

Throughput Analysis in Timed Colored Petri Nets

W.M. Zuberek

Department of Computer Science
Memorial University of Newfoundland
St. John's, Canada A1C-5S7

A b s t r a c t

It is shown that the steady-state behavior of a class of timed colored Petri nets can be analyzed using the concept of throughput and simple rules of operational analysis. Since such analysis is based on the average values of firing times rather than firing time distribution functions, the same approach can be applied to a variety of net models. Throughput analysis is based on structural properties of nets only, so it avoids the potential complexity of reachability analysis, and it can be used for analysis of unbounded nets.

1. INTRODUCTION

Modeling and analysis of systems of events in which it is possible for some events to occur concurrently, but there are constraints on the precedence or frequency of these occurrences are typical applications of Petri nets [P81, A84, R85, M89]. Examples of such systems include multiprocessor and distributed systems, communication networks and data flow architectures. Traditional methods, developed for analysis of systems with sequential behavior, are insufficient or inadequate for accurate description of concurrent systems.

In order to study performance aspects of Petri net models, the duration of activities must also be taken into account and included into model specifications. Several types of Petri nets 'with time' have been proposed [A84, S77, Z81] by assigning firing times to the transitions of Petri nets or to places of a net. It should be noted, however, that the basic difference between these two approaches is not in association of time but in the mechanism of transition firings. In t -timed nets [Z91], transition firings are not instantaneous events; a firing occurs in 'real-time', i.e., the tokens are removed from input places at the beginning of the firing period, and they are deposited to the output places at the end of this period (sometimes this is called a 'three-phase' firing mechanism). In p -timed nets [A84], the 'firing time' determines the period of time that tokens must 'wait' before firing may occur, while the firing is an instantaneous event, as in ordinary nets (so the 'firing times' are - in fact - 'enabling times'). A simple consequence of this difference is that in p -timed nets, the conflict resolution policies cannot be defined independently of timing specifications, while in t -timed nets conflict resolution and timing specification are independent aspects. Only t -timed nets are discussed in this paper.

In colored Petri nets [J87], information can be associated with individual tokens. This associated information is called a 'color' of a token. Token colors can be quite complex, for example, they can describe the contents of a message package or the contents of a database, but typically they are rather simple, representing a class of jobs or a type of a message in a communication systems, etc. Token colors can be modified by (firing) transitions and also the conditions enabling transitions can be different for different colors.

Attributes associated with tokens result in net models that contain much fewer places and transitions than would be required in 'ordinary' Petri nets. The basic idea behind colored nets is to 'fold' (similar) sections of an ordinary Petri net; the original set of places is partitioned into a set of disjoint classes, and each class is replaced by a single colored place

with colors indicating which of the original places the tokens belong to. Similarly, the original set of transitions is partitioned into a set of disjoint classes, and each class is replaced by a single colored transition with the occurrence colors indicating which of the original transitions the occurrences belong to. In this sense, 'ordinary' Petri nets correspond to colored nets in which there is just one color (and then the color can be simply ignored).

Analysis of net models can be performed on the basis of the 'behavior' of the net, or on the basis of its structure. The behavior of a net is usually represented by the space of reachable states; analysis of this state space is known as the reachability analysis. Although reachability analysis is quite general (e.g., it can easily handle state-dependent routing probabilities as well as state-dependent timing properties), it becomes very inefficient when the state space is large (for some models, the number of states grows exponentially with model parameters, which is known as the "state explosion problem"). Moreover, reachability analysis is usually restricted to bounded nets. Therefore, other approaches are recently gaining popularity, and approaches based on structural properties of models are especially attractive because they avoid the derivation of the state space. Such structural methods of analysis cannot provide as much information as the reachability approach does, however, quite often all the detailed results of reachability analysis are not really needed, and more synthetic performance measures, which can also be provided by structural approach, are quite satisfactory.

