Advances in Petri Nets 1985 (Lecture Notes in Computer Science 222), pp.478-498.

DOI 10.1007/BFb0016227.

M–TIMED PETRI NETS, PRIORITIES, PREEMPTIONS, AND PERFORMANCE EVALUATION OF SYSTEMS

W.M. Zuberek

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

Abstract

In M–timed Petri nets, firing times are exponentially distributed random variables associated with transitions of a net. Several classes of M–timed Petri nets are discussed in this paper to show increasing "modelling power" of different nets. Conflict–free nets can model M and E_k–type queueing systems. Free—choice nets can also represent H_k–type systems. Systems with several classes of users and with service priorities assigned to user classes require nets with inhibitor arcs. Preemption of service can be represented by extended nets with escape (or generalized inhibitor) arcs. Finally, to provide flexible modelling of scheduling and decision strategies, enhanced Petri nets are introduced with two classes of transitions, immediate and timed ones, and with (exponentially distributed) firing times associated with the timed transitions. It is shown that the behaviour of bounded M–timed Petri nets can be represented by finite "state" graphs which are finite-state continuous-time homogeneous Markov processes. Stationary probabilities of states can thus be obtained by standard techniques used for analysis of Markov chains, and then operational analysis can be applied for performance evaluation. Simple models of interactive systems are used as an illustration of modelling.

1. INTRODUCTION

A Petri net [1,6,23] is an abstract, formal model of systems with interacting, concurrent or parallel components. Petri nets have been successfully used in modelling, validation and analysis of systems in which it is possible for some events to occur concurrently but there are constraints on the concurrence, precedence, or frequency of these occurrences [6,9,11,14,15,22]. Petri nets, however, are not complete enough for the study of performance issues since no assumption is made on the duration of systems activities. Several different concepts of "timed" Petri nets [5,8,19,24,25,26,27,28] and stochastic Petri nets [3,4,20,21] have been proposed by assigning (deterministic or stochastic) firing and/or enabling times to the transitions and/or places of Petri nets.

The approach described in this paper is a continuation of the approach originated by Ramchandani [24] and used to model the performance of digital systems at the register transfer level [27] and to study communication protocols [25] when fixed (or deterministic) firing times can be used. In M–timed Petri nets, similarly as in stochastic nets [20,21], the firing times are exponentially distributed random variables, and their rates are assigned to transitions of a net. In inhibitor Petri nets, inhibitor arcs [1,2,6,23] are used to represent priorities of simultaneous events. In extended Petri nets, escape (or break) arcs are introduced to interrupt firing transitions and to model preemptions of busy servers. In enhanced Petri nets [29,30], the transitions are partitioned into two classes, the immediate and timed transitions, and the firing rates of exponentially distributed firing times are assigned to timed transitions, as in generalized stochastic nets [3]. However, in (basic and generalized) stochastic nets the firings of transitions are instantaneous events, and the tokens actually remain in the input places of firing transitions. The "state" space is thus conveniently determined by the set

of reachable markings, but multiple simultaneous events cannot be directly represented in such a model, and this introduces some restrictions on modelling of even quite simple systems [28]. Moreover, the stochastic approach cannot be applied to models in which the set of "states" (or discrete behaviour of the model) is only loosely related to the set of reachable markings. In M–timed Petri nets, the tokens are removed from corresponding places at the beginning of transition firings and remain in transitions for the whole period of firing. The "state" describes the distribution of tokens in places as well as transitions, and this removes many restrictions of the stochastic approach. For different classes of M–timed nets, a uniform discrete–state continuous–time description is introduced which represents the behaviour of nets by equivalent continuous–time homogeneous Markov chains that can be generated directly from net specifications. For bounded nets, stationary probabilities of system states can thus be obtained by standard techniques, and this provides many performance measures such as utilization of systems components, average queue lengths, average waiting times, etc.

This paper is organized in 6 main sections. Section 2 contains definitions of general concepts for conflict-free M–timed Petri nets. Free–choice M–timed Petri nets are discussed in section 3. Section 4 introduces inhibitor free–choice M–timed Petri nets, and section 5 extended free–choice M–timed Petri nets. Enhanced free–choice M–timed Petri nets are described in section 6. Several simple models of interactive computer systems are used as an illustration of performance evaluation.

2. CONFLICT–FREE M–TIMED PETRI NETS

A basic Petri net $\mathbf{N}$ is a triple $\mathbf{N} = (P, T, A)$ where:

P is a finite, nonempty set of places,

T is a finite, nonempty set of transitions,

A is a set of directed arcs which connect places with transitions and transitions with places, $A \subset P \times T \cup T \times P$, such that:

$$\forall(t \in T)\ \exists(p_i, p_j \in P)\ (p_i, t) \in A \wedge (t, p_j) \in A.$$

A place p is an input (or an output) place of a transition t iff there exists an arc (p, t) (or (t, p), respectively) in the set A. The sets of all input and output places of a transition t are denoted by $Inp(t)$ and $Out(t)$, respectively. Similarly, the sets of input and output transitions of a place p are denoted by $Inp(p)$ and $Out(p)$. The notation is extended in an obvious way to sets of places and transitions, e.g., $Inp(P_i)$, $Out(T_j)$, etc.

A place p is shared iff it is an input place for more that one transition. A net is conflict-free if it does not contain shared places.

A marked Petri net $\mathbf{M}$ is a pair $\mathbf{M} = (\mathbf{N}, m_0)$ where:

$\mathbf{N}$ is a Petri net, $\mathbf{N} = (P, T, A)$,

m_0 is an initial marking function which assigns a nonnegative number of so called tokens to each place of the net, $m_0 : P \to \{0, 1, ...\}$.

Let any function $m : P \to \{0, 1, ...\}$ be called a marking in a net $\mathbf{N}$.

A transition t is enabled by a marking m iff every input place $p \in Inp(t)$ contains at least one token, $m(p) > 0$. The set of transitions enabled by a marking m is denoted by $T(m)$. Every transition enabled by a marking m can fire. When a transition t fires, a token is removed from each

of t's input places and a token is added to each of t's output places. This determines a new marking in a net, a new set of enabled transitions, and so on.

A marking m_j is directly reachable from a marking m_i in a net $\mathbf{N}$ iff there exists a transition t enabled by the marking m_i, $t \in T(m_i)$, such that:

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

A firing sequence of a marking m is any sequence of transitions $(t_{j_1}, t_{j_2}, ...)$ such that for $k = 1, 2, ...$, and for $m_{i_0} = m$, $t_{j_k} \in T(m_{i_{k-1}})$ and m_{i_k} is directly reachable from $m_{i_{k-1}}$ by firing t_{j_k}.

A marking m_j is reachable from a marking m_i in a net $\mathbf{M}$ iff there exists a firing sequence which transforms m_i into m_j, i.e., if there exists a sequence of markings $(m_{i_0}, m_{i_1}, m_{i_2}, ..., m_{i_n})$ such that $m_{i_0} = m_i$, $m_{i_n} = m_j$, and each marking m_{i_k} is directly reachable from the marking $m_{i_{k-1}}$ for $k = 1, ..., n$.

A set $M(\mathbf{M})$ of reachable markings of a Petri net $\mathbf{M}$ is the set of all markings which are reachable from the initial marking m_0.

A net $\mathbf{M}$ is bounded if there exists a positive integer k such that each marking in the set $M(\mathbf{M})$ assigns at most k tokens to each place of the net

$$\exists(k > 0)\ \forall(m \in M(\mathbf{M}))\ \forall(p \in P)\ m(p) \leq k.$$

If a net $\mathbf{M}$ is bounded, its reachability set $M(\mathbf{M})$ is finite. Only bounded Petri nets are considered in this paper.

A marking graph $\mathbf{G}$ of a marked Petri net $\mathbf{M}$ is a labeled directed graph $\mathbf{G}(\mathbf{M}) = (W, D, u)$ where:

W is a set of vertices which is equal to the set of reachable markings of the net $\mathbf{M}$, $W = M(\mathbf{M})$,

D is a set of directed arcs, $D \subset W \times W$, such that (m_i, m_j) is in D iff m_j is directly reachable from m_i in $\mathbf{M}$,

u is a labeling function which assigns a subset of the set of transitions to each arc (s_i, s_j) in the set D, $u : D \to 2^T$, in such a way that $u(m_i, m_j)$ contains all those transitions $t_{ij} \in T(m_i)$, firing of which transforms m_i into m_j.

In timed Petri nets [24,25,26,27,28], each transition t takes a "real" time to fire. When a transition t is enabled, a firing can be initiated by removing tokens from t's input places. The tokens remain in the transition t for the "firing time", and then the firing terminates by adding tokens to each of t's output places. Each of the firings is initiated in the same instant of time in which it is enabled. If a transition is enabled while it fires, a new, independent firing can be initiated.

The firing times of transitions can be described in several ways. In D–timed Petri nets [8, 24, 25, 26, 27] they are deterministic (or constant), i.e., there is a positive (rational) number assigned to each transition of a net which determines the duration of transition's firing. In M–timed Petri nets [28,29] (or stochastic Petri nets [3,4,20,21]), the firing times are exponentially distributed random variables, and the corresponding firing rates are assigned to transitions of a net. The memoryless property of the exponential distribution is the key factor in analysis of M–timed Petri nets.

