ADVANCES IN AUTOMATIC CONTROL

edited by

Mihail Voicu

KLUWER ACADEMIC PUBLISHERS

PETRI NET TOOLBOX – TEACHING DISCRETE EVENT SYSTEMS UNDER MATLAB

Octavian Pastravanu, Mihaela-Hanako Matcovschi and Cristian Mahulea
Department of Automatic Control and Industrial Informatics
Technical University "Gh. Asachi" of Iasi
Blvd. Mangeron 53A, 700050 Iasi, Romania, Tel./Fax: +40-232-230.751
E-Mail: {opastrav, mhanako, cmahulea}@delta.ac.tuiasi.ro

Abstract A MATLAB toolbox has been developed to handle the basic problems of discrete event dynamical systems that are modeled by Petri nets. In the ***Petri Net Toolbox*** five types of Petri nets (untimed, transition-timed, place-timed, stochastic and generalized stochastic), with finite or infinite capacity, can be used. A user-friendly graphical interface allows activating three simulation modes (accompanied or not by animation) and running specific functions that cover the key topics of analysis such as coverability tree, structural properties (including invariants), time-dependent performance indices, max-plus state-space representations. A design procedure is also available, based on parameterized models. By incorporating instruments to explore the dynamics of Petri net models, as well as animation facilities to support the intuitive understanding and to guide the users in the exploitation of the software, the ***Petri Net Toolbox*** proves to be a valuable aid for Control Engineering education.

Keywords Control Engineering education, discrete event systems, Petri nets, MATLAB software

1. Motivation and objectives

The ***Petri Net Toolbox*** (***PN Toolbox***) was designed, implemented and tested at the Department of Automatic Control and Industrial Informatics of the Technical University „Gh. Asachi" of Iaşi. It is software for simulation, analysis and design of *discrete event systems* (DES), based on *Petri net* (PN) models, and embedded in the MATLAB environment. Our initiative brought remarkable benefits for training and research because Control Engineering students are familiar with the exploitation of *Graphical User Interfaces* (GUIs) (The MathWorks, 2001a) based on this popular software. The integration of the ***PN Toolbox*** with the MATLAB philosophy has the incontestable merit of broadening the MATLAB's utilization domain towards the area of discrete-event systems, which is now covered only by the State-Flow package. The orientation of the ***PN Toolbox*** was also meant to permit further development in the sense of hybrid systems, since MATLAB incorporates comprehensive libraries for studying continuous and discontinuous dynamics.

In the current version (namely 2.0) of the ***PN Toolbox***, five types of classic PN models are accepted, namely: untimed, transition-timed, place-timed, stochastic and generalized stochastic. The timed nets can be deterministic or stochastic, and the stochastic case allows using appropriate functions to generate random sequences corresponding to probability distributions with positive support. The default type of an arc is regular, but the user is allowed to change it into double or inhibitor, if necessary. Unlike other PN software, where places are meant as having finite capacity, our toolbox is able to operate with infinite-capacity places. In addition, the ***PN Toolbox*** allows the assignment of priorities and/or probabilities to conflicting transitions. As an educational aid, this software is suitable for applications illustrating the theoretical concepts provided by courses on PNs with different levels of difficulty, e.g. (Pastravanu, 1997), (Pastravanu *et al.* 2002), (Matcovschi, 2003), allowing relevant experiments for studying the event-driven dynamics of physical systems encountered in many technical fields (such as flexible manufacturing systems (FMSs), computer systems, communication protocols, power plants, power electronics).

The main goal envisaged by the designers of the ***PN Toolbox*** was to provide a collection of instruments for education and training at a graduate level, exploitable under MATLAB. Therefore, the focus was placed on developing students' skills in mastering PN models as a generous framework for dealing with discrete-event systems. Although a large number of tools are advertised for various types of PN problems (Mortensen, 2003), the unified treatment permitted by the ***PN Toolbox*** for untimed, deterministic/stochastic *P*- and *T*-timed PNs, stochastic and generalized stochastic PNs, ensures the premises for an efficient instruction. Thus, the user needs a short time to learn how to handle the ***PN Toolbox*** and his major intellectual effort is invested in the construction and careful analysis of the PN models. The interest shown by the authors for the convenient usage of the ***PN Toolbox*** is reflected by the numerous improvements brought to its previous versions (Mahulea *et al.*, 2001), (Matcovschi *et al.*, 2001), (Matcovschi *et al.*, 2002).

