Generalized Stochastic Petri Nets in Performance Evaluation for Queueing Networks

Mihaela-Hanako Matcovschi and Cristian Mahulea*

**Department of Automatic Control and Industrial Informatics*
Technical University "Gh. Asachi" of Iasi, Blvd. Mangeron 53A, 6600 Iasi, Romania
Phone: +40-32-230751*, Fax*: +40-32-214290
E-Mail: {mhanako, cmahulea}@delta.ac.tuiasi.ro

Abstract: Markovian queueing networks may be modeled by generalized stochastic Petri nets. The simulation of a model in a properly chosen simulator provides information on the performances of the actual system, mainly in case that the analytical study is intricate. After presenting some general notions on queueing networks and on generalized stochastic Petri nets, this paper points out the PN Toolbox for MATLAB, an instrument for modeling and simulating several types of Petri nets. The comparison of results obtained through simulation to the analytical ones for a certain model prove the usefulness of this software package for performance evaluation.

Keywords: queueing network, performance evaluation, stochastic Petri nets, MATLAB.

1. Introduction

There are three general techniques for predicting a system's performances: direct measurement, analytical modeling and simulation. Extensive measurement is most accurate but it is expensive and cannot be used in all situations encountered during the lifetime of a system. Due to the increase of the complexity and automation encountered in modern industrial systems, the performance evaluation for such systems relies more and more on simulation rather than on analytical methods. Moreover, the growing interest in simulation-based techniques for the prediction of reliability, availability and combined performances of complex systems has been stimulated by the advent of fault-tolerance concept. Such techniques provide manufacturers cheap and versatile tools, useful in all the phases of system life cycle. The goal of ease of use is best met with software packages that provide an easy to use and powerful user interface for model specification.

Petri nets (PNs) proved to be a powerful graphical and mathematical modeling tool that allows describing and analyzing complex systems exhibiting concurrency, synchronization and conflicts such as computer systems, communication systems, and manufacturing systems. Their ability to model timed and probabilistic behavior is essential in the field of performance evaluation. This need lead to various different extensions of the PN formalism, where the class of *Stochastic Petri nets* (SPNs) gained the widest acceptance. Many a package software for supporting engineers in the definition and analysis of such type of models have been built over the last years, such as SPNP, GreatSPN, TimeNET. The choice of a particular tool is mainly based on its accessibility as well as on the type of models under study.

Our paper focuses on performance evaluation for queueing networks based on SPN models and simulation procedures. Due to our experience in exploiting MATLAB numerical and programming capabilities, we have designed and implemented a new software, namely PN Toolbox for MATLAB, devoted to modeling, simulation and analysis of PN type models.

This paper has the following structure. Section 2 describes the basics of queueing networks and of their performance evaluation. Generalized stochastic Petri nets (GSPN) used for modeling such complex systems are briefly put forward in Section 3. A concise representation of the facilities available in PN Toolbox is realized in Section 4. An example of its usage in analyzing queueing networks are further on delivered in Section 5. The concluding remarks in Section 6 point out the main results of this paper.

2. Performance evaluation for Queueing Networks

In the past few decades queueing systems have become more and more present in every-day life. There are three basic elements that comprise a queueing system [2]: service centers (*servers*, resources), the entities (customers, jobs) that do the waiting in their quest for resources and the *queues* where the waiting is done. In order to specify a queueing system, the capacity of the queue (maximum number of customers it may accommodate) need to be known. Also, the rule according to which the next customer to be served is selected from the queue, the so called *queueing discipline*, must be specified. Several different queueing disciplines may be considered, the most popular of them being the First In First Out (FIFO) one.

A single queue may be connected to multiple servers and several queues may be connected to a single server or both. Arbitrary series and parallel connections of the building blocks are also possible. This construction leads to the so called *queueing networks* (QNs). The customers are viewed as *flowing* through the components (queues and servers) of the network according to specific rules. The term *node* is used to describe a set of identical parallel servers along with the queueing space that precedes it.