A conflict-free M–timed Petri net $\mathbf{T}$ is a pair $\mathbf{T} = (\mathbf{M}, r)$ where:

$\mathbf{M}$ is a conflict-free bounded Petri net, $\mathbf{M} = (\mathbf{N}, m_0)$, and $\mathbf{N} = (P, T, A)$,

r is a firing–rate function which assigns a positive real number $r(t)$ to each transition t of the net, $r : T \to \mathbf{R}^+$, and $\mathbf{R}^+$ denotes the set of positive real numbers; the firing time of a transition t is a random variable $v(t)$ with the distribution function

$$\text{Prob}(v(t) > x) = e^{-x*r(t)}, \quad x > 0.$$

The memoryless property of exponential distributions means that if the duration v of a certain activity (e.g., a firing time) is distributed exponentially with parameter r, and if that activity is observed at time y after its beginning, then the remaining duration of the activity is independent of y and is also distributed exponentially with parameter r:

$$\text{Prob}(v > y + x \mid v > y) = \text{Prob}(v > x) = e^{-x*r}.$$

The exponential distribution is the only continuous distribution with the memoryless property.

A state s of an M–timed Petri net $\mathbf{T}$ is a pair $s = (m, f)$ where:

m is a marking function, $m : P \to \{0, 1, ...\}$,

f is a firing function which indicates (for each transition of the net) the number of active firings, i.e., the number of firings which have been initiated but are not yet terminated (i.e., are "in progress" in the state s), $f : T \to \{0, 1, ...\}$.

The initial state s_1 of a conflict-free net $\mathbf{T}$ is a pair $s_1 = (m_1, f_1)$ where:

$$\forall (t \in T)\ f_1(t) = \min_{p \in Inp(t)} (m_0(p)),$$

$$\forall (p \in P)\ m_1(p) = m_0(p) - \sum_{t \in Out(p)} f_1(t).$$

A state $s_j = (m_j, f_j)$ is directly t_k–reachable from a state $s_i = (m_i, f_i)$ iff the following conditions are satisfied:

(1) $f_i(t_k) > 0$,

(2) $\forall (p \in P)\ m_{ik}(p) = m_i(p) + \begin{cases} 1, & \text{if } p \in Out(t_k), \\ 0, & \text{otherwise,} \end{cases}$

(3) $\forall (t \in T)\ e_\ell(t) = \min_{p \in Inp(t)} (m_{ik}(p))$,

(4) $\forall (p \in P)\ m_j(p) = m_{ik}(p) - \sum_{t \in Out(p)} e_\ell(t)$,

(5) $\forall (t \in T)\ f_j(t) = f_i(t) + e_\ell(t) - \begin{cases} 1, & \text{if } t = t_k, \\ 0, & \text{otherwise.} \end{cases}$

The state s_j which is directly t_k–reachable from the state s_i, is thus obtained by the termination of a t_k firing (1), updating the marking of a net (2), and then initiating new firings (if any) which are determined by the function e_ℓ (3, 4 and 5).

Similarly as for reachable markings, a state s_j is reachable from a state s_i if there is a sequence of directly reachable states from the state s_i to the state s_j. Also, a set $S(\mathbf{T})$ of reachable states is defined as the set of all states of a net $\mathbf{T}$ which are reachable from the initial states. For bounded conflict–free nets the sets of reachable states are finite.

A state graph $\mathbf{G}$ of an M–timed Petri net $\mathbf{T}$ is a labeled directed graph $\mathbf{G}(\mathbf{T}) = (V, D, u)$ where:

V is a set of vertices which is equal to the set of reachable states of the net $\mathbf{T}$, $V = S(\mathbf{T})$,

D is a set of directed arcs, $D \subset V \times V$, such that (s_i, s_j) is in D iff s_j is directly reachable from s_i in $\mathbf{T}$,

u is a transition–rate function which assigns the rate of transitions from s_i to s_j to each arc (s_i, s_j) in the set D, $u : D \to \mathbf{R}^+$, in such a way that if s_j is directly t_k–reachable from $s_i = (m_i, f_i)$, then

$$u(s_i, s_j) = r(t_k) * f_i(t_k).$$

It should be noticed that state graphs of conflict–free bounded timed Petri nets are finite continuous–time homogeneous Markov chains [13,16]. The stationary (or equilibrium) probabilities $x(s)$ of the states $s \in S(\mathbf{T})$ are thus obtained from the state–transition rates by solving a system of simultaneous linear equations [12,16]

$$\begin{cases} \sum\limits_{1 \le j \le N} u(s_j, s_i) * x(s_j) = x(s_i) \sum\limits_{1 \le j \le N} u(s_i, s_j); \quad i = 1, ..., N-1 \\ \sum\limits_{1 \le i \le N} x(s_i) = 1 \end{cases}$$

where N is the number of states in the set $S(\mathbf{T})$.

Example. For a simple conflict–free M–timed Petri net $\mathbf{T}_1$ shown in Fig.1a (as usual, places are represented by circles, transitions by bars, the initial marking is represented by dots inside circles, and the firing rate functions is given as an additional description of transitions), the derivation of the set $S(\mathbf{T}_1)$ of reachable states is shown in Tab.1 which also contains the stationary probabilities of the states, $x(s)$, $s \in S(\mathbf{T}_1)$.

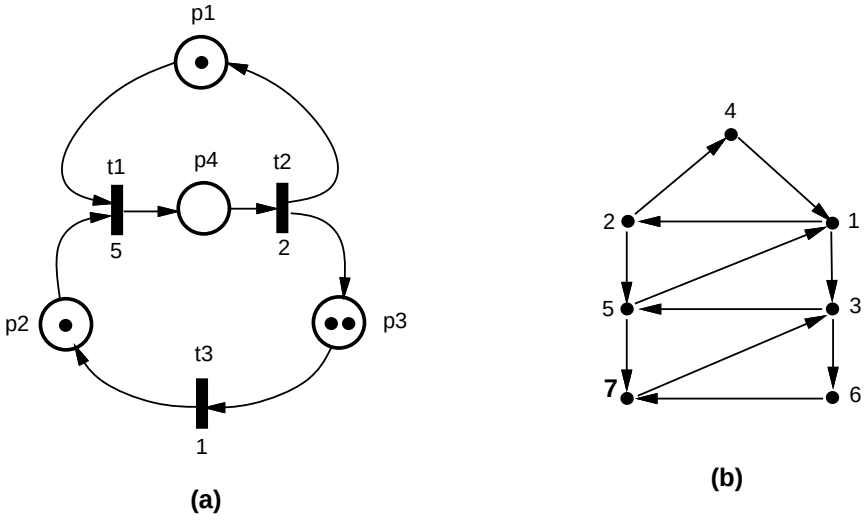

Fig.1. Conflict-free M–timed net $\mathbf{T}_1$ (a) and its state graph (b).

The Petri net from Fig.1a is a model of a simple interactive computer system with a finite population of users (or terminals — transition t_3) and with a central server composed of two consecutive stages (transitions t_1 and t_2). The place p_2 models the queue of waiting jobs, and the place p_1 controls the number of server's channels (the marking of p_1 models the number of idle channels which can be equal to 0 or 1 in this example). The number of users (or terminals) in the system

s_i	$x(s_i)$	m_i 1 2 3 4	f_i 1 2 3	t_k	e_ℓ 1 2 3	s_j	$u(s_i, s_j)$
1	0.127	0 0 0 0	1 0 2	1	0 1 0	2	5.0
				3	0 0 0	3	2.0
2	0.158	0 0 0 0	0 1 2	2	0 0 1	4	2.0
				3	0 0 0	5	2.0
3	0.108	0 1 0 0	1 0 1	1	0 1 0	5	5.0
				3	0 0 0	6	1.0
4	0.105	1 0 0 0	0 0 3	3	1 0 0	1	3.0
5	0.285	0 1 0 0	0 1 1	2	1 0 1	1	2.0
				3	0 0 0	7	1.0
6	0.022	0 2 0 0	1 0 0	1	0 1 0	7	5.0
7	0.196	0 2 0 0	0 1 0	2	1 0 1	3	2.0

Tab.1. The set of reachable states for $\mathbf{T}_1$.

is represented by the total number of initial tokens assigned to places p_2, p_3 and p_4. All service times are assumed to be exponentially distributed, and the corresponding service (or firing) rates are equal to $r(t_1)$=5, $r(t_2)$=2, and $r(t_3)$=1. The net $\mathbf{T}_1$ is thus a model of a queueing system described in Kendall's notation as $M/E_2/1//3$ (in this notation $x/y/m/\ell/n$ describes a central server system with x–distributed interarrival times, y–distributed service times, m channels of the central server, with capacity of the system determined by ℓ (default ∞) and with n sources; for x and y parameters, M denotes exponential distribution, E_k Erlang k–stage distribution, and H_k hyperexponential k-stage distribution [13]).

