Modeling and Verification of Time Dependent Systems Using Time Petri Nets

Bernard Berthomieu and Michel Diaz, *Member, IEEE*

***Abstract*—This paper deals with describing and analyzing concurrent systems, such as communication systems, whose behavior is dependent on explicit values of time. An enumerative method is proposed in order to exhaustively validate the behavior of Merlin's time Petri net models. This new method, which allows one to formally verify time dependent systems, is applied to a simple illustrative example, the specification and verification of the alternating bit protocol.**

***Index Terms*—Modeling, parallel and distributed systems, specification, time, time Petri nets, verification.**

I. Introduction: Time Petri Nets

THIS paper provides a method for analyzing concurrent systems whose behavior is based on explicit temporal parameters. Examples of such systems are communication protocols, for most of them highly depend on time for reliability or performance aspects. Using and extending classical Petri nets or place-transition nets by adding time features is of high interest in many fields, for instance to account for actual values of time-outs [12], [13].

Two basic Petri net based models for handling time have been developed in the last decade, time Petri nets [9], [10] and timed Petri nets [11].

Ramchandani's timed Petri nets [11] are derived from Petri nets by associating a firing finite duration with each transition of the net. The classical firing rule of Petri nets is modified first to account for the time it takes to fire a transition and second to express that a transition must fire as soon as it is enabled. These nets and related models have been used mainly for performance evaluation.

Merlin's time Petri nets (or TPN's for short) are more general than timed Petri nets: a timed Petri net can be modeled by using a TPN, but the converse is not true. TPN's have been proved very convenient for expressing most of the temporal constraints while some of these constraints were difficult to express only in terms of firing durations.

Merlin defined time Petri nets [9], [10] as Petri nets with labels: two values of time, two real numbers, a and b, with $a \leq b$, are associated with each transition. Assuming that any transition, e.g., t_i, is being continuously enabled after it has been enabled,

- a $(0 \leq a)$, is the minimal time that must elapse, starting from the time at which transition t_i is enabled, until this transition can fire, and
- b $(0 \leq b \leq \infty)$, denotes the maximum time during which transition t_i can be enabled without being fired.

Times a and b, for transition t_i, are relative to the moment at which transition t_i is enabled. Assuming that transition t_i has been enabled at time τ, then t_i, even if it is continuously enabled, cannot fire before time $\tau + a$ and must fire before or at time $\tau + b$, unless it is disabled before its firing by the firing of another transition. Using these nets, Merlin discussed some recoverability problems in computer systems and in communication protocols.

The main purpose of this paper is to propose for time Petri nets an enumerative analysis technique which allows one to simultaneously model the behavior and analyze the properties of timed systems. This technique is related to the reachability analysis method for usual Petri nets [7] and is presented in what follows. Other work has been developed using timed nets; [15], [25], and [29] build a reachability graph, but the model they use has firing durations and not firing intervals as time Petri nets have. Furthermore, the method given here is more general, in the sense that state classes will be defined as a set of firing constraints, and not as given firing values. Some more detailed statements and other examples may be found in [8], [3], and [4].

II. Definition and Behavior

This section gives a formal definition of time Petri nets [9], [10] and introduces a notion of state in order to discuss behavioral analysis problems for TPN's.

It will appear that the state of a TPN will be represented as a pair $S = (M, I)$ where M is a marking and I is a vector of pairs of values, the vector of all possible firing intervals representing the firing domain, defined as the product set of the firing intervals of the enabled transitions. The entries of I have one pair of time values per transition and entry i in this vector represents the interval of all possible firing times related to the transition ordered as the ith of the enabled transitions.

A. Time Petri Nets

A time Petri net is a tuple (P, T, B, F, Mo, SIM) where:

- P is a finite nonempty set of places p_i;
- T is a finite nonempty, set of transitions t_i; it will appear in the sequel that it may be convenient to view it as an ordered set $\{t_1, t_2, \cdots t_i, \cdots\}$;
- B is the backward incidence function

$$B : T \times P \rightarrow N$$

where N is the set of nonnegative integers;

- F is the forward Incidence function

$$F : T \times P \rightarrow N;$$

- Mo is the Initial Marking function

$$Mo : P \rightarrow N$$

(P, T, B, F and Mo together define a Petri net);

Manuscript received August 3, 1988; revised November 1, 1990. Recommended by T. Murata.