Two types of queueing networks may be identified: open QNs – in which customers arrive from an external source, spend time in the system and finally depart – and closed QNs – in case that the number of customers circulating among the service centers is constant, i.e., no external source of jobs and no departure.

Queueing networks may model complex systems of different natures, such as computer systems, communication systems, manufacturing systems and so forth.

Queueing networks represent a special case of systems whose dynamics are driven by discrete events. The basic events that pilot a QN are the arrival of customers to a queue, the start and the end of service (departure of customer). The main characteristics associated to these events are the interarrival times and service times, which are, generally speaking, stochastic variables. In case that all external arrival events and all departure events at the servers in the system are generated by processes satisfying the markovian (memoryless) property and are, as a result, characterized by exponential distributions, the QN is said to be *markovian*.

In the analysis of markovian QNs *Burke's theorem* [2]: 'A Poisson process supplying arrivals to a server with exponentially distributed service times results in a Poisson departure process with the exact same rate.' is the most important mathematical instrument. This theorem allows each node to be analyzed independently, the only information required is the number of servers at that node, their service rate and the arrival rate of customers (from other nodes as well as from the outside world). In the example presented in Section 5 the external arrival process is of Poisson type and all the nodes they have 1 server with exponentially distributed service rate, therefore when considered independently they are of M/M/1 type.

The performances of interest when analyzing the steady state behavior of a queueing network are the utilizations of the servers, the expected queue length (the total number of jobs in the queue including the jobs in service), the mean waiting time of a customer at each node, the mean response time of a node for a customer (that is the mean waiting time plus the mean service time). Also, the expected throughput of the system (the rate of customer departures) is of interest. These performances are taken into consideration for the case-study described further on.

3. Generalized Stochastic Petri Nets

A *Petri net* (PN) [11] consists of a particular type of directed weighted bipartite graph and an initial state called *initial marking*. The underlying graph of a PN contains two kinds of nodes, called *places* and *transitions*. Its *weighted arcs* connect a place to a transition or a transition to a place. A *marking* (state) assigns to each position p of the net a nonnegative integer $\mathbf{M}(p)$ (p is marked with $\mathbf{M}(p)$ tokens). A position in a PN may accommodate a finite or infinite number of tokens (*finite* or *infinite capacity nets*).

A state in a Petri net is changed according to the *transition firing rule*. The firing of an enabled transition changes the token distribution in a net. In case that the firings of all the transitions in a PN do not consume time the net is of *untimed* type. For performance evaluation and scheduling problems of dynamic systems it is necessary to introduce time delays associated with transitions and/or places in their net models. This kind of Petri nets models are known as transition (position) timed nets. The delays may be deterministic (if they are a priori known) or stochastic (when they are probabilistically specified).

The Stochastic Petri nets (SPNs) are defined by associating exponentially distributed firing delays to the transitions of a Petri net. Generalized Stochastic Petri nets (GSPNs), introduced in [1] in connection to performance evaluation of multiprocessor systems, relax this condition by also allowing transition firings in constant time zero for the so called *immediate transitions*. The firing of immediate transitions has priority over that of timed transitions. Each immediate transition has associated a weight which determines its firing probability in case this transition is conflicting with some other immediate transition.

In recent years, GSPNs have gained much attention as a useful modeling formalism They have been successfully used in the analysis of the performances of computer systems, communication systems or manufacturing systems [4].

4. PN Toolbox for MATLAB

The Petri Net Toolbox (PN Toolbox) for MATLAB is a software tool designed and implemented at the Department of Automatic Control and Industrial Informatics of the Technical University "Gh. Asachi" of Iasi. Its being embedded in MATLAB presents several advantages when compared to other software due to the powerful algebraic and graphical tools it provides, which exploit the high quality routines available in MATLAB and some of its toolboxes (Statistics Toolbox, Optimisation Toolbox).