Structural approach to analysis of net models is similar to operational methods developed for analysis of queueing systems [B76, D78]. It applies to the steady-state of the system; it uses only the first moment (the average values) of random variables, and general relationships and laws which do not depend upon probability distribution functions. One of the basic variables of operational analysis is throughput. Many other measures can be obtained from throughputs, for example, the throughput and the maximum service rate of a component determine its utilization factor, which - in turn - is an indicator of systems saturation. The throughput is one of the elements used in the Little formula, etc.

The paper first introduces the concept of throughput in timed colored Petri net models and some of its properties, and then applies throughput analysis to performance evaluation of a simple model of a computer systems. The paper is organized in 3 main sections. Section 2 overviews timed colored Petri nets as an extension of marked nets. The concept of throughput is introduced in section 3 and section 4 presents an example of performance analysis based on throughput. A discussion of a few possible extensions concludes the paper.

2. TIMED COLORED PETRI NETS

Colored Petri nets can be defined as an extension of marked Petri nets, as in [Z90]. However, the arc functions are restricted to 'simple' rather than 'multiple' arcs, so sets rather than multisets [J87] are used in the following definition. Also, inhibitor arcs are introduced explicitly.

A simple colored Petri net is a 7-tuple $\mathbf{N} = (P, T, A, \bar{C}, w, h, m_0)$ where:

P is a finite (nonempty) set of places,

T is a finite (nonempty) set of transitions,

A is a (nonempty) set of directed arcs which connect places with transitions and transitions with places, $A \subseteq P \times T \cup T \times P$, such that there are no isolated places or transitions,

C is a finite (nonempty) set of colors which are associated with places (place or token colors) and transitions (occurrence colors),

w is an arc function which assigns to each directed arc from the set A a mapping from the set of occurrence colors C into the powerset of place colors 2^C , $w : A \rightarrow C \rightarrow 2^C$,

h is an inhibitor function which assigns to each directed arc from the set A a mapping from the set of occurrence colors C into the powerset of (place) colors 2^C , $h : A \rightarrow C \rightarrow 2^C$, such that:

$$\forall((p, t) \in A) \forall(o \in C) w(p, t)(o) \cap h(p, t)(o) = \emptyset.$$

m_0 is the initial marking function which assigns multisets of colors to places of a net, $m_0 : P \rightarrow C \rightarrow \mathcal{N}$.

It can be observed that marked nets correspond to such colored nets in which the set of colors C contains just one color.

For all occurrence colors $o \in C$ of all transitions $t \in T$, the sets of input Inp , inhibitor Inh and output Out place colors are defined as:

$$\begin{aligned} Inp(t, o) &= \{(p, c) \in P \times C \mid (p, t) \in A \wedge c \in w(p, t)(o)\} \\ Inh(t, o) &= \{(p, c) \in P \times C \mid (p, t) \in A \wedge c \in h(p, t)(o)\} \\ Out(t, o) &= \{(p, c) \in P \times C \mid (t, p) \in A \wedge c \in w(t, p)(o)\} \end{aligned}$$

where $w(p, t)(o)$ and $h(p, t)(o)$ denote applications of the occurrence color o to the arc functions w and h of the arc (p, t) .

Similarly, for all place colors $(p, c) \in P \times C$:

$$\begin{aligned} Inp(p, c) &= \{(t, o) \in T \times C \mid (t, p) \in A \wedge c \in w(t, p)(o)\} \\ Out(p, c) &= \{(t, o) \in T \times C \mid (p, t) \in A \wedge c \in w(p, t)(o)\} \end{aligned}$$

Let any function m which maps P into multisets of (place) colors, $m : P \rightarrow C \rightarrow \mathcal{N}$, be called a marking in a net \mathbf{N} .