The authors are with the Laboratoire d'Automatique et d'Analyse des Systemes, Centre National de la Recherche Scientifique, 7, Avenue du Colonel Roche, 31077 Toulouse Cedex, France.

IEEE Log Number 9041663.

- SIM is a mapping called static interval

$$SIM : T \rightarrow Q^* \times (Q^* \cup \infty)$$

where Q^* is the set of positive rational numbers.

As it will be seen, analyzing TPN's needs to differentiate static intervals and dynamic intervals associated with transitions. Merlin's times are defined here as constrained static rational values satisfying the following constraints for each transition t_i:

$$SIM(t_i) = (\alpha^S, \beta^S),$$

where α^S and β^S are rationals such that

$$0 \leq \alpha^S < \infty, \qquad 0 \leq \beta^S \leq \infty,$$
$$\alpha^S \leq \beta^S \quad \text{if } \beta^S \neq \infty \quad \text{or}$$
$$\alpha^S < \beta^S \quad \text{if } \beta^S = \infty.$$

Let us assume that $SIM(t_i) = (\alpha_i^S, \beta_i^S)$ for some transition t_i. Then:

- the interval of numbers (α_i^S, β_i^S) will be called the static firing interval of transition t_i;
- the left bound α_i^S will be called the static earliest firing time (static EFT for short);
- the right bound β_i^S will be called the static latest firing time (static LFT for short).

α_i^S and β_i^S represent the Static EFT and LFT of transition t_i.

It will appear that for states other than the initial state, firing intervals in the firing domain will in the general case be different from the static firing intervals; their lower bounds will be called EFT and their upper bounds LFT, written as α_i and β_i, respectively.

Times α_i^S and β_i^S, and times α_i and β_i, for transition t_i are relative to the moment at which transition t_i is enabled. Assuming that transition t_i is enabled at an absolute time τ_{Abs}, then t_i may not fire, while being continuously enabled, before time $(\tau_{\text{Abs}} + \alpha_i^S)$ or $(\tau_{\text{Abs}} + \alpha_i)$ and must fire before or at the latest at time $(\tau_{\text{Abs}} + \beta_i^S)$ or $(\tau_{\text{Abs}} + \beta_i)$.

Note that if transition t_i ends to be continuously enabled, this means that another transition t_m has been fired; firing t_m leads to a new marking, at a different absolute time $\tau_{\text{Abs}'}$.

Time can be either discrete or dense: both cases are covered by the method given in this paper. In this model, firing a transition, and this is of importance, takes no time to complete: firing a transition at time τ leads to a new state defined at the same time τ. When writing specifications, this means that, as it is not accounted for, the time needed to fire a transition is equal to 0. Thus the actual duration of the firing of any transition represented in the model either must be null or must have an actual value which is negligible with respect to all related time values defining the time Petri net. If this firing time is of importance and cannot be neglected, then its value must appear as a time label associated with at least one timed transition.

Furthermore, if a pair (α_i^S, β_i^S) is not defined, then it is implicitly assumed that the corresponding transition is a classical Petri net transition and so has the pair

$$(\alpha_i^S = 0, \beta_i^S = \infty)$$

associated with it: nicely and importantly, TPN's are timed restrictions of Petri nets.

It must be noticed that this definition slightly differs from Merlin's one and in particular Q is restricted to be the set of rationals: numbers α_i^S and β_i^S associated with the transitions are rational numbers, while they were real numbers in [10] and [11]. The reason for this restriction, which in practice induces no limitation, will appear in Section IV where properties of TPN's are investigated.