The first version of this software was presented in [6]. The current version 2, which has just been released, brings a new technology in the implementation of the Graphical User Interface (GUI) [8], [9], that offers an easy user access to the instruments for modelling, simulation and analysis (see figure 2). It also allows links to other software, upgrades several algorithms, offers specific tools to address structural properties [7], the max-plus simulation and analysis of timed event graphs and incorporates facilities for coloured Petri nets [5]. The MATLAB orientation of the PN Toolbox also ensures the necessary flexibility for further improvement, by upgrading the tools that already exist and by adding new ones.

In the present version of PN Toolbox *untimed*, *transition-timed* and *place-timed* PN models are accepted. The timed nets can be *deterministic* or *stochastic*. Priorities or probabilities can be assigned to conflicting transitions. Since MATLAB includes the built-in function **Inf**, our toolbox is able to operate with nets having *infinite capacity* places, unlike other PN software where places are meant as having finite capacity.

When modelling stochastic Petri nets, there is the possibility of working with both untimed and exponentially distributed timed transitions. The choice of a marking dependent firing rate for timed transitions has also been considered.

The instruments available for generating the accessibility tree for the model loaded in the GUI offers the possibility of generating the corresponding continuous time Markov chain that can be used in the analysis of the model.

The algorithm for the simulation of stochastic Petri nets is based on the general guiding lines presented in [3]. Breakpoints on the time of simulation, number of firings of a transition, number of tokens arrived in a place or number of events may be set.

At the end of simulation, the *Analysis Menu* offers statistics both on transitions (see figure 3) and on places, in separate windows. These results may also be saved as files on the hard-disk in *html* format. From the point of view of this paper the most important ones are:

- for transitions: the total number of firings during the simulation (*Service Sum*), the number of firings per time unit (*Service/Time*);

- for places: the total number of tokens in and out of each position (*Arrival Sum* and *Throughput Sum*), the average number of tokens weighted by time (*Queue length*), the mean time a token spends in a position (*Waiting Time*).

5. Case-study

The example of an open queueing network, adopted from [2], has been considered as illustrative for the usage of PN Toolbox in performance evaluation. The three nodes of the Jackson network presented in figure 1 have infinite capacity queues (Qi, $i = 1,2,3$). The service times of the servers (Si, $i = 1,2,3$) are exponentially distributed, their rates being $m_1 = m_2 = 4\ \mathrm{s}^{-1}$ and $m_3 = 2\ \mathrm{s}^{-1}$ respectively. Each server may process only job at a time.

There is a single external Poisson arrival process at node 1 with rate $r_1 = 1\ \mathrm{s}^{-1}$. This queueing network includes feedback paths from node 1 to itself and from node 2 to both nodes 1 and 2. Numerical values of the routing probabilities p_{ij}, $i, j = 1,2,3$, from node i to node j are: $p_{11} = 30\%$, $p_{12} = 20\%$, $p_{13} = 50\%$, $p_{21} = 20\%$ and $p_{22} = 80\%$.

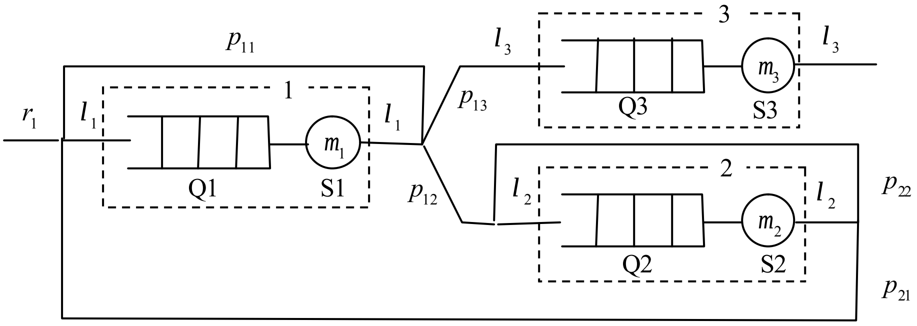


Figure 1. *Queueing network used as case-study.*

In order to test the effectiveness of the simulation-based technique in performance evaluation for this model, the analytical method is employed at the beginning.