Performance analysis is based on stationary probabilities $x(s)$ of the states $s \in S(\mathbf{T}_1)$, and on general laws of operational analysis [7,10,18]. Since the system is idle only in the state s_4 ($m_4(p_1) = 1$, i.e., the only channel of the server is idle), the stationary probability that the system is idle is equal to $x(s_4) = 0.105$, and then the utilization of the system is equal to $1 - x(s_4) = 0.895$. Since the average service time is equal to $(1/r(t_1) + 1/r(t_2)) = 0.7$ time units, the average throughput rate of this system is equal to 0.895/0.7=1.279 jobs per time unit, and the average turnaround time (which can be determined using the response time law [10,18]) is equal to $3/1.279 = 2.346$ time units. Since the average terminal time is equal to 1 time unit, the average waiting time is equal to $2.346 - 1.0 - 0.7 = 0.646$ time units.

If the system has two identical service channels (i.e., for initial marking $m_0 = [2, 1, 2, 0]$ instead of $m_0 = [1, 1, 2, 0]$), the set $S(\mathbf{T})$ contains 9 states, and the same performance measures are as follows:

the equilibrium probability that the system is idle	0.194
the utilization of the system	0.806
the equilibrium probability that 1 channel is busy	0.414
the equilibrium probability that 2 channels are busy	0.392
the average throughput rate	1.711
the average turnaround time	1.753
the average waiting time	0.053

□

3. FREE-CHOICE M–TIMED PETRI NETS

Conflict–free Petri nets cannot represent systems in which some events are random and their (nondeterministic) occurrences are described by corresponding probabilities. Free–choice Petri nets [6,15] provide a simple random "choice" mechanism which can be used in such cases.

A shared place p is free–choice (or extended free–choice [6]) iff the input sets of all transitions sharing p are identical. A Petri net $\mathbf{N}$ is free–choice iff all its shared places are free–choice.

In free–choice nets, free–choice places imply an equivalence relation in the set of transitions T, and determine a partition of T into a set of free–choice equivalence classes of transitions denoted by $Free(T) = \{T_1, T_2, ..., T_k\}$. In a free–choice marked net $\mathbf{M}$, for each marking m and for each "free–choice" equivalence class T_i, either all transitions of T_i are enabled simultaneously, or none of them. Moreover, free–choice nets are nonpersistent [6], i.e., the firing of one of enabled transitions may disable other enabled transitions (in the same free–choice equivalence class). Therefore, it is convenient to define all possibilities of transition firings as a function of an actual marking. The set of selection functions is introduced to describe all such possibilities.

A selection function of a marking m in a free–choice net $\mathbf{N}$ is any such function $e : T \to \{0, 1, ...\}$ that:

(1) there exists a sequence of transitions $w = (t_{i_1}, t_{i_2}, ..., t_{i_k})$ in which $t_{i_j} \in T(m_{i_{j-1}})$ for $j = 1, ...k$ and for $m_{i_0} = m$, where

$$\forall (p \in P)\ m_{i_j}(p) = m_{i_{j-1}}(p) - \begin{cases} 1, & \text{if } p \in Inp(t_{i_j}), \\ 0, & \text{otherwise,} \end{cases}$$

(2) the set of transitions enabled by m_{i_k}, $T(m_{i_k})$, is empty,

(3) for each transition $t \in T$, the number of occurrences of t in the sequence w is equal to $e(t)$.

The set of all selection functions of a marking m is denoted by $Sel(m)$.

It is assumed that the "choices" within free–choice equivalence classes are random and can be described by a "choice function" $c : T \to [0, 1]$ in such a way that for each equivalence class $T_i \in Free(T)$, the sum of probabilities $c(t)$ is equal to 1. Then, the probability associated with a selection function $e \in Sel(m)$ is equal to

$$\prod_{T_i \in Free(T)} a(T_i, e) \prod_{t \in T_i} c(t)^{e(t)}$$

where the coefficient $a(T_i, e),\ i = 1, ...k$, describes the number of different ways in which the (random) choices can be made within the free–choice class T_i. It can be determined as follows. Let an n–argument function z be defined recursively:

(1) $z(0, 0, ..., 0) = 1$,

(2) $z(k_1, k_2, ..., k_n) = \sum_{1 \le i \le n} \begin{cases} z(k_1, ..., k_{i-1}, k_i - 1, k_{i+1}, ..., k_n), & \text{if } k_i > 0, \\ 0, & \text{if } k_i = 0. \end{cases}$

Then, for the class $T_i = \{t_{i_1}, t_{i_2}, ..., t_{i_n}\}$:

$$a(T_i, e) = z(e(t_{i_1}), e(t_{i_2}), ..., e(t_{i_n})),$$

and, for any marking m:

$$\sum_{e \in Sel(m)} \prod_{T_i \in Free(T)} a(T_i, e) \prod_{t \in T_i} c(t)^e(t) = 1.$$

In fact, the values of $a(T_i, e)$ are coefficients of a multinomial expansion when the variables correspond to the transitions in T_i, and the exponents of variables are determined by the selection function e.

i	e_i 1 2 3	$a(T_j, e_i)$	$\prod_{t \in T_j} c(t)^{e_i(t)}$
1	2 0 0	1	q_1^2
2	0 2 0	1	q_2^2
3	0 0 2	1	q_3^2
4	1 1 0	2	$q_1 q_2$
5	1 0 1	2	$q_1 q_3$
6	0 1 1	2	$q_2 q_3$

Tab.2. The set of selection functions and their probabilities.

Example. For a free–choice place p with a free–choice class of transitions $Out(p) = T_j = \{t_1, t_2, t_3\}$, let the probabilities of "free choices" be equal to $c(t_1) = q_1$, $c(t_2) = q_2$ and $c(t_3) = q_3$. For a marking m such that $m(p) = 2$, there are 6 selection functions (with respect to t_1, t_2 and t_3) shown in Tab.2 together with corresponding probabilities. Since $q_1 + q_2 + q_3 = 1$, the sum of all selection probabilities is equal to $(q_1 + q_2 + q_3)^2 = 1$. □

A free–choice M–timed Petri net $\mathbf{T}$ is a triple $\mathbf{T} = (\mathbf{M}, c, r)$ where:

$\mathbf{M}$ is a free–choice marked Petri net, $\mathbf{M} = (\mathbf{N}, m_0)$, and $\mathbf{N} = (P, T, A)$,

c is a choice function which assigns a "free–choice" probability to each transition of a net in such a way that:

$$\forall (T_i \in Free(T)) \sum_{t \in T_i} c(t) = 1,$$

r is a firing–rate function, $r : T \to \mathbf{R}^+$.

A state s of a free–choice M–timed Petri net $\mathbf{T}$ is a pair $s = (m, f)$, as before. An initial state s_i of a net $\mathbf{T}$ is a pair $s_i = (m_i, f_i)$ where f_i is a selection function from the set $Sel(m_0)$, $f_i \in Sel(m_0)$, and the marking m_i is determined by

$$\forall (p \in P)\ m_i(p) = m_0(p) - \sum_{t \in Out(p)} f_i(t).$$

A free–choice M–timed net $\mathbf{T}$ may have several different initial states.

A state $s_j = (m_j, f_j)$ is directly (t_k, e_ℓ)–reachable from a state $s_i = (m_i, f_i)$ iff the following conditions are satisfied:

(1) $f_i(t_k) > 0$,

(2) $\forall (p \in P)\ m_{ik}(p) = m_i(p) + \begin{cases} 1, & \text{if } p \in Out(t_k), \\ 0, & \text{otherwise,} \end{cases}$

(3) $e_\ell \in Sel(m_{ik})$,

(4) $\forall (p \in P)\ m_j(p) = m_{ik}(p) - \sum_{t \in Out(p)} e_\ell(t)$,

(5) $\forall (t \in T)\ f_j(t) = f_i(t) + e_\ell(t) - \begin{cases} 1, & \text{if } t = t_k, \\ 0, & \text{otherwise.} \end{cases}$

The state s_j which is directly (t_k, e_ℓ)–reachable from the state s_i, is thus obtained by the termination of a t_k firing (1), updating the marking of a net (2), and then initiating new firings (if any) which are determined by a selection function e_ℓ from the set $Sel(m_{ik})$ (3, 4 and 5).

A state graph $\mathbf{G}$ of a free–choice M–timed Petri net $\mathbf{T}$ is a labeled directed graph $\mathbf{G}(\mathbf{T}) = (V, D, u)$ where:

V is a set of vertices which is equal to the set of reachable states of the net $\mathbf{T}$, $V = S(\mathbf{T})$,

D is a set of directed arcs, $D \subset V \times V$, such that (s_i, s_j) is in D iff s_j is directly reachable from s_i in $\mathbf{T}$,

u is a transition–rate function which assigns the rate of transitions from s_i to s_j to each arc (s_i, s_j) in the set D, $u : D \to \mathbf{R}^+$, in such a way that if s_j is directly (t_k, e_ℓ)–reachable from $s_i = (m_i, f_i)$, then

$$u(s_i, s_j) = r(t_k) * f_i(t_k) \prod_{T_n \in Free(T)} a(T_n, e_\ell) \prod_{t \in T_n} c(t)^{e^\ell(t)}.$$

Similarly as for conflict-free nets, the state graphs of free–choice bounded M–timed Petri nets are finite continuous–time homogeneous Markov chains, and the stationary probabilities $x(s)$ of the states $s \in S(\mathbf{T})$ are obtained from the state-transition rates in exactly the same way.