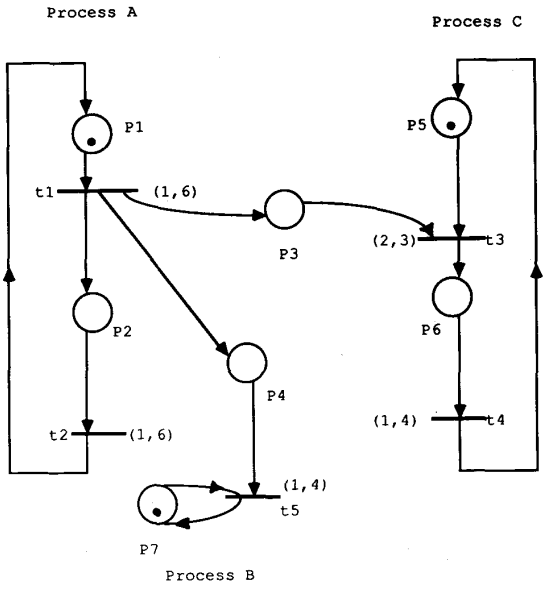


Fig. 1. A time Petri net.

B. States in a TPN

Let us first suppose that the TPN's under consideration are such that none of the transitions may become enabled more than once "simultaneously"; for any marking M, it holds for any enabled transition t_i: $(\exists\, p)(M(p) < 2 \cdot B(t_i, p))$, i.e., there is at least one place which prevents t_i to be firable twice. This nonessential restriction is for simplification and will be discussed at the end of the section.

A general form for a state S of a TPN can be defined as a pair $S = (M, I)$ consisting of:

- a marking M;
- a firing interval set I which is a vector of possible firing times. The number of entries in this vector is given by the number of the transitions enabled by marking M. Furthermore, as vectors I have one entry for each transition enabled by a given marking, the number of the entries of I will vary during the behavior of the net according to the number of transitions enabled by the current marking. Let us assume that the enabled transitions are ordered in I; then entry i in I corresponds to the transition ordered as the ith in the set of transitions enabled by M. This entry i defines a pair of time values, the minimum and maximum times between which this ith transition is, individually, allowed to fire.

Let us consider the TPN given in Fig. 1. Process A sends two messages consumed by processes B and C; process C has two distinct actions (receiving and computing), and process B receives and consumes in an indivisible action. The corresponding times are attached to the transitions.

The initial state of the TPN given in Fig. 1 has $M = p_1$ (1), p_5 (1), p_7 (1): three places are marked and contain one token. As

transition t_1 is the only transition enabled, I has one entry equal to $(1, 6)$ which means that t_1 can fire at any time between 1 and 6.

It follows that firing possibilities may be expressed as the product set of the firing intervals of the enabled transitions, with intervals appearing in the product in the same order as enabled transitions appear in the ordered vector I.

C. Enabledness and Firability Condition of a Set of Transitions

Let us assume the current state of the TPN be $S = (M, I)$. Some transitions may be enabled by marking M, but not all of them may be allowed to fire due to the firing constraints of transitions (EFT's and LFT's).

Let us assume that transition t_i becomes enabled, which means that it is enabled in the usual Petri net sense at time τ in state $S = (M, I)$.

The "firability condition" is formally expressed by two conditions, 1) and 2) which follow, where

1) is the current enabledness condition for Petri nets and
2) expresses the fact that an enabled transition may not fire before its EFT and must fire before or at its LFT unless another fires before and modifies marking M and so state S.

Accordingly, transition t_i is firable from state $S = (M, I)$ at time $\tau + \theta$ iff both the following conditions hold:

1) t_i is enabled by marking M at time τ:

$$(\forall\, p)(M(p) \geq B(t_i, p))\,;$$

2) the relative firing time θ, relative to the absolute enabling time τ, is not smaller than the EFT of transition t_i and not greater than the smallest of the LFT's of all the transitions enabled by marking M:

$$\text{EFT of } t_i \leq \theta \leq \min\{\text{LFT of } t_k\}$$

where k ranges over the set of transitions enabled by M.

Note that 2) holds because, at time $\theta = \min\{\text{LFT of } t_k\}$, the corresponding transition, the one for which the LFT is minimum, must fire, modifying the marking and so the state of the TPN.

Delay θ is not a global time; it can be seen as given by a virtual clock, local to the transition, that must have the same timing unit (e.g., in terms of seconds) than the others in the TPN. Then, as θ is relative to the time τ at which state S has been reached, the absolute firing time, at which t_i may be fired, is defined when needed as "θ + the absolute time τ at which state S has been reached."