For attaining the proposed teaching goal, we preferred to orient our work towards enhancing the quality and reliability of the procedures devoted to standard topics rather than developing new algorithms. Consequently, the authors' attention focused on the following targets: (i) implementation of efficient algorithms for simulation, analysis and synthesis, (ii) creation of powerful visual support for the intuitive understanding of PN model usage, and (iii) elaboration of a comprehensive online help including animated demonstrative examples of handling the software.

2. Simulation, analysis and design

The ***PN Toolbox*** has an easy to exploit GUI (Matcovschi *et al.*, 2003) that gives the possibility to draw PNs in a natural fashion and allows a straightforward access to various commands starting adequate procedures for exploiting the PN models.

The simulation mechanism is based on the rule for enabling and firing of transitions specific to the type of the current PN model. Consequently, the simulation is driven by an asynchronous clock corresponding to the occurrence of events (Cassandras, 1993). In the untimed case, the sequencing of the events is reduced to simply ordering their occurrence, without any temporal significance, unlike the timed case when simulation requires a continuous correlation with physical time.

Three modes of simulation are implemented in the ***PN Toolbox***, namely: *Step*, *Run Slow* and *Run Fast*. The *Step* and *Run Slow* simulation modes are accompanied by animation; the user can record the progress of the simulation in a log file with HTML format. After ending a simulation (run in any of the three modes) a number of *Performance Indices* are available to globally characterize the simulated dynamics. They refer to: (i) transitions: *Service Sum* (the total number of firings during the simulation), *Service Rate* (the mean frequency of firings), *Service Distance* (the mean time between two successive firings), *Utilization* (the fraction of time when server is busy); and (ii) places: *Arrival Sum*, *Throughput Sum* (the total number of arrived/departed tokens), *Arrival Distance*, *Throughput Distance* (the mean time between two successive instants when tokens arrive in/depart from the place), *Waiting Time* (the mean time a token spends in a place), *Queue Length* (the average number of tokens weighted by time). For timed or (generalized) stochastic PNs, while in the *Step* and *Run Slow* simulation modes, the *Scope* facility opens a new MATLAB window that displays (dynamically) the evolution of a selected performance index versus time.

For untimed PN models, the *behavioral properties* (e.g. boundedness, liveness, reversibility, etc.) may be studied based on the *coverability tree* of the net. The coverability tree is built with or without the ω-convention. The ω-convention means the usage of a generic symbol (herein denoted by "ω") for referring to unbounded markings (Murata, 1989). The *structural* *properties* are approached as integer programming problems (Matcovschi *et al.*, 2001); the minimal-support P- and T-invariants (Martinez and Silva, 1982), (David and Alla, 1992) are displayed, on request, in separate windows.

A facility for the synthesis of timed or (generalized) stochastic PN models is *Design*, which allows exploring the dependence of a *Design Index* on one or two *Design Parameters* that vary within intervals defined by the user. For each test-point belonging to this (these) interval(s) a simulation-experiment is performed in the *Run Fast* mode. The results of all these simulation-experiments yield a graphical plot (2-D or 3-D, respectively) defining the dependence of the selected *Design Index* on the *Design Parameter*(*s*); the extreme values of the *Design Index* are numerically displayed.

The ***PN Toolbox*** is able to derive, directly from the topology and initial marking of a place-timed event graph, the max-plus state-space representation (Bacelli *et al.*, 1992). The following facilities are available for the max-plus analysis (Matcovschi *et al.*, 2002): ● displaying the matrix-form of the equations; ● max-plus simulation; ● graphical plots of the simulation results.

3. Visual information and animated demos

To enlarge the addressability of the ***PN Toolbox***, it includes a series of animation facilities aiming either to support the intuitive understanding or to guide the users in the exploitation of the software.

In the simulation modes *Step* and *Run Slow*, numerical computation is accompanied by animation whose role consists in feeding the user with visual information (current token contents of the places, currently firing transition), complementary to the numerical data available at the end of a simulation experiment. The animation technique is based on the general philosophy of the *object-oriented graphics system*, called **Handle Graphics** (The MathWorks Inc., 2001b). The nodes and arcs of a model are uniquely identified as MATLAB objects whose properties define (i) the characteristics of the PN, (ii) the graphical representation of the objects in the special area reserved for model drawing and (iii) the simulation status. The animation effects are obtained by automatically calling the **set** function for the properties referring to the appropriate instance of an object.