An occurrence color $o \in C$ of a transition $t \in T$ is enabled at the marking m if and only if

$$\forall((p, c) \in Inp(t, o)) m(p, c) > 0 \wedge \forall((p, c) \in Inh(t, o)) m(p, c) = 0.$$

If an occurrence o of the transition t is enabled at the marking m_i , it can fire. The firing of the occurrence (t, o) transforms m_i into another marking m_j such that for each $(p, c) \in P \times C$:

$$m_j(p, c) = \begin{cases} m_i(p, c) - 1, & \text{if } (p, c) \in Inp(t, o) - Out(t, o), \\ m_i(p, c) + 1, & \text{if } (p, c) \in Out(t, o) - Inp(t, o), \\ m_i(p, c), & \text{otherwise;} \end{cases}$$

A place color (p, c) of a net \mathbf{N} is shared iff there are exist occurrences (t_i, o_k) and (t_j, o_ℓ) such that:

$$(p, c) \in Inp(t_i, o_k) \cap Inp(t_j, o_\ell).$$

A shared place color (p, c) is guarded if for all pairs of occurrences (t_i, o_k) and (t_j, o_ℓ) sharing (p, c) there exists another place p_h such that:

$$(p_h, c) \in Inp(t_i, o_k) \wedge (p_h, c) \in Inh(t_j, o_\ell) \vee (p_h, c) \in Inh(t_i, o_k) \wedge (p_h, c) \in Inp(t_j, o_\ell).$$

A shared place color (p, c) is free-choice if all occurrences sharing it have the same input and inhibitor sets, i.e., iff:

$$\forall((t_i, o_k), (t_j, o_\ell) \in Out(p, c)) Inp(t_i, o_k) = Inp(t_j, o_\ell) \wedge Inh(t_j, o_k) = Inh(t_j, o_\ell).$$

A net \mathbf{N} is free-choice iff all its shared place colors are either guarded or free-choice. Only free-choice nets are considered in this paper because almost all practical net models are either free-choice or can easily be converted into equivalent free-choice nets.

The relation of sharing a free-choice place is an equivalence relation in the set $T \times C$, so it implies a partition of this set into a number of free-choice equivalence classes. It can be observed that at any marking m , either all occurrences within any free-choice class are simultaneously enabled or they are simultaneously disabled. It is assumed that selections of firing occurrences within free-choice classes of occurrences are random processes which can be described by probabilities associated with occurrences.

In timed colored nets, transition occurrences fire in ‘real-time’, which means that there is a ‘firing time’ associated with each occurrence of each transition. This firing time may be deterministic, as in D-timed nets [Z91], or it can be a random variable with some distribution function, for example, negative exponential distribution, as in M-timed nets [Z91]. In the second case, the average value of the firing time is used to characterize the timed behavior.

Any enabled occurrence starts its firing in the same time instant in which it becomes enabled. Each firing can be considered as a three-phase event; first, the (colored) tokens are removed from the input places as indicated by the arc functions of the firing occurrence, the second phase is the firing time period, and when it is finished, (colored) tokens are deposited to output places as indicated by the arc functions of the firing occurrence. If a transition occurrence becomes enabled while it is firing, a new independent firing cycle begins.

A timed colored net is a triple, $\mathbf{T} = (\mathbf{N}, u, f)$ where

\mathbf{N} is a colored net, $\mathbf{N} = (P, T, A, C, w, h, m_0)$,

u is a choice function which, for each marking m of \mathbf{N} , assigns the ‘choice’ probability to each occurrence color of each transition, $u : T \times C \rightarrow \mathbf{R}^{[0,1]}$, such that for each free-choice class of occurrences the sum of probabilities is equal to 1,

f is a firing-time function which assigns the average firing time to each occurrence color of each transition of the net, $f : T \times C \rightarrow \mathbf{R}^+$, where \mathbf{R}^+ denotes the set of nonnegative real numbers.

The concept of states and derivations of state space for timed colored nets is given in Z90].

