their tool-supported approach to the design and analysis of complex systems at the architectural level, which can also be used as an initial basis for validating requirements. The analysis of SAs in themselves, as independent software artifacts, is the theme of Alex Wolf, Judy Stafford, and Mauro Caporuscio's paper, which illustrates the application of dependence analysis to architectural descriptions of software systems.

At the design level, component-based design techniques are gaining popularity, and in this context SA plays a key role since it represents the reference skeleton component-based applications are based on. The two papers by Nima Kaveh and Wolfgang Emmerich and by Paola Inverardi and Massimo Tivoli describe how analysis and verification of such systems can be carried out by taking advantage of the architectural/middleware supporting infrastructure. The role of SA for the validation of the implemented systems is addressed in Antonia Bertolino, Henry Muccini, and Paola Inverardi's paper. There an approach for SA-based conformance testing is described: architectural tests are selected from a labelled transition system representing the SA behavior and are then refined into concrete tests to be executed on the implemented system. José Luiz Fiadeiro and Luis Filipe Andrade devise in their paper the use of SA for evolutionary purposes, that is as a means to cope with system changes throughout the system's life time. The architectural support for mobile applications is addressed in Amy Murphy, Gian Pietro Picco, and Gruia-Catalin Roman's paper, where the role of formal middleware modeling is exploited to develop mobile applications. The last two papers deal with SA and nonfunctional system attributes. The paper by Simonetta Balsamo, Marco Bernardo, and Marta Simeoni proposes a methodology based on the combined use of stochastic process algebras and queueing networks for predicting, improving, and comparing the performance of alternative architectural designs. The paper by Valérie Issarny and Apostolos Zarras focuses on dependability issues and presents an approach, supported by suitable tools, to depict what needs to be specified at the architectural level to enable the automated generation of models for dependability analysis.

We believe that this book offers a comprehensive view of what has been done and what is going on at present in terms of SA description languages and analysis and testing support in the international community. We wish to thank all the lecturers and all the participants for a lively and fruitful school.

September 2003

Marco Bernardo and Paola Inverardi
SFM 2003:SA Directors