At the same time, the ***PN Toolbox*** was meant to illustrate, by short movies, behaviors that are typical for discrete event systems, for example sequential/parallel sharing of resources, routing policies, services in queuing networks, etc. The implementation combines, by means of the ActionScript Toolbox for Macromedia Flash (Macromedia, 2003), various techniques such as 2D and 3D graphics developed in Adobe Photoshop 7 (Adobe Systems Inc, 2003) and Maya 4.5 (Alias|Wavefront Inc, 2003), respectively. Each movie shows the physical motion of a real-life system synchronized with the token dynamics in the associated PN model, as resulting from the tutorial examples commented on in the following section. The movies are accessible on the web site we have created for the ***PN Toolbox*** (Mahulea *et al.*, 2003).

On the ***PN Toolbox*** site, the user can also find the online help of our software as well as some animated demos whose purpose is to present specific sequences of operations in handling the GUI and the interpretation of numerical results. Watching these demos, the user learns how to handle the key problems of discrete event systems within a PN framework: usage of adequate PN type (untimed, P/T-timed, stochastic or generalized stochastic) in model construction, study of behavioral/structural properties, analysis of max-plus representation, simulation and interpretation of the results, parameterized design, etc.

4. Tutorial examples

The four tutorial examples briefly described below were designed to prove the effectiveness of the ***PN Toolbox*** in assisting the DES training based on the Petri net theory. These examples cover a large area of classical topics and the incorporated animation is extremely profitable especially for the beginners along the lines detailed in the previous section.

Demo 1 refers to a computer system with two processors sharing two disks (in parallel) which is a version of the "Two Dinning Philosophers" well-known problem (Dijkstra, 1968), illustrated by a movie from which a frame is captured in fig. 1.

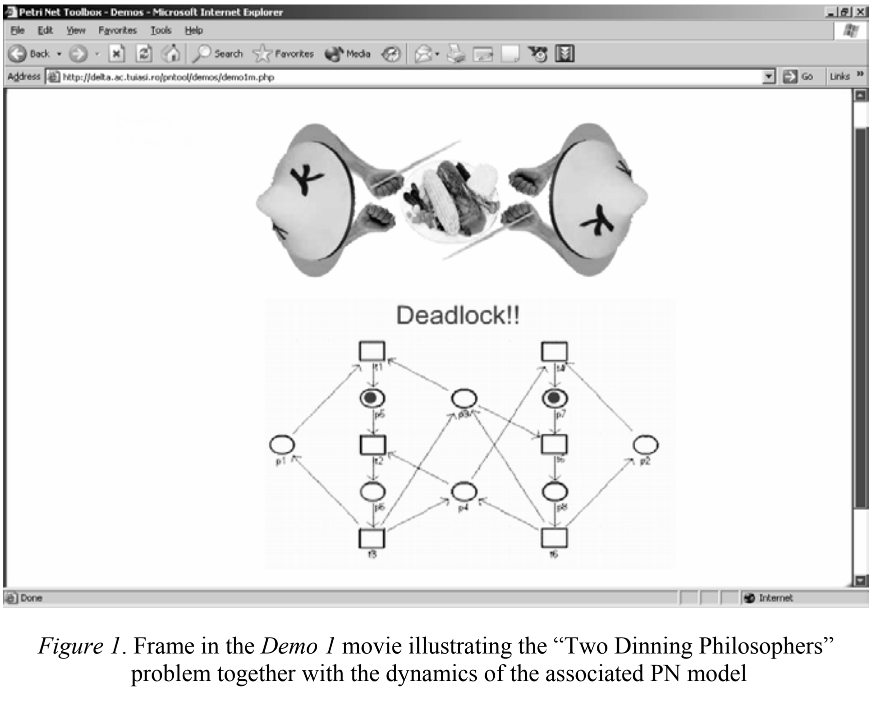


Figure 1. Frame in the *Demo 1* movie illustrating the "Two Dinning Philosophers" problem together with the dynamics of the associated PN model

The addressed problems are: ● construction of an untimed Petri net model; ● analysis of deadlock (via the coverability tree); ● prevention of deadlock through lookahead feedback (Lewis *et al.*, 1995); ● access to the following information about the Petri net model: incidence matrix, minimal-support P- and T-invariants, structural properties.