3. THROUGHPUT IN TIMED COLORED NETS

Intuitively, throughput of a place color (p, c) in a timed net \mathbf{T} is equal to the average number of (colored) tokens entering (or leaving) the place p in a unit time; in the steady-state of the net, the average numbers of (colored) tokens entering and leaving each place must be equal since no ‘accumulation’ of tokens can occur. Similarly, throughput of an occurrence color (t, o) is equal to the average number of occurrence’s firings in a unit time. It should be noted that the throughput of an occurrence does not depend upon the number of incoming or outgoing arcs; in the steady-state, the (average) numbers of tokens removed from each of the input places and deposited to each of the output places in a unit time are the same.

More formally, the throughput of a timed net \mathbf{T} is defined as a function $\theta : (P \cup T) \rightarrow C \rightarrow \mathbf{R}^\oplus$ which assigns a nonnegative number to each place color and each occurrence color of the net:

$$\forall(a \in P \cup T) \forall(c \in C) \theta(a, c) = \lim_{n \rightarrow \infty} \frac{n}{\tau_n(a, c)}$$

where $\tau_n(a, c)$ denotes the time instant at which the n -th consecutive token of color c enters (or leaves) the place a or at which the occurrence color (a, c) initiates (or terminates) its n -th firing.

The “non-accumulation” property of the steady-state implies the following properties of throughput:

Property 1: For each place color (p, c) , its throughput is equal to the sum of throughputs of its input occurrence colors as well as the sum of throughputs of its output occurrence colors:

$$\forall((p, c) \in P \times C) \theta(p, c) = \sum_{(t, o) \in \text{Inp}(p, c)} \theta(t, o) = \sum_{(t, o) \in \text{Out}(p, c)} \theta(t, o).$$

Property 2: For each non-shared place color (p, c) , the throughput of (p, c) 's output transition is equal to the throughput of (p, c) :

$$\forall((p, c) \in P \times C) \text{Out}(p, c) = \{(t, o)\} \Rightarrow \theta(t, o) = \theta(p, c).$$

Property 3: For each free-choice place color (p, c) , throughputs of (p, c) 's output transitions are determined by the choice function u :

$$\forall((t, o) \in \text{Out}(p, c)) \theta(t, o) = u(t, o)\theta(p, c).$$

Property 4: For each place color (p, c) that is shared by two occurrences (t_i, o_k) and (t_j, o_ℓ) , and which is guarded by a place p_h :

$$\begin{aligned} p_h \in \text{Inp}(t_i, o_k) &\Rightarrow \theta(t_i, o_k) = \theta(p_h), \\ p_h \in \text{Inh}(t_j, o_\ell) &\Rightarrow \theta(t_j, o_\ell) = \theta(p, c) - \theta(p_k, c). \end{aligned}$$

An elementary colored net is a net in which there is exactly one input place color and exactly one output place color for each occurrence color of the net, and one input occurrence color and one output occurrence color for each place color of the net. In other words, the (directed) graph of an elementary net is a (simple) cycle, and there is no interaction between different colors.

It follows immediately from the property 2 that in elementary nets the throughputs of all transitions and all places are the same for each color.

Property 5: For an elementary net \mathbf{T} :

$$\forall(c \in C) \forall(a \in P \cup T) \theta(a, c) = \frac{\sum_{p \in P} m_0(p, c)}{\sum_{t \in T} f(t, c)}$$

4. EXAMPLE

In Petri net models of service systems (in general sense), places usually represent system queues, transitions servers, directed arcs model the flow of activities in the model as well as synchronization constraints for concurrent activities, colors are used to represent different classes of activities, and arc functions describe priorities of simultaneous events, queueing disciplines, etc.

The timed net shown in Fig.1 is a typical model of a memory-constrained multiprogramming system [L84] in which colors are used to represent different classes of jobs (or customers) with different priorities of acquiring resources of the system. Only two classes of jobs are considered in this example but any generalization is quite straightforward; these two classes of jobs be denoted by colors ‘ H ’ and ‘ L ’, for ‘high-priority’ and ‘low-priority’ jobs, respectively. Moreover, another color, ‘ A ’ is used for representing the numbers of available resource units.