The first step in the *analytical analysis* of this system is to calculate the arrival rates l_1, l_2 and l_3 at each node assuming that it is at steady state. In the hypothesis that all nodes are stable, l_1, l_2 and l_3 also represent the throughputs (departure rates) of nodes 1, 2 and 3 respectively. The customer flow balance equations give the desired values, that is:

$$\begin{cases} l_1 = r_1 + p_{11} l_1 + p_{21} l_2, \\ l_2 = p_{12} l_1 + p_{22} l_2, \\ l_3 = p_{13} l_1, \end{cases} \quad \Rightarrow \quad \begin{cases} l_1 = \dfrac{1}{p_{13}} r_1 = 2\ \mathrm{s}^{-1}, \\ l_2 = \dfrac{p_{12}}{p_{21} p_{13}} r_1 = 2\ \mathrm{s}^{-1}, \\ l_3 = r_1 = 1\ \mathrm{s}^{-1}. \end{cases}$$

Taking into account that at steady state the only customer flow out of the system is from node 3 and it must be balanced by the incoming flow the relationship $l_3 = r_1$ could have been directly obtained.

The second step in this investigation is to obtain the utilizations of each server, namely

$$r_1 = \frac{l_1}{m_1} = 0.5, \qquad r_2 = \frac{l_2}{m_2} = 0.5, \qquad r_3 = \frac{l_3}{m_3} = 0.5\ .$$

Since $r_i < 1$ for all $i = 1,2,3$, the stability condition of the system is fulfilled.

The third step is to study each node separately as an M/M/1 system. The performances of node i are given by the mean queue length $\mathrm{M}[X_i]$ (including the clients in the queue and the one currently served), the mean response time $\mathrm{M}[S_i]$ (from the moment a client joins queue Q*i* until it leaves server S*i*) and the mean waiting time $\mathrm{M}[W_i]$ of a client in queue Q*i*, for $i = 1,2,3$. By applying the corresponding formulas, the following results are obtained:

- for node 1:

$$\mathrm{M}[X_1] = \frac{r_1}{1 - r_1} = 1\,\text{client}, \qquad \mathrm{M}[S_1] = \frac{r_1}{l_1 (1 - r_1)} = 0.5\,\mathrm{s}, \qquad \mathrm{M}[W_1] = \mathrm{M}[S_1] - \frac{1}{m_1} = 0.25\,\mathrm{s};$$