Example. A simple free–choice M–timed Petri net $\mathbf{T}_2$ shown in Fig.2a (the choice function is given as an additional description of transitions) contains two free–choice places, p_1 and p_2. The derivation of the set $S(\mathbf{T}_2)$ of reachable states is given in Tab.3 which also shows the stationary probabilities of the states, $x(s)$.

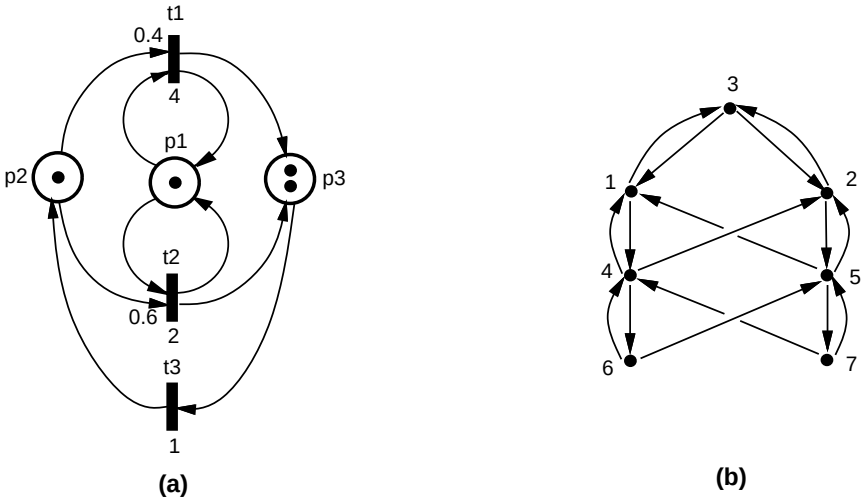

Fig.2. Free–choice M–timed net $\mathbf{T}_2$ (a) and its state graph (b).

The Petri net from Fig.2a is a model of another interactive computer system with a finite population of users (t_3 models the "thinking" or terminal times) who submit two types of jobs, "short" jobs (with probability 0.4) which require 0.25 time units of processor time on average, and "long" ones (with probability 0.6) requiring on average 0.5 time units of processor time. The central server is thus composed of two alternate stages (transitions t_1 and t_2) which are selected with probabilities 0.4 and 0.6, respectively. The place p_2 models the queue of waiting jobs, while the place p_1 controls the number of server's channels, as before. The number of terminals in the system is represented by the total number of initial tokens assigned to places p_2 and p_3. All service times are assumed to

s_i	$x(s_i)$	$m(s_i)$ 1 2 3	$f(s_i)$ 1 2 3	t_k	1 2 3	s_j	$u(s_i, s_j)$
1	0.228	0 0 0	0 1 2	2	0 0 1	3	2.0
				3	0 0 0	4	2.0
2	0.101	0 0 0	1 0 2	1	0 0 1	3	4.0
				3	0 0 0	5	2.0
3	0.287	1 0 0	0 0 3	3	0 1 0	1	1.8
					1 0 0	2	1.2
4	0.205	0 1 0	0 1 1	2	0 1 1	1	1.2
					1 0 1	2	0.8
				3	0 0 0	6	1.0
5	0.062	0 1 0	1 0 1	1	0 1 1	1	2.4
					1 0 1	2	1.6
				3	0 0 0	7	1.0
6	0.103	0 2 0	0 1 0	2	0 1 1	4	1.2
					1 0 1	5	0.8
7	0.015	0 2 0	1 0 0	1	0 1 1	4	2.4
					1 0 1	5	1.6

Tab.3. The set of reachable states for $\mathbf{T}_2$.

be exponentially distributed, and the corresponding service (or firing) rates are equal to $r(t_1) = 4$, $r(t_2) = 2$, and $r(t_3) = 1$. The net $\mathbf{T}_2$ is thus a model of an $M/H_2/1//3$ queueing system.

Performance evaluation uses (again) the stationary probabilities of the states. The stationary probability that the system is idle is equal to $x(s_3) = 0.287$ ($m_3(p_1) = 1$, i.e., the server is idle), and then the utilization of the system is equal to $1-x(s_3) = 0.713$. Since the average service time is equal to $(c(t_1)/r(t_1) + c(t_2)/r(t_2)) = 0.4$ time units, the average throughput rate of this system is equal to $0.713/0.4 = 1.783$ jobs per time unit, and the average turnaround time is equal to 3/1.783=1.683 time units. Since the average terminal time is equal to 1 time unit, the average waiting time is equal to $1.683 - 1.0 - 0.4 = 0.283$ time units.

If the system has two identical (two-stage) service channels (i.e., for initial marking $m_0 = [2, 1, 2]$ instead of $m_0 = [1, 1, 2]$), the set $S(\mathbf{T})$ contains 9 states, and the same performance measures are as follows:

the equilibrium probability that the system is idle	0.361
the utilization of the system	0.639
the equilibrium probability that 1 channel is busy	0.432
the equilibrium probability that 2 channels are busy	0.209
the average throughput rate	2.115
the average turnaround time	1.418
the average waiting time	0.018

□

4. INHIBITOR M–TIMED PETRI NETS

It is well–known [2,17,23] that Petri nets without inhibitor arcs (or an equivalent extension) cannot represent priorities of simultaneous events. The inhibitor arcs [1,2,23] provide a "test if zero" condition which appears to be a convenient mechanism for different priority schemes.

An inhibitor Petri net $\mathbf{N}$ is a quadruple $\mathbf{N} = (P, T, A, B)$ where:

(P, T, A) is a basic Petri net,

B is a set of inhibitor arcs, $B \subset P \times T$, and A and B are disjoint sets.

A place p is an inhibitor place of a transition t iff there exists an inhibitor arc (p, t) in the set B. The set of all inhibitor places of a transition t is denoted by $Inh(t)$.

A marked inhibitor Petri net $\mathbf{M}$ is defined as a pair $\mathbf{M} = (\mathbf{N}, m_0)$, similarly as before.

In inhibitor nets, a transition t is enabled by a marking m iff every input place of this transition contains at least one token and every inhibitor place of t contains zero tokens. The set of all transitions enabled by a marking m is denoted by $T(m)$, as before.

A shared place p is guarded iff for each two different transitions t_i and t_j sharing p there is another place p_k such that p_k is in the input set of one and in the inhibitor set of another of these two transitions

$$\forall(t_i \in Out(p))\ \forall(t_j \in Out(p) - \{t_i\})\ \exists(p_k \in P - \{p\})$$
$$((p_k, t_i) \in A \wedge (p_k, t_j) \in B)\ \vee\ ((p_k, t_j) \in A \wedge (p_k, t_i) \in B),$$

i.e., no two transitions from the set $Out(p)$ can be enabled by the same marking m. An inhibitor net is conflict-free iff all its shared places are guarded.

A shared place p is free–choice (or extended free–choice [6]) iff the input sets and inhibitor sets of all transitions sharing p are identical, i.e., iff

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

An inhibitor net is free–choice iff all its shared places are either guarded or free–choice.

It should be noticed that in inhibitor M–timed Petri nets, there is no direct relationship between the number of reachable markings and the number of reachable states. For many M–timed nets with finite sets of reachable states, the underlying ordinary nets (i.e., the nets "without" time) are unbounded (an example is shown in Fig.3a), and there are also bounded (ordinary) nets which (augmented by a firing rate function) have infinite sets of reachable states (an example is shown in Fig.3b).

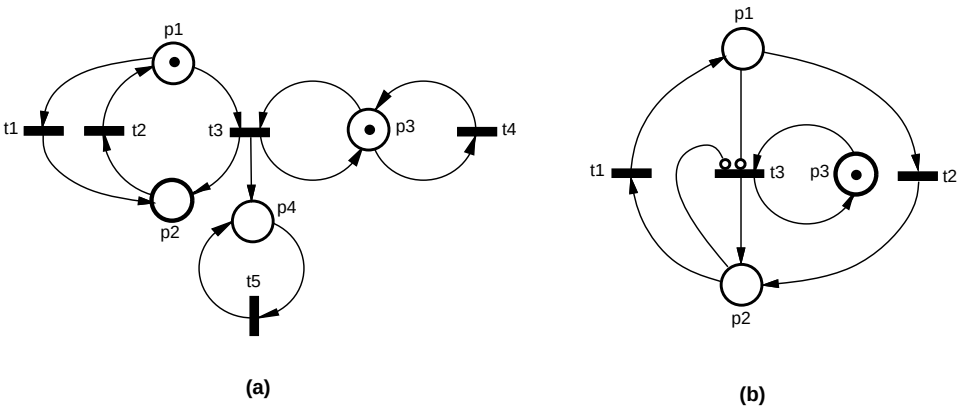

Fig.3. Bounded M–timed net $\mathbf{T}$ with unbounded ordinary net $\mathbf{M}$ (a) and unbounded M–timed net $\mathbf{T}$ with bounded ordinary net $\mathbf{M}$ (b).