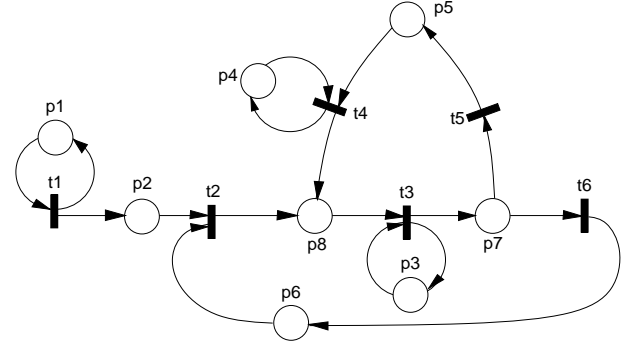


Fig.1. Petri net model of a multiprogramming system.

In Fig.1, p_1 and t_1 model the source of jobs which arrive with the average interarrival times equal to the firing time functions of occurrences ‘ H ’ and ‘ L ’ of t_1 ; the initial marking function typically assigns one ‘ H ’ and one ‘ L ’ token to p_1 . p_2 is the memory queue for jobs that must wait for entering the execution stage; the system limits the number of jobs that receive allocation of memory and can execute concurrently (this limit is also called the level of multiprogramming), so only certain number of jobs can enter into the central server. The memory constraint is represented by the place p_6 (with its initial marking $m_0(p_6, A)$); each job entering the central system reduces the number of tokens in p_6 by one, and each job leaving the system (transition t_6) increases the number of tokens in p_6 . p_3 and t_3 represent the central server with its waiting queue p_8 , while p_4 and t_4 model a disk server with its queue p_5 ; the number of (identical) processors in the central server is determined by the initial marking of the place p_3 , and the number of (identical) disk drives by the initial marking of the place p_4 . The place p_7 is a free-choice place with two choices, termination of job execution (transition t_6 with probability q) or continuation of execution (transition t_5 with probability $(1-q)$). The probability q can be determined on the basis of the average number of disk requests per job execution; if each job requests K disk operations on average, $q = 1/(1 + K)$.

In the colored net from Fig.1, the color ‘ A ’ is effectively used for p_3 , p_4 and p_6 only; all other places and transitions use only colors ‘ H ’ and ‘ L ’. All color distributions are determined by the arc functions w , so $w(p_1, t_1)(H) = \{H\}$, $w(p_1, t_1)(L) = \{L\}$, $w(t_1, p_1)(H) = \{H\}$, $w(t_1, p_1)(L) = \{L\}$, $w(p_6, t_2)(H) = w(p_6, t_2)(L) = \{A\}$, etc. The priorities of the ‘ H ’-class jobs are introduced by the inhibitor functions h which are non-empty only for arcs (p_2, t_2) , (p_8, t_3) and (p_5, t_4) , and for all these arcs $h(p_i, t_j)(H) = \Phi$, while $h(p_i, t_j)(L) = \{H\}$.

It should be observed that the place p_2 is potentially unbounded; it becomes unbounded if there is no limit on the number of t_1 's firings with respect to firings of other transitions.

For $c \in \{H, L\}$, the steady-state throughput relations are as follows:

$$\begin{aligned} \theta(t_1, c) &= \theta(p_1, c) & \theta(p_1, c) &= \theta(t_1, c) \\ \theta(t_2, c) &= \theta(p_2, c) & \theta(p_2, c) &= \theta(t_1, c) \\ \theta(t_3, c) &= \theta(p_8, c) & \theta(p_3, A) &= \theta(t_3, H) + \theta(t_3, L) \\ \theta(t_4, c) &= \theta(p_5, c) & \theta(p_4, A) &= \theta(t_4, H) + \theta(t_4, L) \\ \theta(t_5, c) &= (1 - q_c) \theta(p_7, c) & \theta(p_5, c) &= \theta(t_5, c) \\ \theta(t_6, c) &= q_c \theta(p_7, c) & \theta(p_6, A) &= \theta(t_6, H) + \theta(t_6, L) \\ & & \theta(p_7, c) &= \theta(t_3, c) \\ & & \theta(p_8, c) &= \theta(t_2, c) + \theta(t_4, c) \end{aligned}$$