D. Firing Rule Between States

Let us assume transition t_i be firable at time $\tau + \theta$ from state $S = (M, I)$. Then the state $S' = (M', I')$ reached from S by firing t_i at the relative time θ can be computed as follows.

1) M' is computed, for all places p, as:

$$(\forall\, p)M'(p) = M(p) - B(t_i, p) + F(t_i, p), \text{ as usually in Petri nets,}$$

2) I' is computed in three steps:
   a) Remove from the expression of I the intervals that are related to the transitions disabled when t_i is fired: these transitions disabled are those enabled by M and not enabled by $M(\cdot) - B(t_i, \cdot)$; they include transition t_i.
   b) Shift of the value θ towards the origin of times all remaining firing intervals, i.e., the intervals that remain enabled and so remain in I, and truncate them, when necessary, to nonnegative values.
   c) Introduce in the domain the static intervals of the new transitions enabled; the new transitions enabled are those not enabled by $M(\cdot) - B(t_i, \cdot)$ and enabled by $M'(\cdot) = M(\cdot) - B(t_i, \cdot) + F(t_i, \cdot)$.

In other terms, step a) corresponds to projecting the domain on the dimensions corresponding to the transitions that remained enabled after t_i has fired. At step b), time is incremented by the value θ. This appears by shifting the remaining intervals of the value θ, with time always nonnegative. At step c), the domain of the new state is defined as the product of the current domains of the transitions that remained enabled and of the static firing intervals of the newly enabled transitions.

For instance, let us consider the net of Fig. 1 in order to illustrate a firing. As it has been seen, $M0 = p_1(1), p_5(1), p_7(1)$ and $I0 = (1, 6)$; remember that $I0$ has only one entry, one pair, because $t1$ is the only transition enabled by $M0$. Firing t_1, at time θ_1, for any value in a potentially infinite number of values in $(1, 6)$, for instance 4.55, leads to the state $S1 = (M1, I1)$ with:

$$M1 = p_2(1), p_3(1), p_4(1), p_5(1), p_7(1) \quad \text{and}$$
$$I1 = (1, 6)\ (2, 3)\ (1, 4),$$

having three entries because three transitions t_2, t_3, and t_5 are enabled by $M1$.

Firing t_2 is allowed between the relative time value 1, the minimum value of $(1, 6)$, and the relative time value 3, the minimum value of the maximum values, in this case the minimum value of 6 for t_2, 3 for t_3, and 4 for t_5.

Firing t_2 at time θ_2 in the interval $(1, 3)$ leads to the state $S2 = (M2, I2)$ with

$$M2 = p_1(1), p_3(1), p_4(1), p_5(1), p_7(1)$$
$$I2 = (1, 6) \quad (\max(0, 2 - \theta_2), 3 - \theta_2)$$
$$(\max(0, 1 - \theta_2), 4 - \theta_2)$$

for t_1, t_3, and t_5. Three transitions are still enabled, giving again three pairs for $I2$.

Those intervals can range

$$\text{from } (1, 6)\ (1, 2)\ (0, 3) \quad \text{for } \theta_2 = 1 \quad \text{(earliest firing)}$$
$$\text{to } (1, 6)\ (0, 0)\ (0, 1) \quad \text{for } \theta_2 = 3 \quad \text{(latest firing).}$$

Note for instance that in the latest firing case t_3 must be fired immediately, at $\theta = 0$, because of $(0, 0)$ and t_5 may also be fired at $\theta = 0$, because of $(0, 1)$.

E. Characterizing the Behavior of a TPN

The behavior "transition t_i is firable from state S at time θ and its firing leads to state S'" will be denoted as:

$$S \xrightarrow{(t_i, \theta)} S'.$$

A firing schedule will be a sequence of pairs

$$(t_1, \theta_1) \cdot (t_2, \theta_2) \cdot \cdots \cdot (t_n, \theta_n)$$

in which $t_1, t_2, \cdots, t_n$ are transitions and $\theta_1, \theta_2, \cdots, \theta_n$ are times. This firing schedule is feasible from a state S iff there exist states

$S1, S2, \cdots, Sn$ such that:

$$S \xrightarrow{(t_1,\theta_1)} S1 \xrightarrow{(t_2,\theta_2)} S2 - - - - \rightarrow Sn-1 \xrightarrow{(t_n,\theta_n)} Sn.$$

The firing rule permits one to compute states and a reachability relation among them. The set of states that are reachable from the initial state or the set of firing schedules feasible from the initial state characterize the behavior of the TPN, in the same way as the set of reachable markings or the language of firing sequences characterize the behavior of a Petri net.

