Preface

Increasing complexity in engineering projects together with requirements on reduced design time and low costs raise difficult challenges in industry. Thus, effective tools for correct by construction design or for design verification are very much needed. In practice, design and verification are done hierarchically and at several abstraction levels. This book addresses the design of tools for correct by construction synthesis at the discrete-event level of abstraction. The approach of this book is to use Petri nets as discrete-event models and structural methods for the synthesis of supervisors enforcing the design specifications. This approach promises significant computational benefits.

Discrete-event systems are a high-level representation of dynamical systems that exclude continuous dynamics details and timing information. Discrete-event modeling has been used to represent a variety of systems, from computer programs and logic circuits to manufacturing systems and traffic control systems. Notably, purely continuous dynamic systems can also be abstracted as discrete-event systems.

In the supervision of discrete-event systems, the goal is to design supervisors that ensure that only the behaviors consistent with the specification may occur in the system. The supervisor can be seen as an enhancement of the design, and is implemented by additional programming code or hardware. This book addresses the formal development of tools that design supervisors automatically, given a Petri net model and a specification. Apart from supervision design tools, a designer must do by hand the mistake-prone part of coordinating the discrete-event modules such that specification constraints are satisfied. On the other hand, when supervision design tools are used, the focus of the designer is shifted towards developing a formal specification of the desired properties of the system. For instance, a software developer could use such tools to generate the code that ensures that parallel threads of a program do not deadlock each other.

Petri net models arise naturally in the context of concurrency and discrete-event systems. Petri nets have been used in applications from various fields, such as flexible manufacturing, process control in the chemical industry, soft-

ware specification, communication protocols, asynchronous digital circuits, and robotics. Compared to automata, Petri nets have been appreciated for offering a compact, higher level representation of concurrent systems. Note that reachability analysis can be used in order to obtain an automaton equivalent to a Petri net. However, since automata represent explicitly the reachable states of a system, an automaton representation of a Petri net model may not be finite.

This book focuses on methods that exploit the structure of Petri nets for supervisor design. This structural approach avoids reachability analysis and promises to alleviate or, for certain problems, altogether remove the state-space explosion problem. This benefit is often achieved at the cost of the suboptimality of the design. Indeed, structural analysis cannot provide all details available in a reachability analysis. Compared to the supervision of discrete-event systems based on automaton models, the supervisor design for Petri net models is equivalent only if reachability analysis is used. Thus, structural methods provide an interesting alternative to the traditional supervision of discrete-event systems, which is based on automaton models.

In this book we present structural methods for the design of Petri net supervisors. Following much of the literature, we focus on specifications represented by conjunctions of linear constraints, specifically by systems of linear inequalities in terms of the Petri net marking. This class of specifications has grown out of the need to express mutual exclusion constraints and has been proven to be useful in many applications. Further, as we show in this book, many problems involving more general specifications, such as languages or disjunctions of linear constraints, can be written in terms of the linear inequality specifications on transformed Petri nets.

Much of the material in this book deals with methods for the design of supervisors in both centralized and decentralized settings. The important and difficult problem of liveness enforcement is also addressed: a structural method for liveness enforcement is presented. We conclude with additional results that explore the application of Petri net methods to the supervision of concurrent hybrid systems. Note that hybrid systems are systems having both discrete-event dynamics and continuous dynamics.

After the introduction in Chapter 1, a concise introduction to Petri nets is presented in Chapter 2. More Petri net concepts are defined in the following chapters, as needed. Chapter 3 introduces the supervision problem for Petri nets and contains an overview of the methods available in the literature. Specifically, the chapter focuses on structural methods dealing with a particular type of specifications, which are described by systems of linear inequalities in the Petri net marking. In Chapter 4 it is shown how some important classes of specifications can be reduced to the marking inequality specifications on transformed Petri nets. Notably, language specifications and constraints described by disjunctions of inequalities are treated here, under appropriate assumptions.

Chapter 5 considers the problem of decentralized or distributed supervision. Here, the problem is to design decentralized/distributed supervisors that ensure that a global specification on a system is satisfied. The design problem is approached in several settings: when the supervisors cannot communicate; when the communication between supervisors is unrestricted; and when the communication is constrained. The goal here is also to minimize a communication cost. Chapters 6 and 7 deal with liveness specifications. Note that a system is live when no deadlocks are possible. Important concepts and results relating the structure of Petri nets to deadlocks are presented in Chapter 6. These are used in Chapter 7 in a structural method for the design of liveness enforcing supervisors.

Chapters 8 and 9 consider the supervision of concurrent hybrid systems. The approach here is to reduce the problem to the supervision of Petri nets. The approach is hierarchical, involving low-level controllers of hybrid systems, and a higher level discrete-event supervisor, which issues commands to the low-level controllers. The discrete-event supervisor is designed based on the Petri net abstraction of the closed loop of controllers and hybrid systems. The control and abstraction of hybrid systems is described in Chapter 9, while supervisor design for abstractions is discussed in Chapter 8.

The authors have made a significant effort to ensure that the book is readable. All relevant background is explained in the book, and numerous examples are used to illustrate the material. The index at the end contains references to all concepts defined in the book. This book is rigorous, with results being formally stated and proven. The book is also practical, as it includes fully developed methods, ready to be implemented in software. A MATLAB® toolbox has already been developed containing many of the methods presented in this work. The toolbox is available on the web, at `http://www.letu.edu/people/marianiordache`. Many of the examples given in this book have been solved using this software toolbox.

This book represents a novel contribution to the field. The literature already contains some books on this topic that provide valuable material, which is complemented and expanded in the present book. The authors believe the ideas presented here may help researchers and developers from various engineering fields reduce the computational burden of verification or supervision problems for discrete-event models. This book can also be used by graduate students and for advanced courses in discrete-event systems.

LeTourneau University, *Marian V. Iordache*
University of Notre Dame, *Panos J. Antsaklis*
November 2005