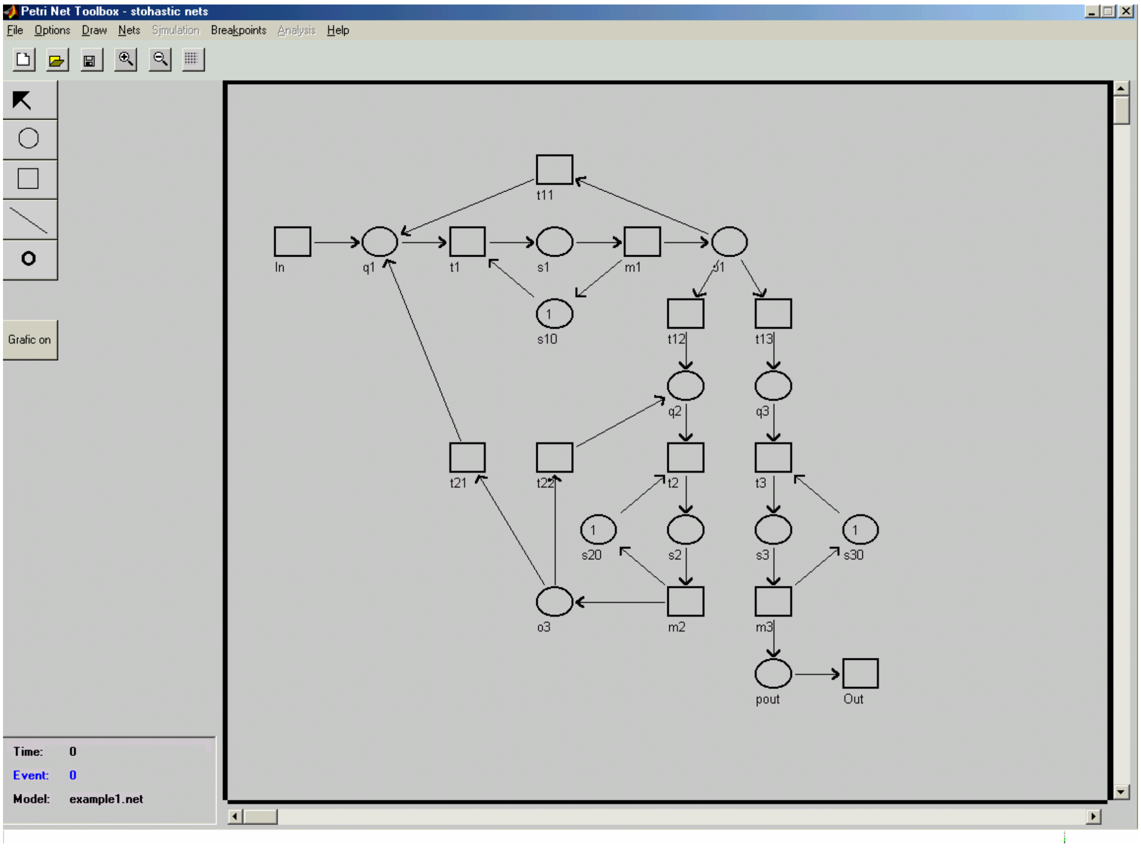
- for node 2:

$$\mathrm{M}[X_2] = \frac{r_2}{1 - r_2} = 1\,\text{client}, \qquad \mathrm{M}[S_2] = \frac{r_2}{l_2 (1 - r_2)} = 0.5\ \mathrm{s}, \qquad \mathrm{M}[W_2] = \mathrm{M}[S_2] - \frac{1}{m_2} = 0.25\,\mathrm{s};$$

- for node 3:

$$\mathrm{M}[X_3] = \frac{r_3}{1-r_3} = 1\,\text{client}, \qquad \mathrm{M}[S_3] = \frac{r_3}{l_3\left(1-r_3\right)} = 1\,\text{s}, \qquad \mathrm{M}[W_3] = \mathrm{M}[S_3] - \frac{1}{m_3} = 0.5\,\text{s}.$$

For comparison, further on the study on the performances of this queueing network are accomplished by means of simulation. Figure 2 presents the main window of the graphical user interface of PN Toolbox. In the drawing area (in the center of the figure) there is the GSPN type model of this queueing network. Positions q1, q2 and q3 correspond to the queues. The two states of server *i* (busy and idle) are modeled by means of positions s*i* and s*i*0, for $i = 1,2,3$.


Petri Net Toolbox - stohastic nets
File Options Draw Nets Simulation Breakpoints Analysis Help
Grafic on
t11
In
q1
t1
s1
m1
s1
1
s10
t12
t13
q2
q3
t21
t22
t2
t3
1
s20
s2
s3
1
s30
o3
m2
m3
pout
Out