Unfortunately, using this set of states for analysis purposes is not possible in general since this set may be infinite: only simulation can be conducted by using such a simple approach; an appropriate, general method for an exhaustive analysis will be given in the next section.

F. Comments

1) Multiple Enabledness of Transitions: Let us first assume that transition t_i, with firing interval (α_i, β_i), is enabled by the current marking and that time θ (with $\alpha_i \leq \theta \leq \beta_i$) has elapsed since it was enabled. Let us also assume that a different transition t_i' is fired at time θ; this is possible because another transition can fire independently of t_i. After step 2)-b) of the firing of t_i', the current firing interval of t_i is $(\max(0, \alpha_i - \theta), \beta_i - \theta)$.

Let us now assume that firing this transition t_i' makes transition t_i twice enabled: firing t_i' adds one or more tokens to the input places of transition t_i in such a way that the new marking M' becomes only twice (but not three times) enabled, with

$$(\forall\, p)(2 \cdot B(t_i, p) \leq M'(p))$$

and (there exists p) $(M'(p) < 3 \cdot B(t_i, p))$.

After step 2)-c) of the firing rule for t_i', transition t_i is now twice enabled and is related to two intervals:

- $(\max(0, \alpha_i - \theta), \beta_i - \theta)$ for the first time it was enabled, timeinterval1;
- (α_i^S, β_i^S) for the second time it is enabled, timeinterval2.

One of these intervals must be considered for t_i to be fired when time elapses: anyone randomly? the oldest? etc.

Several interpretations seem possible for multiple enabledness, and the firing rule will depend on the chosen interpretation.

Many strategies may be devised and consistent meanings may certainly be found for most of them. In what seems to be a general interpretation, transitions enabled several times simultaneously can be considered as independent occurrences of the same transition. As a consequence, the choice has been taken to fire the occurrence that is related to the oldest interval: then transitions are fired with, in some sense, a first-in first-out discipline. The general theory and a general software exist based on the later choice, but for simplicity and unless explicitly mentioned, the nets considered in the remainder of this paper often obey to the previous restriction, i.e., no transition can be enabled more than once: the generalization is not difficult but complicates the notations because the set of enabled transitions must be enlarged to include any transition t_i multiplely enabled j times, for instance using a notation such as $t_{(i,\text{enabled}(j))}$.

2) Firing: Firing an enabled transition t_i at time θ leads to a new state $S' = (M', I')$ where the new firing intervals I' for transitions are:

a) For all transitions not enabled by the new marking M', then empty;

b) For all transitions t_k enabled by marking M and not in conflict with transition t_i, i.e., transitions which remain enabled, then:

$$I' = (\max(0, \text{EFT}k - \theta), \text{LFT}k - \theta)$$

where EFTk and LFTk denote the lower and upper bound of interval in I corresponding to t_k, respectively;

c) All other transitions (those newly enabled by M' and those enabled by M' but already enabled by M and in conflict with t_i in M) move their firing interval set equal to their static firing interval.

Only the enabled transitions not in conflict with t_i have their intervals shifted by the value of θ toward the time origin; the remaining transitions have their interval set to their static firing interval.

The firing rule above defines a reachability relation among states of a time Petri net. Firing sequences can be defined but enumerating this set of states is not possible for a very high or infinite number of time values can be selected to fire a transition from a given marking. State classes $C = (M, D)$ will be proposed in the sequel to define a finite enumerative analysis method for characterizing the behavior of TPN's.

III. An Enumerative Method for Analyzing Time Petri Nets: State Classes and Reachability

A. Introduction

Informally speaking, state classes will be defined as the union of all firing values which are possible from a given marking; a state class, $C = (M, D)$, will be defined by a pair: a marking M and a domain D. In particular, D will finitely represent the infinite number of firing times possible from marking M. The marking will be the usual marking in Petri nets and the domain D will be defined as the set of solutions of a system of inequalities, these inequalities capturing the global timed behavior of the TPN.

