# Preface

There are two aspects to object-oriented programming: encapsulation of state and procedures to manipulate the state; and inheritance, polymorphism, and code reuse. The first aspect provides a correspondence with the natural separation of physical objects, and the second aspect is a way of organizing into categories and sharing by abstracting commonalities. Although much of the early research in object-based programming on sequential systems, in recent years concurrency and distribution have become the dominant concern of computer science. In fact, objects are a natural unit of distribution and concurrency – as elucidated quite early on by research on the Actor model.

It is therefore natural to look at models and theories of concurrency, the oldest of these being Petri nets, and their relation to objects. There are a number of directions in which models of concurrency need to be extended to address objects – these include the semantics of concurrent object-oriented programming languages, object-oriented specification languages, and modeling methods. Developments in these areas will lead to better system development and analysis tools. Among the issues that must be addressed by formal models of concurrency are inheritance, polymorphism, dynamically changing topologies, and mobility. These present deep challenges for the modeling capabilities of the existing formalisms and require investigation of ways to extend these formalisms.

For these reasons, we organized a series of two workshops on Models of Concurrency and Object-Oriented Programming which were held in conjunction with the Annual Petri Net Conference in 1995 (Turin, Italy) and in 1996 (Osaka, Japan). Subsequently, a subset of authors at these workshops, as well as a few others, were invited to submit papers for this volume. The papers then underwent a referee cycle with revisions.

The papers in this volume are organized into three sections. The first consists of long papers, each of which presents a relatively detailed approach to integrating Petri nets and object orientation – i.e. defining a syntax and its semantics, and illustrating it with examples. Section II includes shorter papers where the emphasis is on concrete examples to demonstrate an approach. Finally, Section III includes papers which significantly build on the Actor model of computation.

In order to provide an opportunity to more easily understand and compare the various approaches, we solicited some sample problems at the first workshop, which could serve as canonical case studies. In preparation of the second workshop, two case studies were then suggested to the authors – namely, the Hurried Philosophers problem, proposed by C. Sibertin-Blanc, and the specification of a Cooperative Petri Net Editor proposed by R. Bastide, C. Lakos, and P. Palanque. Many papers in this collection deal with these two problems and the original text describing the problems is included in the last section of this volume.

We are grateful to the referees and to the authors for their patience in the production of this volume. We would like to thank the organizers of the Petri Net conference for their help in organizing the workshops. Finally, we are also very grateful to Alfredo Chizzoni for his support in organizing the two workshops, and to Nadeem Jamali and Patricia Denmark for their assistance in preparing the manuscript.

January 2001

G. Agha
F. De Cindio
G. Rozenberg