An inhibitor M–timed Petri net $\mathbf{T}$ is bounded if there exists a positive integer k such that for each state $s = (m, f)$ from the set of reachable states, $s \in S(\mathbf{T})$, the number of tokens assigned to

each place $p \in P$, $m(p)$, and the number of active firings for each transition $t \in T$, $f(t)$, are not greater than k. Only bounded M–timed Petri nets are discussed in this paper.

All the remaining concepts for inhibitor nets are defined in the same way as for free–choice nets with the only correction that instead of "free–choice" property, "inhibitor free–choice" should be used.

Example. An inhibitor free–choice M–timed Petri net $\mathbf{T}_3$ shown in Fig.4a (inhibitor arcs have small circles instead of arrowheads) contains one guarded place (p_1). The derivation of the set $S(\mathbf{T}_3)$ of reachable states is given in Tab.4 which also shows the stationary probabilities of the states, $x(s)$.

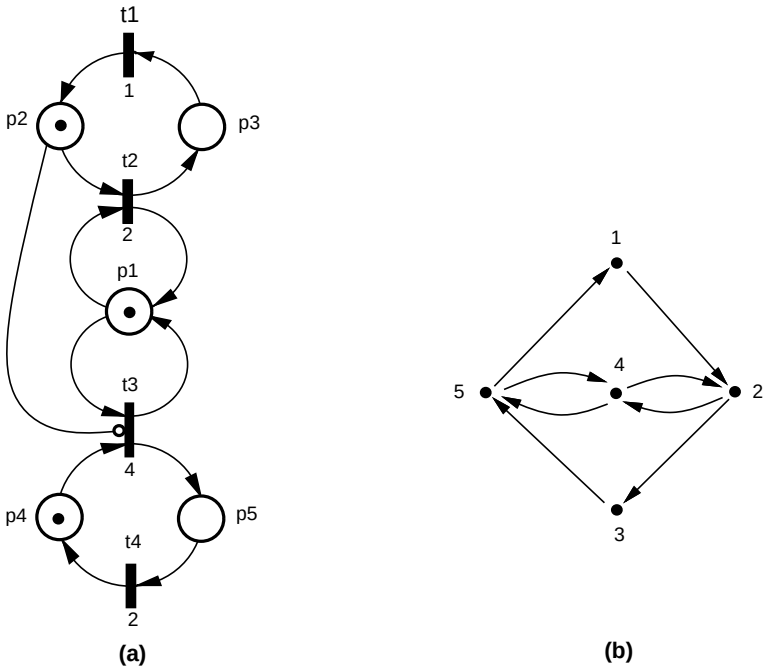

Fig.4. Inhibitor M–timed Petri net $\mathbf{T}_3$ (a) and its state graph (b).

		m_i	f_i		e_ℓ		
s_i	$x(s_i)$	1 2 3 4 5	1 2 3 4	t_k	1 2 3 4	s_j	$u(s_i, s_j)$
1	0.157	0 0 0 1 0	0 1 0 0	2	1 0 1 0	2	2.0
2	0.225	0 0 0 0 0	1 0 1 0	1	0 0 0 0	3	1.0
				3	0 0 0 1	4	4.0
3	0.056	0 1 0 0 0	0 0 1 0	3	0 1 0 1	5	4.0
4	0.404	1 0 0 0 0	1 0 0 1	1	0 1 0 0	5	1.0
				4	0 0 1 0	2	2.0
5	0.157	0 0 0 0 0	0 1 0 1	2	1 0 0 0	4	2.0
				4	0 0 0 0	1	2.0

Tab.4. The set of reachable states for $\mathbf{T}_3$.

The M–timed Petri net $\mathbf{T}_3$ from Fig.4a is a simple model of an interactive system with 2 classes of users (and jobs) and nonpreemptive priority scheduling. The system contains one central server (p_1, t_2 and t_3) with two queues of waiting jobs (p_2 and p_4) for class–1 and class–2 jobs, respectively, n_1 terminals in class 1 and n_2 terminals in class 2. All class–1 jobs have "higher" priority than the class–2 ones, i.e., if class–1 and class–2 jobs are waiting for the central server, the class–1 jobs will

receive the service first (the inhibitor arc (p_2, t_3) disables t_3 if p_2 contains at least one token). For $n_1 = n_2 = 1$ there are 5 states of the system, as shown in Tab.4.

Since there are two classes of jobs, performance analysis can be done with respect to each class as well as for the whole system. The stationary probability that the system is idle is equal to $x(s_4) = 0.404$, and then the utilization of the system is equal to 0.596 which is composed of 0.314 for class–1 jobs ($x(s_1) + x(s_5)$ since $f_1(t_2) = f_5(t_2) = 1$) and 0.282 for class–2 jobs ($x(s_2) + x(s_3)$). The throughput rates are equal to 0.314/0.5=0.628 for class–1 and 0.282/0.25=1.128 for class–2 jobs; the average throughput rate for the whole system is thus 0.628+1.128=1.756 jobs per time unit. The average turnaround times are equal to 1.592 and 0.887 for class–1 and class–2 jobs, respectively, and then the average waiting times are equal to 0.092 and 0.137 time units for class–1 and class–2, respectively.

The differences between these two classes of jobs are more significant for increased "traffic" in the system. For $n_1 = n_2 = 3$, the same performance measures are as follows:

the equilibrium probability that the system is idle	0.017
the utilization of the system	0.983
the average throughput rate	2.470
the class-1 utilization of the system	0.731
the average class-1 throughput rate	1.462
the average class-1 turnaround time	2.052
the average class-1 waiting time	0.552
the class-2 utilization of the system	0.252
the average class-2 throughput rate	1.008
the average class-2 turnaround time	2.976
the average class-2 waiting time	2.226

□

5. EXTENDED M–TIMED PETRI NETS

Inhibitor arcs provide a flexible mechanism for modelling priorities of simultaneous events, however, the priorities are taken into account only during initiating new firings. In order to allow "dynamic" priorities of events, yet another mechanism must be introduced to "interrupt" firing transitions in order to preempt the resources used by lower priority activities. The escape arcs, an extension of inhibitor arcs, are used for this purpose.

An extended Petri net $\mathbf{N}$ is a 5–tuple $\mathbf{N} = (P, T, A, B, C)$ where:

(P, T, A, B) is an inhibitor Petri net,

C is a set of escape arcs which is a subset of the set of inhibitor arcs, $C \subseteq B$.

A place p is an escape place of a transition t iff $(p, t) \in C$. The set of all escape places of t is denoted by $Esc(t)$, and the set of transitions connected by escape arcs with p by $Esc(p)$, $Esc(p) = \{t \mid p \in Esc(t)\}$. The notation is generalized in usual way for sets of places and transitions.

In Petri nets without time, escape arcs are equivalent to inhibitor arcs. In extended M–timed Petri nets, escape arcs may "interrupt" firing transitions. If, during a firing period of a transition t, at least one of t's escape places becomes nonempty (i.e., a token is deposited to it), the firing of t ceases and the tokens removed from t's input places at the beginning of firing, are returned to their original places.

An extended Petri net $\mathbf{N} = (P, T, A, B, C)$ is simple iff input sets of transitions with nonempty escape sets do not contain escape places, i.e.

$$\forall(t \in T)\ Esc(t) = \emptyset\ \vee\ Esc(Inp(t)) = \emptyset$$

where $\emptyset$ denotes the empty set.

Simple nets do not allow "propagation" of interrupts (which simplifies the description of state transitions). Nonsimple extended nets must take into account sequences of "indirect" interrupts when an interrupted transition, through its input places, interrupts other transitions.

For extended nets, the definition of an M–timed Petri net, a state and an initial state is the same as for inhibitor nets (with "extended" property replacing "inhibitor" one), but the description of state transitions requires another refinement.

In extended M–timed Petri nets, a state $s_j = (m_j, f_j)$ is directly (t_k, e_ℓ)–reachable from a state $s_i = (m_i, f_i)$ iff the following conditions are satisfied:

(1) $f_i(t_k) > 0$,

(2) $\forall(p \in P)\ m_{ik}(p) = m_i(p) + \begin{cases} 1, & \text{if } p \in Out(t_k), \\ 0, & \text{otherwise,} \end{cases}$

(3) $\forall(t \in T)\ f_{ik}(t) = f_i(t) - \begin{cases} 1, & \text{if } t = t_k, \\ 0, & \text{otherwise,} \end{cases}$

(4) $\forall(t \in T)\ d_n(t) = \min(f_{ik}(t), \sum\limits_{p \in Esc(t)} m_{ik}(p))$

(5) $\forall(p \in P)\ m_{ikn}(p) = m_{ik}(p) + \sum\limits_{t \in Out(p)} d_n(t)$,

(6) $e_\ell \in Sel(m_{ikn})$,

(7) $\forall(p \in P)\ m_j(p) = m_{ikn}(p) - \sum\limits_{t \in Out(p)} e_\ell(t)$,

(8) $\forall(t \in T)\ f_j(t) = f_{ik}(t) - d_n(t) + e_\ell(t)$.