Demo 2 refers to a manufacturing system with a sequentially shared robot (Desrocheres and Al-Jaar, 1993), (Zhou and DiCesare, 1993), illustrated by a movie from which a frame is captured in fig. 2. The addressed problems are: ● construction of a P-timed Petri net model; ● analysis of deadlock (via simulation); ● prevention of deadlock by limiting the number of pallets; ● analysis of time-dependent performance indices and ● study of a performance index depending on two design parameters (see fig. 3).

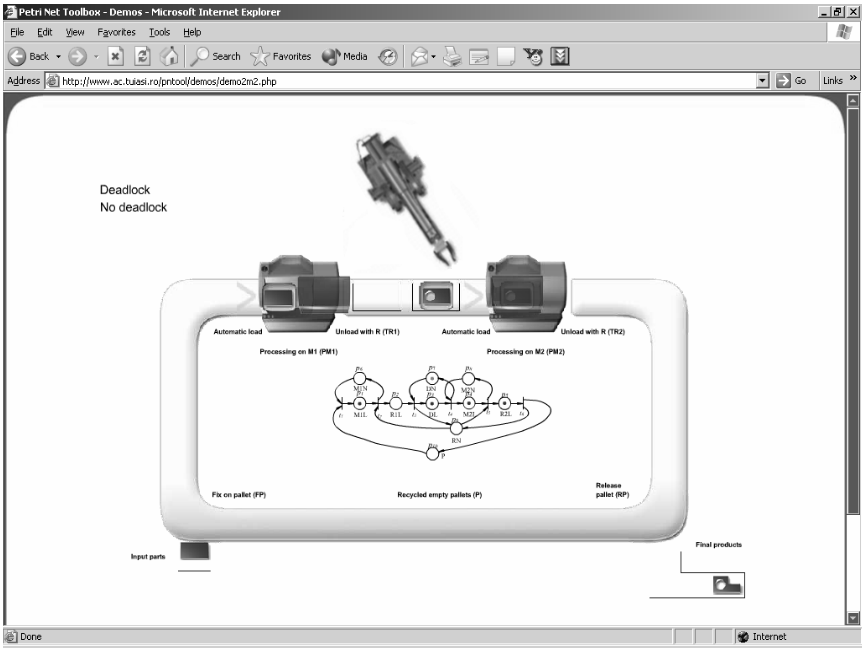
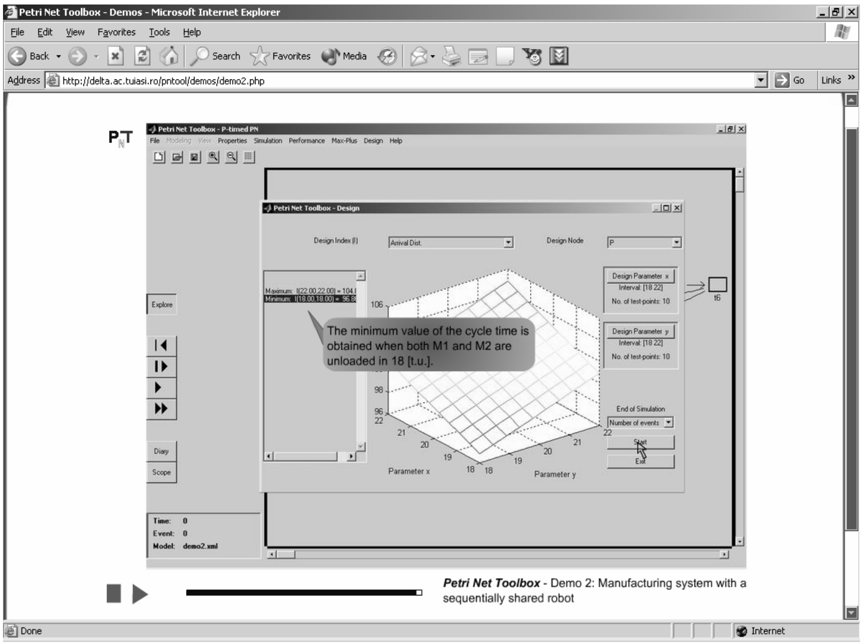


Figure 2. Frame in the *Demo 2* movie illustrating the functioning of a manufacturing system concomitantly with the dynamics of the associated PN model

Demo 3 refers to a flow-shop system with three machines, adapted from (Bacelli *et al.*, 1992). The addressed problems are: ● simulation and animation in the Run Slow mode; ● record of the simulation results in a log file; ● computation of the cycle time, ● max-plus analysis of a place-timed event graph: max-plus state-space representation, setting of the values for the input vectors, max-plus based simulation and plots of the components for the input, state or output vectors (see fig. 4).