It will be shown that the number of state classes, with static EFT's and LFT's for transitions chosen among rational numbers, is bounded iff the TPN is bounded in the sense of ordinary Petri net theory, i.e., iff the number of tokens in any place, and for any reachable marking, is bounded.

First, it should be clear that the necessary and sufficient condition [7] for boundedness of usual Petri nets and vector addition systems provides a sufficient condition for boundedness of TPN's. This condition allows checking of boundedness of a large class of TPN's but has proven to be too weak for applications in protocol design. Boundedness has been shown undecidable for TPN's but, fortunately, it will be shown that suitable sufficient conditions can be stated. As a consequence, a tree of state classes can be built and, for the tree not to be infinite, the state classes have to be checked for equality: some finiteness conditions will have to be fulfilled and an enumerative method will be given. It follows that using the resulting bounded state class graph of a TPN allows one to check the behavior of systems represented by TPN's.

Let us now informally develop all previously mentioned items, introducing state classes and firing rules, where computing state classes extends computing states as defined in Section II. Formal definitions will appear in Section III-B.

1) State Classes: In the previously defined notion of state, a state is reached from the initial state by a given sequence of values of firing times related to a firing sequence ω. Rather than considering this state, consider the set of all states reached from

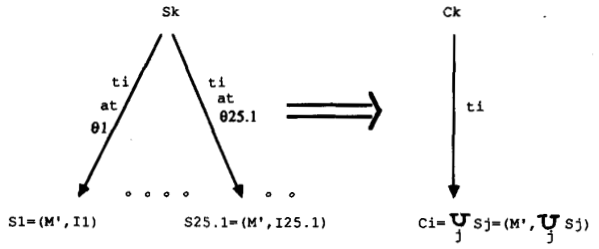


Fig. 2. From states to state classes.

the initial state by firing all feasible firing values corresponding to the same firing sequence ω: it defines a set of states, all reachable by firing ω. This set of states will be considered to be an aggregated pseudostate and will be called the state class associated with the firing sequence ω.

It will be shown in Section III-B that state classes are pairs $C = (M, D)$ in which:

- M is a marking, the marking of the class: all states in the class have the same marking;
- D is the firing domain of the class; it is defined as the union of the firing domains of all the states in the class.

D will be defined as the solution set of a system of inequalities in which variables are 1 to 1 associated with the transitions enabled by marking M.

$$D = \{t \mid A \cdot t \geq b\},$$

where A is a matrix, b a vector of constants, and t a vector of variables, depending on the enabled transitions.

Section III-B will show that the firing domain of the initial state class can be expressed as above and that the definition of the firing rule maintains the form above.

As in the Firing Interval set I in Section II, all transitions are assumed to be ordered in t, and t (i) will refer to the ith transition enabled by M; in the general case, t (i) will refer to a given transition, say t_k, ordered as the ith in t.

The step from states to state classes, given in the diagram of Fig. 2 for a given transition, shows that state classes are defined as containing all possible firing times that may occur from a given reachable marking. Of course, this representation starts from the initial marking which appears in the initial state and in the initial class.

2) Computing Classes: In practice, it is interesting to compute recursively the set of classes, i.e., to derive the class associated with sequence $\omega \cdot t_i$ from the class associated with sequence ω followed by firing transition t_i. The initial class will of course be defined as the class containing the initial static state; it will account for the initial marking and the initial static intervals.

Unlike expressions of intervals for states, expressions of domains for state classes are able to introduce and capture subtle relationships between the firing times of several transitions, for instance when they remain enabled after a firing; this is the key point.

3) Firability of Transitions from Classes: Still assuming that transition $t(i)$ is the ith transition enabled by marking M, let us first illustrate that transition $t(i)$ is firable from class $C = (M, D)$ iff both the following conditions hold:

a) $t(i)$ is enabled by marking M:

$$(\forall\, p)\,(M(p) \geq B(t(i), p))\,.$$

b) The firing interval related to transition $t(i)$ must satisfy the following augmented system of inequalities:

$$A \cdot t \geq b$$
$$t(i) \leq t(j) \qquad \text{for all } j, j \neq i$$

where $t(j)$ also denotes the firing interval related to the jth component of vector t: this enforces that firing $t(i)$ must occur before the min of all LFT's related to all enabled transitions.