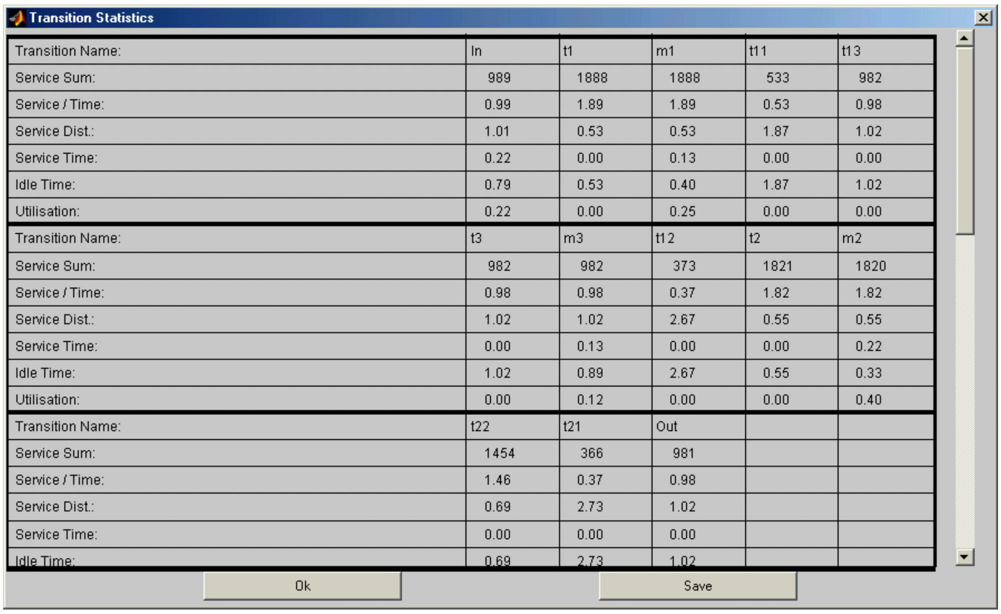
Time: 0
Event: 0
Model: example1.net

Figure 2. *Generalized stochastic Petri net model of the queueing network used as case-study.*

In this model the delays associated to the transitions representing the processes of arrival (In) and the ones of customer service (m1, m2 and m3) are exponentially distributed with the corresponding rates (r_1, m_1, m_2 and m_3 respectively). The rest of the transitions are immediate. The routing probabilities p_{11}, p_{12}, p_{13}, p_{21} and p_{21} are associated to the executions of transitions t11, t12, t13, t21 and t22 respectively. Transitions t1, t2 and t3 model the beginning of service for a customer. The event of customer departure from the system is modeled by the last transition (Out).

The length of the simulation run in PN Toolbox was 1000 s. The numerical values of the parameters related to the transitions are presented in figure 3. A similar window provides statistics corresponding to the positions in the GSPN model.

The *Service Sum* parameters for source transition In and sink transition Out represent the total number of customers that have entered (989) and have gone out (982) of the system during the simulation. In case of transitions m1, m2 and m3 the same parameter gives the total number of clients served by each server, that is 1888, 1820 and 982. The same numerical values are obtained for the *Throughput Sum* parameters corresponding to positions s1, s2 and s3 respectively. The total number of clients that have been routed from server *i* to *j* is given by the total number of executions for t*ij*.



Transition Name:	In	t1	m1	t11	t13
Service Sum:	989	1888	1888	533	982
Service / Time:	0.99	1.89	1.89	0.53	0.98
Service Dist.:	1.01	0.53	0.53	1.87	1.02
Service Time:	0.22	0.00	0.13	0.00	0.00
Idle Time:	0.79	0.53	0.40	1.87	1.02
Utilisation:	0.22	0.00	0.25	0.00	0.00
Transition Name:	t3	m3	t12	t2	m2
Service Sum:	982	982	373	1821	1820
Service / Time:	0.98	0.98	0.37	1.82	1.82
Service Dist.:	1.02	1.02	2.67	0.55	0.55
Service Time:	0.00	0.13	0.00	0.00	0.22
Idle Time:	1.02	0.89	2.67	0.55	0.33
Utilisation:	0.00	0.12	0.00	0.00	0.40
Transition Name:	t22	t21	Out		
Service Sum:	1454	366	981		
Service / Time:	1.46	0.37	0.98		
Service Dist.:	0.69	2.73	1.02		
Service Time:	0.00	0.00	0.00		
Idle Time:	0.69	2.73	1.02		

Figure 3. *Transition statistics obtained through simulation of GSPN model of the queueing network used as case-study.*

The *Service/Time* parameters for transitions In and Out (0.99 and 0.98 respectively) represent the rates of the customer flows in and out of the systems, being approximately equal to the input arrival rate and to the throughput r_1. For transitions t1, t2 and t3, the *Service/Time* parameter represent the arrival rates l_1, l_2 and l_3 at each node.