Demo 4 refers to an open markovian queuing network (Cassandras, 1993). This demo illustrates: ● construction of a generalized stochastic Petri net model; ● usage of the Scope and Diary facilities; ● analysis of time-dependent performance indices.



Petri Net Toolbox - Demos - Microsoft Internet Explorer

Address http://delta.ac.tuiasi.ro/pntool/demos/demo2.php

Petri Net Toolbox - P-timed PN

Petri Net Toolbox - Design

Design Index (I): Arrival Dist.

Design Node: P

Maximum: I(22.00,22.00) = 104.

Minimum: I(18.00,18.00) = 96.8

The minimum value of the cycle time is obtained when both M1 and M2 are unloaded in 18 [t.u.].

Design Parameter x – Interval: [18 22] – No. of test-points: 10

Design Parameter y – Interval: [18 22] – No. of test-points: 10

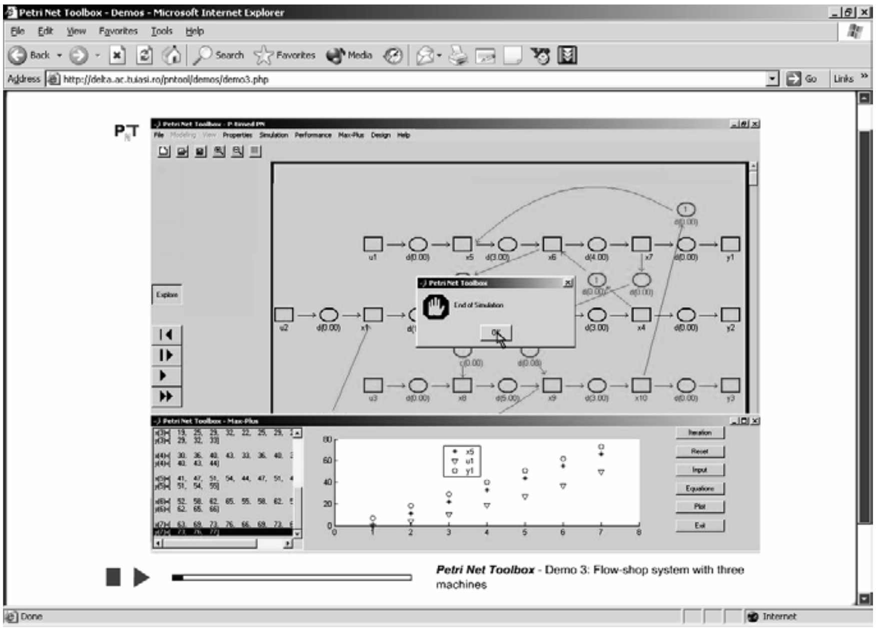
End of Simulation: Number of events

Time: 0
Event: 0
Model: demo2.xml

Petri Net Toolbox - Demo 2: Manufacturing system with a sequentially shared robot

File Edit View Favorites Tools Help
Back Search Favorites Media
Go Links
File Modeling View Properties Simulation Performance Max-Plus Design Help
PNT
Explore
Diary
Scope
106
98
96
22
21
20
19
18
Parameter x
Parameter y
t6
Start
Exit
Done
Internet

Figure 3. Frame in *Demo 2* introducing, by animation, the usage of parameterized design



Petri Net Toolbox - Demos - Microsoft Internet Explorer

Address http://delta.ac.tuiasi.ro/pntool/demos/demo3.php

Petri Net Toolbox - P-timed PN

Petri Net Toolbox

End of Simulation

Petri Net Toolbox - Max-Plus

Petri Net Toolbox - Demo 3: Flow-shop system with three machines

Figure 4. Frame in *Demo 3* presenting, by animation, the usage of max-plus analysis

5. Conclusions

Despite the large offer of software products available for MATLAB, none of its toolboxes provides instruments able to handle Petri net models. This fact has motivated the development of the ***PN Toolbox*** based on a user-friendly graphical interface that makes it very attractive for students because they don't have to spend time for code writing and their attention can exclusively focus on the topics of Control Engineering. The facilities created for simulation, analysis and design prove useful in many types of applications including a wide range of event-driven dynamics, as illustrated by the four tutorial examples briefly presented in the text.