Note that $t(i) \leq t(j)$ for all j, $j \neq i$ in b) above refers to condition 2) in Section II-C.

It would be difficult to express condition b) above for classes only using the EFT's and LFT's of the transitions, as we did for states. This is because some nontrivial relationships exist between the firing times of different transitions, as it will be shown now.

Let us consider two illustrative cases related to classes (Fig. 1).

Case 1, simple case. The initial Class, $C0$ is given by

$$M_0 = p_1, p_5, p_7$$
$$D_0 = (\text{all solutions of})\ 1 \leq \theta_1 \leq 6\,.$$

After the firing of t_1, the next Class $C1$ is simply given by

$$M_1 = p_2, p_3, p_4, p_5, p_7$$
$$D_1 = 1 \leq \theta_2 \leq 6$$
$$2 \leq \theta_3 \leq 3$$
$$1 \leq \theta_5 \leq 4\,.$$

Then transition t_2 can fire if, furthermore,

$$\theta_2 \leq \theta_3$$
$$\theta_2 \leq \theta_5.$$

Similar constraints will have to be used when considering the firing of transitions t_3 and t_5.

Let us now consider the general case.

Case 2, general case. When firing t_1, no transition already enabled remained enabled after the firing, giving a simple case. A complex case occurs when some transitions remain enabled after a firing; it will be considered now.

After the firing of t_1, transition t_2 for instance can fire from time $\theta = 1$ to $\theta = \theta$ max related to an upper bound (of another constraining transition or of its one) because at that θ max the corresponding transition has to fire: then t_2 can fire at any θ_2 in the interval $1 \leq \theta_2 \leq 3$, because of t_3.

From marking $M1$, enabling transitions t_2, t_3, and t_5, firing t_2 is possible if the following system has a solution:

$$1 \leq \theta_2 \leq 6 \tag{1}$$
$$2 \leq \theta_3 \leq 3 \tag{2}$$
$$1 \leq \theta_5 \leq 4 \tag{3}$$
$$\theta_2 \leq \theta_3 \tag{4}$$
$$\theta_2 \leq \theta_5. \tag{5}$$

Note that Equations (1)–(3) hold for t_2, t_3, t_5; if firing $t3$ is considered instead of t_2, then (4) and (5) must be replaced by $\theta_3 \leq \theta_2$, $\theta_3 \leq \theta_5$. As a consequence, from marking $p_2p_3p_4p_5p_7$, transition t_3 can be fired at $2 \leq \theta_3 \leq 3$ (and t_5 at $1 \leq \theta_5 \leq 3$).

Computation of all possible firing times for transitions, including the ones that remain enabled, can, in fact, be rather elegantly handled by an adequate change of variables resulting from the

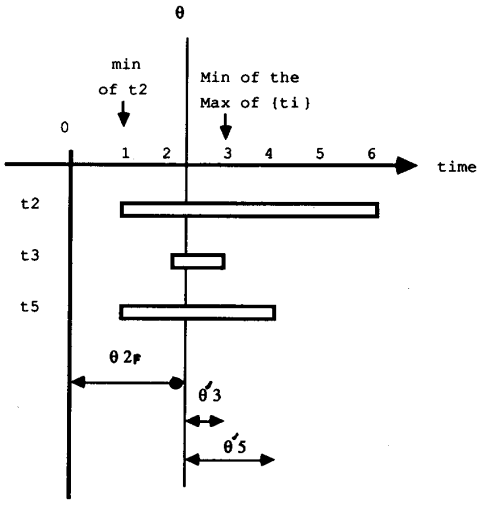


Fig. 3. A timed firing schedule.

translation in the augmented system, as illustrated below, where θ_{2F} denotes the relative time at which t_2 is fired.

Let us suppose that t_2 is fired after a given time θ_{2F} (between 1 and 3), as shown in Fig. 3. It appears that, after the firing of t_2, transitions t_3 and t_5 remain enabled while a given time, θ_{2F}, has elapsed. After the firing, their new time values, θ'_3 and θ'_5, can be defined by a translation where $\theta_i = \theta'_i + \theta_{2F}$.