The piece of information concerning the utilization of the servers (the number of clients being served per time unit) may be extracted from the *Queue length* parameters corresponding to places s1, s2 and s3, being equal to 0.48, 0.26 and 0.47 respectively. For places q1, q2 and q3, the *Queue length* parameter is the average number of clients in Q1, Q2 and Q3 (0.49, 0.24 and 0.49). The mean queue length of each node may be obtained as the sum of the two parameters corresponding to the same node. The *Waiting time* statistics provide information about the mean time a customer spend in the queues (for places q*i*) and at the servers (in case f places s*i*).

The results obtained through simulation in this example, as well as the test that have been made for other systems modeled by GSPNs, were satisfactory so as to conclude that the simulator implemented in PN Toolbox may be used for performance evaluation of queueing networks.

6. Concluding remarks

This purpose of this paper was meant to point out the possibility of performance evaluation through simulation for markovian queueing networks modeled by generalized stochastic Petri nets. The software tool PN Toolbox for MATLAB designed and implemented by us proved to be well suited for this task. The comparison between the analytical results and the ones obtained through simulation led to the conclusion that this instrument may be utilized even for more complex tasks than the one presented in detail in this paper.

Further work is to be done in implementing non-markovian stochastic Petri nets. Our plan also includes the implementation instruments for modeling and analysing Stochastic Reward Petri Nets [10], which are SPNs augmented with the ability to specify output measures as reward based functions for the evaluation of reliability of complex systems.

References

[1]. Ajmone Marsan, M., Balbo, G. and Conte, G. (1984). *A class of stochastic Petri nets for the performance evaluation of multiprocessor systems*. ACM Transactions on Computer Systems, 2 (2), pp. 93-122.

[2]. Cassandras, C.G., Lafortune, S., Olsder, G.J. (1995). *Introduction to the modeling, control and optimization of discrete event systems*. In: (Isidori A., Ed.) Trends in Control – A European Perspective, Springer, Berlin.

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

[4]. Desrocheres, A.A. and Al-Jaar, R.Y. (1993). *Modeling and control of automated manufacturing systems*. IEEE Computer Society Press, Rensselaer, Troy, New-York.

[5]. Mahulea, C. (2002). *Petri Net Toolbox – a MATLAB library for discrete event systems*, MS thesis, The Sheffield University, Department of Automatic Control & System Engineering, UK (ERASMUS / SOCRATES grant).

[6]. Mahulea, C., Bârsan, L., Pastravanu, O. (2002), *MATLAB tools for Petri-net-based approaches to flexible manufacturing systems*. In: (F.G. Filip, I. Dumitrache, S. Iliescu, Eds.) Large Scale Systems: Theory and Applications 2001, Proceedings volume from the 9th IFAC / IFORS / IMACS / IFIP Symposium, Bucharest, Romania, 18-20 July 2001, Elsevier Science, IFAC Proceedings Volumes.

[7]. Matcovschi, M., Mahulea, C., Pastravanu, O., (2001), *Exploring structural properties of Petri nets in MATLAB*, 7th International Symposium on Automatic Control and Computer Science and Parallel Workshop on Control Theory, Editura "Gh. Asachi" Iasi. CD Rom.

[8]. The MathWorks, Inc. (2000a). *Building GUIs with MATLAB*. Natick, Massachusetts.

[9]. The MathWorks, Inc. (2000b). *Using MATLAB graphics*. Natick, Massachusetts.

[10]. Muppala, J.K., Ciardo, G. and Trivedi, K.S. (1994). *Stochastic Reward Nets for Reliability Prediction*, Communications in Reliability, Maintainability and Serviceability: An International Journal published by SAE International, Vol. 1, No. 2, pp. 9-20.

[11] Murata, T. (1989). *Petri nets: properties, analysis and applications.* In: Proceedings of the IEEE, **77**, pp.541-580.