The state s_j which is directly (t_k, e_ℓ)–reachable from the state s_i, is thus obtained by the termination of a t_k firing (1), updating the marking and firing of a net (2, 3), performing all escapes of the firing transitions (if any) as indicated by the function d_n which transforms the marking m_{ik} into m_{ikn} (4 and 5), and then initiating new firings (if any) which are determined by a selection function e_ℓ from the set $Sel(m_{ikn})$ (5, 7 and 8).

Example. A simple extended free–choice M–timed Petri net $\mathbf{T}_4$ shown in Fig.5a (escape arcs are marked by dots instead of arrowheads) is a simple modification of the net $\mathbf{T}_3$ from Fig.4a; the inhibitor arc (p_2, t_3) is replaced by the escape arc which changes a nonpreemptive priority scheduling of jobs into a preemptive one. The derivation of the set $S(\mathbf{T}_4)$ and the stationary probabilities of the states are shown in Tab.5.

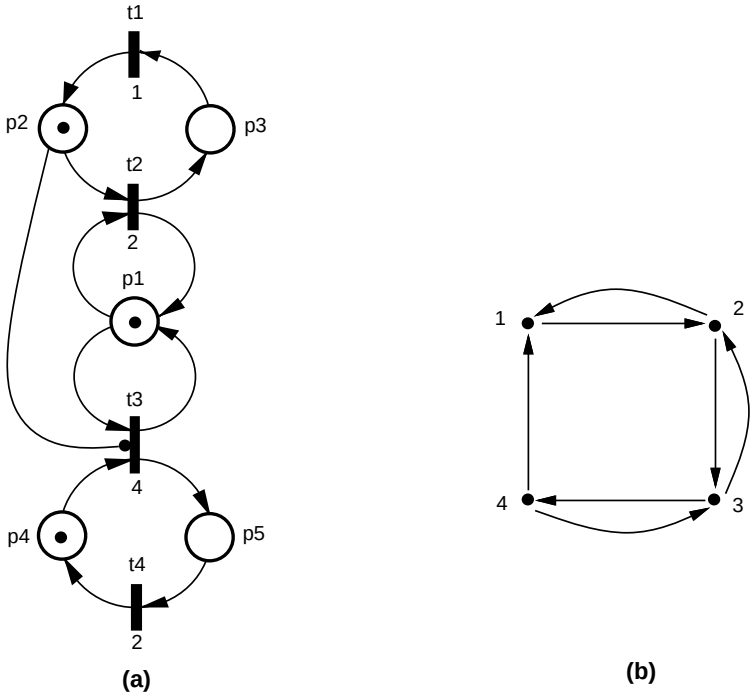

Fig.5. Extended M–timed Petri net $\mathbf{T}_4$ (a) and its state graph (b).

s_i	$x(s_i$	m_i 1 2 3 4 5	f_i 1 2 3 4	t_k	e_ℓ 1 2 3 4	s_j	$u)s_i, s_j)$
1	0.231	0 0 0 1 0	0 1 0 0	2	1 0 1 0	2	2.0
2	0.256	0 0 0 0 0	1 0 1 0	1	0 1 0 0	1	1.0
				3	0 0 0 1	3	4.0
3	0.410	1 0 0 0 0	1 0 0 1	1	0 1 0 0	4	1.0
				4	0 0 1 0	2	2.0
4	0.103	0 0 0 0 0	0 1 0 1	2	1 0 0 0	3	2.0
				4	0 0 0 0	1	2.0

Tab.5. The set of reachable states for $\mathbf{T}_4$.

The same performance indices as in the previous section, for $n_1 = n_2 = 1$ and $n_1 = n_2 = 3$, are as follows:

	$n_1 = n_2 = 1$	$n_1 = n_2 = 3$
the equilibrium probability that the system is idle	0.410	0.018
the utilization of the system	0.590	0.982
the average throughput rate	1.691	2.348
the class-1 utilization of the system	0.334	0.790
the average class-1 throughput rate	0.667	1.580
the average class-1 turnaround time	1.500	1.899
the average class-1 waiting time	0.000	0.399
the class-2 utilization of the system	0.256	0.192
the average class-2 throughput rate	1.024	0.768
the average class-2 turnaround time	0.977	3.906
the average class-2 waiting time	0.227	3.156

□

6. ENHANCED M–TIMED PETRI NETS

Enhanced Petri nets combine two different classes of Petri nets, "immediate" nets which are in fact ordinary (i.e., without time) inhibitor free–choice "straight" Petri nets, and "timed" nets which are extended simple free–choice bounded M–timed Petri nets. The resulting nets are more powerful than other classes of M–timed Petri nets since the only restriction on the complexity of conditions associated with timed transitions is that the "immediate" subnets modelling them must be "straight", i.e., must guarantee finite firing sequences (to provide effective evaluations).

A net $\mathbf{N}_i = (P_i, T_i, A_i, B_i, C_i)$ is a T_i–implied subnet of an extended net $\mathbf{N} = (P, T, A, B, C)$ iff:

(1) $T_i \subseteq T$,

(2) $A_i = A \cap (P \times T_i \cup T_i \times P)$,

(3) $B_i = B \cap (P \times T_i)$,

(4) $C_i = C \cap (P \times T_i)$, and

(5) $P_i = Inp(T_i) \cup Out(T_i) \cup Inh(T_i) \cup Esc(T_i)$.

Also, a net $\mathbf{N}$ is straight iff for all initial markings m_0, the marking graph $\mathbf{G}(\mathbf{M}, m_0)$ is acyclic. In straight nets all firing sequences of (finite) markings are finite. If a graph of a net $\mathbf{N} = (P, T, A, B, C)$ does not contain cycles, i.e., if a directed graph $(P \cup T, A)$ is acyclic, the net is obviously straight, but many cyclic nets are also straight.

An enhanced Petri net $\mathbf{H}$ is a 6-tuple $\mathbf{H} = (P, T_t, T_0, A, B, C)$ where:

$(P, T_t \cup T_0, A, B, C)$ is an extended free–choice Petri net,

T_t is a set of timed transitions,

T_0 is a set of immediate transitions such that the T_0–implied subnet of $(P, T_t \cup T_0, A, B, C)$ is straight, the sets T_t and T_0 are disjoint, and

$$\forall(T_i \in Free(T_t \cup T_0))\ T_i \subseteq T_t \vee T_i \subseteq T_0,$$

i.e., each free–choice equivalence class of transitions must belong either to the timed or to the immediate set of transitions.

The set of all transitions is denoted by T, $T = T_t \cup T_0$.

An enhanced marked Petri net $\mathbf{M}$ is a pair $\mathbf{M} = (\mathbf{H}, m_0)$, as before. Similarly, the set of transitions enabled by a marking m is denoted by $T(m)$. Moreover, $T_0(m)$ denotes the set of immediate transitions enabled by m, and $T_t(m)$ the set of timed transitions enabled by m.

An enhance function of a marking m in a net $\mathbf{M}$ is any function $h : T \to \{0, 1, ...\}$ such that:

(1) there exists a finite (possibly empty) firing sequence of immediate transitions $v = (t_{i_1}, t_{i_2}, ..., t_{i_k})$ which transforms the marking m into a marking m_k, and the set of immediate transitions enabled by m_k, $T_0(m_k)$, is empty, and

(2) for each immediate transition $t - in T_0$ the number of occurrences of t in the sequence v is equal to $h(t)$, while for each timed transition $t \in T_t$, $h(t) = 0$.

The set of all enhance functions of a marking m is denoted by $Enh(m)$. Since the immediate subnet (i.e., the T_0–implied subnet) is straight, for each marking m the set $Enh(m)$ is finite.

In enhanced nets, selection functions are defined similarly as in extended (inhibitor or free–choice) nets but with respect to timed transitions only. A selection function of a marking m in an enhanced net $\mathbf{H}$ is thus any such function $e : T \to \{0, 1, ...\}$ that:

(1) there exists a sequence of timed transitions $w = (t_{i_1}, t_{i_2}, ..., t_{i_k})$ in which $t_{i_j} \in T_t(m_{i_{j-1}})$ for $j = 1, ..., k$ and for $m_{i_0} = m$, where:

$$\forall(p \in P)\ m_{i_j}(p) = m_{i_{j-1}}(p) - \begin{cases} 1, & \text{if } p \in Inp(t_{i_j}), \\ 0, & \text{otherwise,} \end{cases}$$

(2) the set of timed transitions enabled by m_{i_k}, $T_t(m_{i_k})$, is empty,

(3) for each timed transition $t \in T_t$ the number of occurrences of t in the sequence w is equal to $e(t)$, while for each immediate transition $t \in T_0$, $e(t) = 0$.

It should be observed that there are two basic differences between enhance and selection functions:

(1) enhance functions are (in fact) defined for immediate transitions while selection functions for timed transitions,

(2) enhance functions describe sequences of (complete) firings while selection functions indicate initiations of (timed) firings.

