Chapter 1
Conceptual Modeling Using Petri Nets

Antoni Guasch
Universitat Politècnica de Catalunya - Barcelona Tech, Spain

Jaume Figueras
Universitat Politècnica de Catalunya - Barcelona Tech, Spain

Josep Casanovas
Universitat Politècnica de Catalunya - Barcelona Tech, Spain

ABSTRACT

Petri nets are used by our students as a formal modeling technique before building a working simulation model in Arena or Simio. The Petri net model enables the simulation analyst to build a complete, unambiguous, and readable model of the target process before coding it in the target simulation tool. One of the aims of this chapter is to emphasize the need for formal specification of the simulation model before it is coded in the chosen target simulation environment. Formal specification of the model is of great help throughout the simulation project life cycle, especially in the coding and verification phase.

INTRODUCTION

Petri nets have proved to be a successful tool for modeling logistic and manufacturing systems thanks to a series of properties, including the conciseness with which they embody static structure and dynamics, the availability of mathematical analysis techniques, and the clear graphical nature (Jensen, 1997; Silva & Valette, 1989;

DOI: 10.4018/978-1-4666-4369-7.ch001

Zimmermann, Dalkowski, & Hommel, 1996). Furthermore, Petri nets are very suitable for modeling and visualizing patterns of behavior comprising concurrency, synchronization, and resource sharing, which are key factors in optimizing system performance.

Another key feature that makes Petri nets an ideal choice of conceptual modeling formalism is the ease with which a Petri net model can be mapped into Arena simulation code. From a scientific perspective, this might seem redundant, since Petri net simulators (www.informatik.uni-hamburg.de/TGI/PetriNets/tools/quick.htm) can be used to analyze the model behavior, making the translation to Arena unnecessary. However, classical simulators such as Arena are the standard choice in industry, whereas Petri net simulators are largely confined to academic circles.

This chapter introduces the reader to ordinary Petri nets, which are called Place/Transition nets (PTN), Timed Petri nets (TPN), and Colored Timed Petri nets (CTPN). We explain how basic Petri net structures can be mapped to Arena code. Finally, we present a series of examples that combine the Petri net modeling formalism and Arena coding.

The chapter is intentionally written in an informal style, with the aim of explaining the key concepts through examples and omitting reference to formal specifications where they are not strictly relevant.

PETRI NETS

Petri nets originate from Carl Adam Petri's doctoral thesis of 1962, "Kommunikation mit Automaten" (Petri, 1962), which introduced a new model of information flow in systems. Today, Petri nets are commonly used to model a variety of discrete event systems, such as communication protocols and networks; manufacturing, production and scheduling systems; logistic systems; and the design, specification, simulation and validation of software systems. Petri nets have a number of advantages:

- They capture the precedence relations and structural interactions of concurrent and asynchronous events.
- Their precise graphical formalism simplifies the visualization of complex systems.
- Petri net theory provides an integrated methodology for modeling physical systems and complex decision processes.
- They provide a uniform environment for modeling and formal analysis. The same model supports the construction of discrete event simulators and controllers as well as the formal verification of behavioral properties such as the