References

Adobe Systems Inc (2003). *Home Page*, http://www.adobe.com.

Alias|Wavefront Inc. (2003). *Home Page*, http://www.aliaswavefront.com.

Bacelli, F., G. Cohen, G.J. Olsder and J.P. Quadrat (1992). *Synchronization and Linearity, An Algebra for Discrete Event Systems*, Wiley, New York.

Cassandras, C.G. (1993). *Discrete Event Systems: Modeling and Performance Analysis*, Irwin.

David, R. and H. Alla (1992). *Du Grafcet aux réseaux de Petri* (2e édition), Hermes, Paris.

Desrocheres, A.A. and R.Y. Al-Jaar (1993). *Modeling and Control of Automated Manufacturing Systems*. IEEE Computer Society Press, Rensselaer, Troy, New-York.

Djikstra, E. W. (1968). "Co-operating sequential processes", in Genyus, F. (ed.) *Programming Languages*, New York Academic, pp. 43-112.

Lewis, F., H.H. Huang, O. Pastravanu and A. Gurel (1995). "Control Systems Design for Flexible Manufacturing Systems", In: A. Raouf, and M. Ben-Daya (Eds.), *Flexible Manufacturing Systems: Recent Developments*, Elsevier Science, pp. 259-290.

Macromedia Inc. (2003). *Home Page*, http://www.macromedia.com.

Mahulea, C., L. Bârsan and O. Pastravanu (2001). "Matlab Tools for Petri-Net-Based Approaches to Flexible Manufacturing Systems", In: F.G. Filip, I. Dumitrache and S. Iliescu (Eds.), *9th IFAC Symposium on Large Scale Systems LSS 2001*, Bucharest, Romania, pp. 184-189.

Mahulea, C., M.H. Matcovschi and O. Pastravanu (2003). *Home Page of the Petri Net Toolbox*, http://www.ac.tuiasi.ro/pntool.

Martinez, J. and M. Silva (1982). "A Simple and Fast Algorithm to Obtain All Invariants of a Generalized Petri Net", In: C. Girault and W. Reisig (Eds), *Application and Theory of Petri Nets*, Informatik Fachberichte 52, Springer, pp. 301-310.

Matcovschi, M.H., C. Mahulea and O. Pastravanu (2001). "Exploring Structural Properties of Petri Nets in MATLAB", *7th International Symposium on Automatic Control and Computer Science SACCS 2001*, Iasi, Romania.

Matcovschi, M.H., C. Mahulea and O. Pastravanu (2002). "Computer Tools For Linear Systems Over Max-Plus Algebra", *5th International Conference on Technical Informatics CONTI'2002*, Timisoara, Romania.

Matcovschi, M.H. (2003). *Markov Chains and Markovian Queueing Systems* (in Romanian), Ed. Gh. Asachi.

Matcovschi, M.H., C. Mahulea and O. Pastravanu (2003). "Petri Net Toolbox for MATLAB", *11th IEEE Mediterranean Conference on Control and Automation MED'03*, Rhodes, Greece.

The MathWorks Inc. (2001a). *Building GUIs with MATLAB*. Natick, Massachusetts.

The MathWorks Inc. (2001b). *Using MATLAB Graphics*. Natick, Massachusetts.

Mortensen, K.H. (2003). *Petri Nets Tools and Software*, http://www.daimi.au.dk/PetriNets/tools.

Murata, T. (1989). "Petri Nets: Properties, Analysis and Applications", *Proc. of the IEEE*, vol. **77**, pp. 541-580.

Pastravanu, O. (1997). *Discrete Event Systems. Qualitative techniques in a Petri Net Framework* (in Romanian), Ed. Matrix Rom.

Pastravanu, O., M.H. Matcovschi and C. Mahulea (2002). *Applications of Petri Nets in Studying Discrete Event Systems* (in Romanian), Ed. Gh. Asachi.

Zhou, M.C. and F. DiCesare (1993). *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*, Kluwer, Boston.