An enhanced timed Petri net $\mathbf{T}$ is a triple $\mathbf{T} = (\mathbf{M}, c, r)$,

$\mathbf{M}$ is a marked enhanced simple free–choice bounded Petri net, $\mathbf{M} = (\mathbf{H}, m_0)$, and $\mathbf{H} = (P, T_t, T_0, A, B, C)$,

c is a choice function, $c : T_t \cup T_0 \to [0, 1]$ such that

$$\forall(T_i \in Free(T)) \sum_{t \in T_i} c(t) = 1,$$

r is a firing–rate function which assigns firing rates to timed transitions of the net, $r : T_t \to \mathbf{R}^+$.

A state s of an enhanced M–timed Petri net $\mathbf{T}$ is defined as a pair $s = (m, f)$, as before. An initial state s_i of a net $\mathbf{T}$ is a pair $s_i = (m_i, f_i)$ where f_i is a selection function from the set $Sel(m_{ij})$, $f_i \in Sel(m_{ij})$, the marking m_{ij} is determined by an enhance function h_j from the set $Enh(m_0)$, $h_j \in Enh(m_0)$,

$$\forall(p \in P)\ m_{ij}(p) = m_0(p) - \sum_{t \in Out(p)} h_j(t) + \sum_{t \in Inp(p)} h_j(t),$$

and the marking m_i is defined as

$$\forall(p \in P)\ m_i(p) = m_{ij}(p) - \sum_{t \in Out(p)} f_i(t).$$

An enhanced free–choice net $\mathbf{T}$ may have several different initial states.

A state $s_j = (m_j, f_j)$ is directly (t_k, h_ℓ, e_n)–reachable from a state $s_i = (m_i, f_i)$ iff the following conditions are satisfied:

(1) $f_i(t_k) > 0$,

(2) $\forall(p \in P)\ m_{ik}(p) = m_i(p) + \begin{cases} 1, & \text{if } p \in Out(t_k), \\ 0, & \text{otherwise,} \end{cases}$

(3) $\forall(t \in T)\ f_{ik}(t) = f_i(t) - \begin{cases} 1, & \text{if } t = t_k, \\ 0, & \text{otherwise,} \end{cases}$

(4) $h_\ell \in Enh(m_{ik})$,

(5) $\forall(p \in P)\ m_{ik\ell}(p) = m_{ik}(p) - \sum_{t \in Out(p)} h_\ell(t) + \sum_{t \in Inp(p)} h_\ell(t)$,

(6) $\forall(t \in T)\ d_{ik\ell}(t) = \begin{cases} \min(f_{ik}(t), \sum_{p \in Esc(t)} m_{ik\ell}(p), & \text{if } t \in T_t, \\ 0, & \text{otherwise,} \end{cases}$

(7) $\forall(p \in P)\ m_{ik\ell j}(p) = m_{ik\ell}(p) + \sum_{t \in Out(p)} d_{ik\ell}(t)$,

(8) $e_n \in Sel(m_{ik\ell j})$,

(9) $\forall(p \in P)\ m_j(p) = m_{ik\ell j}(p) - \sum_{t \in Out(p)} e_n(t)$,

(10) $\forall(t \in T)\ f_j(t) = f_{ik}(t) - d_{ik\ell}(t) + e_n(t)$.

The state s_j which is directly (t_k, h_ℓ, e_n)–reachable from the state s_i, is thus obtained by the termination of a t_k firing (1), updating the marking and firing functions (2, 3), performing all immediate firings (if any) which correspond to an enhance function h_ℓ from the set $Enh(m_{ik})$, and which results in an intermediate marking $m_{ik\ell}$ (4 and 5), then performing all escapes of the firing timed transitions (if any) as indicated by the function $d_{ik\ell}$, which transforms the marking $m_{ik\ell}$ into $m_{ik\ell j}$ (6 and 7), and finaly initiating new timed firings (if any) which are determined by a selection function e_n from the set $Sel(m_{ik\ell j})$ (8, 9 and 10).

A state graph $\mathbf{G}$ of an enhanced M–timed Petri net $\mathbf{T}$ is a labeled directed graph $\mathbf{G}(\mathbf{T}) = (V, D, u)$ where:

V is a set of vertices which is equal to the set of reachable states of the net $\mathbf{T}$, $V = S(\mathbf{T})$,

D is a set of directed arcs, $D \subset V \times V$, such that (s_i, s_j) is in D iff s_j is directly reachable from s_i in $\mathbf{T}$,

u is a transition–rate function, $u : D \to \mathbf{R}^+$, such that if s_j is directly (t_k, h_ℓ, e_n)–reachable from $s_i = (m_i, f_i)$, then:

$$u(s_i, s_j) = r(t_k) * f_i(t_k) \prod_{T_g \in Free(T)} a(T_g, h_\ell + e_n) \prod_{t \in T_g} c(t)^{h_\ell(t) + e_n(t)}.$$

Example. A simple enhanced free–choice M–timed Petri net $\mathbf{T}_5$ shown in Fig.6a (timed transitions are represented by solid bars while immediate transitions by bars) contains one guarded place (p_1) and one free–choice place (p_6). The immediate subnet is implied by t_6 and t_7, and it includes p_1, p_2, p_4, p_5, p_6 and p_7 with all arcs connecting these places with t_6 and t_7; the subnet is obviously acyclic, so it is straight. The derivation of the set $S(\mathbf{T}_5)$ of reachable states and the stationary probabilities of the states are shown in Tab.6.

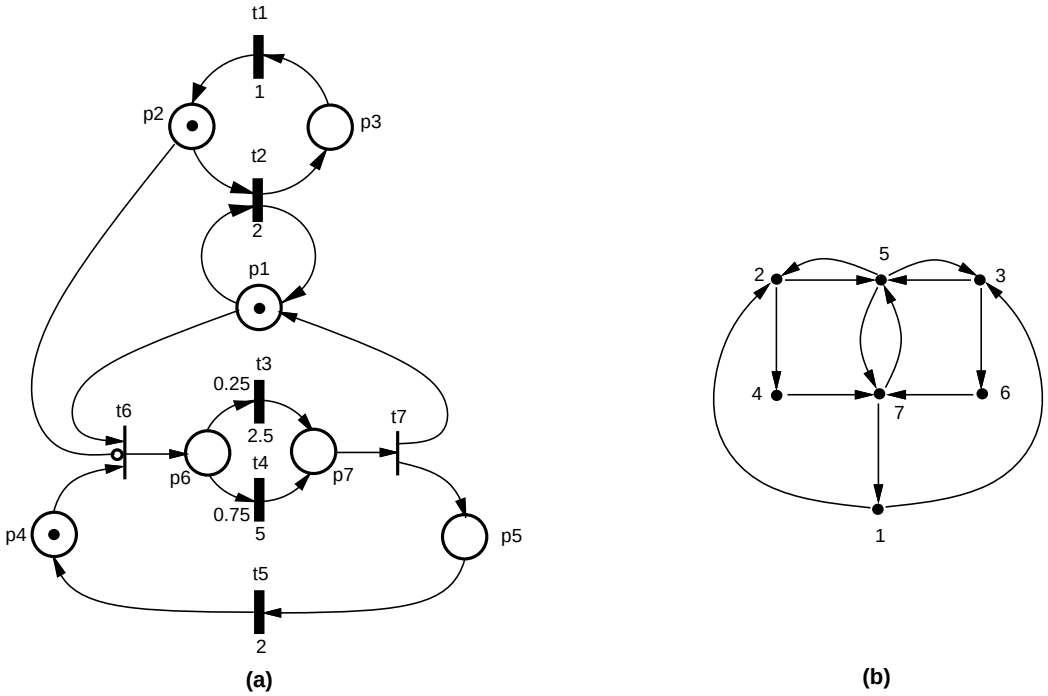

Fig.6. Enhanced M–timed Petri net $\mathbf{T}_5$ (a) and its state graph (b).

		m_i	f_i		h_ℓ	e_n		
s_i	$x(s_i)$	1 2 3 4 5 6 7	1 2 3 4 5	t_k	6 7	1 2 3 4 5	s_j	$u(s_i, s_j)$
1	0.157	0 0 0 1 0 0 0	0 1 0 0 0	2	1 0	1 0 0 1 0	2	1.5
						1 0 1 0 0	3	0.5
2	0.141	0 0 0 0 0 0 0	1 0 0 1 0	1	0 0	0 0 0 0 0	4	1.0
				4	0 1	0 0 0 0 1	5	5.0
3	0.080	0 0 0 0 0 0 0	1 0 1 0 0	1	0 0	0 0 0 0 0	6	1.0
				3	0 1	0 0 0 0 1	5	2.5
4	0.028	0 1 0 0 0 0 0	0 0 0 1 0	4	0 1	0 1 0 0 1	7	5.0
5	0.406	1 0 0 0 0 0 0	1 0 0 0 1	1	0 0	0 1 0 0 0	7	1.0
				5	1 0	0 0 0 1 0	2	1.5
						0 0 1 0 0	3	0.5
6	0.032	0 1 0 0 0 0 0	0 0 1 0 0	3	0 1	0 1 0 0 1	7	2.5
7	0.157	0 0 0 0 0 0 0	0 1 0 0 1	2	0 0	1 0 0 0 0	5	2.0
				5	0 0	0 0 0 0 0	1	2.0

Tab.6. The set of reachable states for $\mathbf{T}_5$.