Because of the relations:

$$\begin{aligned} \theta(t_1, c) &= \theta(p_1, c) \\ \theta(p_1, c) &= \theta(t_1, c) \end{aligned}$$

p_1 and t_1 determine a simple subnet, so (property 5):

$$\begin{aligned}\theta(t_1, H) &= \theta(p_1, H) = m_0(p_1, H)/f(t_1, H), \\ \theta(t_1, L) &= \theta(p_1, L) = m_0(p_1, L)/f(t_1, L),\end{aligned}$$

and then the remaining (symbolic) throughputs are:

$$\begin{aligned}\theta(t_2, H) &= \theta(t_6, H) = \theta(p_2, H) = m_0(p_1, H)/f(t_1, H), \\ \theta(t_2, L) &= \theta(t_6, L) = \theta(p_2, L) = m_0(p_1, L)/f(t_1, L), \\ \theta(t_3, H) &= \theta(p_7, H) = \theta(p_8, H) = \frac{m_0(p_1, H)}{q_H f(t_1, H)}, \\ \theta(t_3, L) &= \theta(p_7, L) = \theta(p_8, L) = \frac{m_0(p_1, L)}{q_L f(t_1, L)}, \\ \theta(t_4, H) &= \theta(t_5, H) = \theta(p_5, H) = \frac{(1-q_H) m_0(p_1, H)}{q_H f(t_1, H)}, \\ \theta(t_4, L) &= \theta(t_5, L) = \theta(p_5, L) = \frac{(1-q_L) m_0(p_1, L)}{q_L f(t_1, L)}, \\ \theta(p_3, A) &= \frac{m_0(p_1, H)}{q_H f(t_1, H)} + \frac{m_0(p_1, L)}{q_L f(t_1, L)}, \\ \theta(p_4, A) &= \frac{(1-q_H) m_0(p_7, H)}{q_H f(t_7, H)} + \frac{(1-q_L) m_0(p_7, L)}{q_L f(t_7, L)}, \\ \theta(p_6, A) &= \frac{m_0(p_7, H)}{f(t_7, H)} + \frac{m_0(p_7, L)}{f(t_7, L)}\end{aligned}$$

This solution is valid only for the steady-state of the net. The validity of this solution can be verified by checking the utilization factors of occurrence colors. Generally, for each $(t, o) \in T \times C$, the utilization factor $u(t, o)$ is equal to the ratio of the throughput $\theta(t, o)$ to the maximum firing rate, and the maximum firing rates, if they are limited, can be determined from the finite resources of the modeled system. For transitions t_3 and t_4 these maximum firing rates correspond to the maximal utilization of processors, so:

$$\begin{aligned}u(t_2) &= \theta(t_2, H) \frac{f(t_2, H)}{m_0(p_6, A)} + \theta(t_2, L) \frac{f(t_2, L)}{m_0(p_6, A)}, \\ u(t_3) &= \theta(t_3, H) \frac{f(t_3, H)}{m_0(p_3, A)} + \theta(t_3, L) \frac{f(t_3, L)}{m_0(p_3, A)}, \\ u(t_4) &= \theta(t_4, H) \frac{f(t_4, H)}{m_0(p_4, A)} + \theta(t_4, L) \frac{f(t_4, L)}{m_0(p_4, A)}, \\ u(t_6) &= \theta(t_6, H) \frac{f(t_6, H)}{m_0(p_6, A)} + \theta(t_6, L) \frac{f(t_6, L)}{m_0(p_6, A)}\end{aligned}$$

while the remaining transitions have unlimited maximum firing rate.