The timed Petri net $\mathbf{T}_5$ is also a model of an interactive system with 2 classes of users and a nonpreemptive priority scheduling (compare with $\mathbf{T}_3$) but there are two types of class–2 jobs, "short" (requiring on average 0.2 time units of central processor time) and "long" ones (with the average service time equal to 0.4 time units), with probabilities 0.75 and 0.25, respectively, while all class–1 jobs are statistically identical. Central processor is thus modelled by three timed transitions (t_2, t_3 and t_4) with firing rates equal to 2, 2.5 and 5 for class–1, long class–2 and short class–2 jobs, respectively; the remaining two timed transitions model class–1 (t_1) and class–2 (t_5) terminals.

The average service time for class–2 jobs is equal to $(c(t_3)/r(t_3) + c(t_4)/r(t_4) = 0.25$ time units, and it is equal to the average service time for class–2 jobs in $\mathbf{T}_3$. Consequently, it can be verified

that the performance results for the net $\mathbf{T}_5$ are almost the same as for the net $\mathbf{T}_3$. □

Many other results can be derived in a similar way.

7. CONCLUDING REMARKS

The widespread acceptance that Petri nets have gained in modelling of queueing systems is due to very natural representation of concurrency, synchronization and precedence of events. Places represent systems queues, transitions servers, directed arcs model flow of jobs (or transactions), and inhibitor as well as escape arcs are used to indicate priorities and to model preemptions. It has been shown that the class of finite M, E_k and H_k–type queueing systems with priority and preemptive scheduling disciplines can be represented by enhanced free–choice bounded M–timed Petri nets. In fact, for each finite–state continuous–time Markovian queueing system there exists an enhanced simple free–choice bounded M–timed Petri net with the state space isomorphic to that of the queueing system [30]. These two classes of models are thus equivalent.

The M–timed Petri net models are usually much simpler than the state representations (or chains) of the modelled Markovian systems, and Petri net parameters correspond directly to components or activities of the modelled systems (e.g., the number of terminals, the number of processors, etc.). The state space can be automatically generated from model specifications, and also many "standard" performance evaluations can be built into software tools for analysis of timed Petri nets.

In M–timed Petri nets, the states describe the distribution of tokens in places as well as in (firing) transitions. In general case, there is no direct relationship between the set of reachable markings and the set of reachable states (Fig.3a and b). It also means that the class of M–timed nets properly contains the class of stochastic (and generalized stochastic) models [3,4,20,21], in which state graphs are isomorphic images of graphs of reachable markings. In fact, it is quite straightforward to restrict an M–timed Petri net in such a way that its state graph is isomorphic to the graph of reachable markings, and then to "follow" the stochastic approach. On the other hand, the stochastic approach, based on the set of reachable markings, cannot represent an infinite state space when the set of reachable markings is finite (Fig.3b).

The most general class of M–timed Petri nets discussed in this paper, i.e., the class of enhanced M–timed Petri nets, is restricted in several ways (simple free–choice bounded nets), some of the restrictions, however, can be removed easily by appropriate extensions of the formalism. In particular, nets with more general conflicts (or "random switches" [3]) can be described in a very similar way provided the probabilities of conflicting firings are known and included in the state description, and nonsimple (extended) nets with sequences of possible escapes, can be conveniently handled by "escape" functions defined similarly to enhance functions. Also, since the immediate subnets can model all the "decisions" required for firing the timed transitions, the timed subnets can be restricted to very simple forms, and this can simplify the (general) description of enhanced nets. For example, in [29] an additional requirement was introduced that each of timed transitions can have only a single input and a single output place, and that input places of timed transitions cannot be shared; it can be shown that this restriction does not change the "modelling power" of enhanced M–timed Petri nets.

Acknowledgements

The author is grateful to his three anonymous referees for their interesting and helpful remarks, and he acknowledges a number of very accurate corrections and valuable improvements indicated by two of them. The fourth referee report is appreciated for its brevity.

REFERENCES

1. T. Agerwala, "Putting Petri nets to work"; IEEE Computer Magazine, vol.12, no.12, pp.85–94, 1979.

2. T. Agerwala, M. Flynn, "Comments on capabilities and 'correctness' of Petri nets"; Proc. 1–st Annual Symp. on Computer Architecture, Gainesville FL, pp.81–86, 1973.

3. 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.

4. B. Bayaert, G. Florin, P. Lonc, S. Natkin, "Evaluation of computer systems dependability using stochastic Petri nets"; Proc. 11—th Annual Int. Symp. on Fault–Tolerant Computing, Portland, MA, pp.79–81, 1981.

5. B. Berthomieu, M. Menasche, "An enumerative approach for analyzing time Petri nets"; Information Processing 83, R.E.A. Mason (ed.), pp.41–45, IFIP 1983.

6. W. Brauer (ed.), "Net theory and applications"; Proc. of the Advanced Course on General Net Theory of Processes and Systems, Hamburg 1979 (Lecture Notes in Computer Science 84); Springer Verlag 1980.

7. J.P. Buzen, "Fundamental operational laws of computer system performance"; Acta Informatica, vol.7, no.2, pp.167–182, 1976.

8. J. Carlier, P. Chretienne, C. Girault, "Modelling scheduling problems with timed Petri nets"; in: "Advances in Petri Nets 1984*" (Lecture Notes in Computer Science 188); G. Rozenberg (ed.), pp.62–82, Springer Verlag 1985.

9. A. Danthine, "Modeling and verification of end-to-end protocols"; in: "New Advances in Distributed Computer Systems" (Proc. NATO Advanced Study Institute, Bonas, France, June 1982), K.G. Beauchamp (ed.), pp.125–158, Reidel Publ. Co. 1982.

10. P.J. Denning, J.P. Buzen, "The operational analysis of queueing network models"; Computing Survey, vol.10, no.3, pp.225–261, 1978.

11. M. Diaz, "Modeling and analysis of communication and cooperation protocols using Petri net based models"; Computer Networks, vol.6, no.6, pp.419–441, 1982.

12. D. Ferrari, "Computer systems performance evaluation"; Prentice–Hall 1978.

13. E. Gelenbe, I. Mitrani, "Analysis and synthesis of computer systems"; Academic Press 1980.

14. C. Girault, W. Reisig (eds.), "Application and theory of Petri nets" (Informatik-Fachberichte 52); Springer Verlag 1982.

15. M. Hack, "Analysis of production schemata by Petri nets"; Project MAC Technical Report TR–94, Massachusetts Institute of Technology, Cambridge MA, 1972.

16. L. Kleinrock, "Queueing systems", vol.1: "Theory", vol.2: "Computer applications"; J. Wiley & Sons 1975, 1976.

17. S.R. Kosaraju, "Limitations of Dijkstra's semaphore primitives and Petri nets"; Operating Systems Review, vol.7, no.4, pp.122–126, 1973.

18. E.D. Lazowska, J. Zahorjan, G.S. Graham, K.C. Sevcik, "Quantitative system performance - computer system analysis using queueing network models"; Prentice–Hall 1984.

19. P.M. Merlin, D.J. Farber, "Recoverability of communication protocols - implications of a theoretical study"; IEEE Trans. on Communications, vol.24, no.9, pp.1036–1049, 1976.

20. M.K. Molloy, "Performance analysis using stochastic Petri nets"; IEEE Trans. on Computers, vol.31, no.9, pp.913–917, 1982.

21. S. Natkin, "Timed and stochastic Petri nets: from the validation to the performance of synchronization schemes"; Proc. Int. Workshop on Timed Petri Nets, Torino, Italy, pp.2–3, 1985.

22. A. Pagnoni, G. Rozenberg (eds.), "Applications and theory of Petri nets" (Informatik-Fachberichte 66); Springer Verlag 1983.

23. J.L. Peterson, "Petri net theory and the modeling of systems", Prentice–Hall 1981.

24. C. Ramchandani, "Analysis of asynchronous concurrent systems by timed Petri nets"; Project MAC Technical Report TR–120, Massachusetts Institute of Technology, Cambridge, MA, 1974.

25. R.R. Razouk, "The derivation of performance expressions for communication protocols from timed Petri nets"; Computer Communication Review, vol.14, no.2, pp.210–217, 1984.

26. J. Sifakis, "Use of Petri nets for performance evaluation"; in: "Measuring, modelling and evaluating computer systems", pp.75–93, North-Holland 1977.

27. W.M. Zuberek, "Timed Petri nets and preliminary performance evaluation"; Proc. IEEE 7–th Annual Symp. on Computer Architecture, La Baule, France, pp.88–96, 1980.

28. W.M. Zuberek, "Performance evaluation using extended timed Petri nets"; Proc. Int. Workshop on Timed Petri Nets, Torino, Italy, pp.37–42, 1985.

29. W.M. Zuberek, "Modelling and performance evaluation of systems using enhanced timed Petri nets"; Proc. 12–th IASTED Conf. on Applied Simulation and Modelling, Montreal, Canada, pp.154–157, 1985.

30. W.M. Zuberek, "M–timed Petri nets and Markov chains in modelling of computer systems"; Proc. ACM 14–th Annual Computer Science Conf., Cincinnati, OH, 1986.