The values of utilization factors beyond the interval $[0,1]$ indicate that the requirement of steady-state behavior is not satisfied, so the solution is not valid. For example, for $m_0(p_1, H) = m_0(p_1, L) = m_0(p_3) = 1$, $f(t_1, H) = 2$, $f(t_1, L) = 1$, $f(t_3, H) = f(t_3, L) = 0.1$, and $q_H = q_L = 0.01$, $u(t_3) = 15$ which clearly indicates that the net is not ergodic; actually, it indicates that the processor represented by t_3 should be at least 15 times faster to handle the service demand.

Many other results can be derived in a similar way.

5. CONCLUDING REMARKS

It has been shown that, for a class of timed color Petri nets, steady-state throughput analysis can be performed on the basis of structural properties of the net rather than the state space. Moreover, since throughput analysis uses the average values of firing times only, the exact firing time distribution function are not important and they may even be unknown. Also, the approach can be applied to nets which have different distribution functions associated with different transitions of the same nets; for example, some transitions may have deterministic firing times while other transitions may fire in a stochastic manner.

The timed net used in the example corresponds to an open network of the queueing theory [K90]. It is believed that the proposed approach can be extended to (at least) some classes of closed network models.

The proposed approach can be used for analysis of unbounded nets, then however the verification of the ergodicity condition must be performed. As shown in the example, quite often the solution can be ‘verified’ by checking some additional requirements (e.g., utilization factors or other known performance bounds).

An attractive aspect of throughput analysis is the possibility of obtaining the solutions in symbolic form rather than a numerical value. Different combinations of parameter values, sensitivity of the results with respect to different parameters or functional dependencies can be investigated very conveniently when symbolic solutions are available.

Finally, it should be noticed that the proposed throughput analysis can be automated quite easily and can be integrated with more general tools for analysis Petri net models as one of alternative analysis methods.

Acknowledgement

The Natural Sciences and Engineering Research Council of Canada partially supported this research through Operating Grant A8222.

References

- [A84] M. Ajmone Marsan, G. Conte, G. Balbo, “A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems”; ACM Trans. on Computer Systems, vol.2, no.2, pp.93–122, 1984.
- [B76] J.P. Buzen, “Fundamental operational laws of computer system performance”; Acta Informatica, vol.7, no.2, pp.167–182, 1976.
- [D78] P.J. Denning, J.P. Buzen, “The operational analysis of queueing network models”; ACM Computing Surveys, vol.10, no.3, pp.225–261, 1978.
- [J87] K. Jensen, “Coloured Petri nets”; in: “Advanced Course on Petri Nets 1986” (Lecture Notes in Computer Science 254), G. Rozenberg (ed.), pp.248–299, Springer Verlag 1987.
- [K90] P.J.B. King, “Computer and communication systems performance modelling”; Prentice-Hall 1990.
- [L84] E.D. Lazowska, J. Zahorjan, G.S. Graham, K.C. Sevcik, “Quantitative system performance”; Prentice-Hall 1984.
- [M89] T. Murata, “Petri nets: properties, analysis and applications”; Proceedings of IEEE, vol.77, no.4, pp.541–580, 1989.
- [P81] J.L. Peterson, “Petri net theory and the modeling of systems”, Prentice-Hall 1981.
- [R85] W. Reisig, “Petri nets – an introduction”; Springer Verlag 1985.
- [S77] J. Sifakis, “Use of Petri nets for performance evaluation”; in: “Measuring, modelling and evaluating computer systems”, pp.75–93, North-Holland 1977.
- [Z81] W.M. Zuberek, “Timed Petri nets and preliminary performance evaluation”; Proc. 7-th Annual Symp. on Computer Architecture, pp.89–96, 1981.
- [Z90] W.M. Zuberek, “Performance evaluation using timed colored Petri nets”; protocols”; Proc. Midwest Symp. on Circuits and Systems, pp.779–782, 1990.
- [Z91] W.M. Zuberek, “Timed Petri nets – definitions, properties and applications”; Microelectronics and Reliability (Special Issue on Petri Nets and Related Graph Models), vol.31, no.4, pp.627–644, 1991.