Firing t_2, using: $2 \le \theta_3 \le 3$, $1 \le \theta_5 \le 4$, and $\theta_i = \theta'_i + \theta_{2F}$, gives

$$2 \le \theta'_3 + \theta_{2F} \le 3 \tag{6}$$
$$1 \le \theta'_5 + \theta_{2F} \le 4 \tag{7}$$

or

$$2 - \theta_{2F} \le \theta'_3 \le 3 - \theta_{2F} \tag{8}$$
$$1 - \theta_{2F} \le \theta'_5 \le 4 - \theta_{2F} \tag{9}$$

with

$$1 \le \theta_{2F} \le 3\,. \tag{10}$$

As θ_{2F} introduces a relationship between t_3 and t_5, [6] and [7] can be written as

$$2 - \theta'_3 \le \theta_{2F} \le 3 - \theta'_3 \tag{11}$$
$$1 - \theta'_5 \le \theta_{2F} \le 4 - \theta'_5. \tag{12}$$

Eliminating θ_{2F} gives the relationships between the firing times of t_3 and t_5 and defines the firing times of the next state Class:

$$0 \le \theta'_3 \le 2 \quad \text{from (8) and (10)}$$
$$0 \le \theta'_5 \le 3 \quad \text{from (9) and (10)}$$
$$\theta'_5 - \theta'_3 \le 2 \quad \text{from (11) and (12)}$$
$$\theta'_3 - \theta'_5 \le 2 \quad \text{from (11) and (12)}$$

where the two latter equations, defining an area of plan, expresses the existing and needed relationships between t_3 and t_5: those relationships which existed must not be relaxed when firing t_2.

Then the state class reached after the firing of t_2, $C2$, is given by

$$M_2 = p_1, p_3, p_4, p_5, p_7$$

and

$$D_2 = 0 \le \theta_3 \le 2$$
$$0 \le \theta_5 \le 3$$
$$\theta_5 - \theta_3 \le 2$$
$$\theta_3 - \theta_5 \le 2$$
$$1 \le \theta_1 \le 6 \quad \text{(for the newly enabled transition } t_1\text{)}.$$

Note that some or all of the constraints such as $\theta_5 - \theta_3 \le 2$ and $\theta_3 - \theta_5 \le 2$ can be redundant; for instance it can be checked in the example that the former ($\theta_5 - \theta_3 \le 2$) is not, while the second ($\theta_3 - \theta_5 \le 2$) is. So, in this example, $\theta_5 - \theta_3 \le 2$ must still hold after the firing of t_2 and must appear in $D2$.

Then, TPN's are also not easy to analyze, since there exists an infinite number of firing instants for any transition, and when a transition fires, its firing must keep relevant constraints.

4) Firing Rule for Time Petri Nets: Let us now consider again the formal definition of classes and assume that transition $t(f)$ is fired from a class $C = (M, D)$ where

$$D = \{t \mid A \cdot t \le b\}.$$

Class $C' = (M', D')$, reached from class $C = (M, D)$ by firing transition $t(f)$ is defined, extending Section II, to be computed as follows.

a) As in Petri nets, the new marking is defined by

$$(\forall\, p)\,(M'(p) = M(p) - B(t(f), p) + F(t(f), p))\,.$$

b) Domain D' is computed from domain D by a three-step procedure.

i) Add to the system $A \cdot t \ge b$, that defines domain D, the firability conditions for transition $t(f)$; this leads to the augmented system:

$$A \cdot t \ge b$$
$$t(f) \le t(j) \qquad \text{for all } j, j \ne f.$$

Make the following change of variables: express all times related to variables $t(j)$, with $j \ne f$, as the sum of the time of fired transition $t(f)$ and of a new variable $t''(j)$ with

$$t(j) = t(f) + t''(j), \qquad \text{for all } j, j \ne f$$

and eliminate from the system the variable $t(f)$ by deriving the new firing intervals and the needed constraint relationships.

The resulting system may be written:

$$A'' \cdot t'' \ge b''$$
$$0 \le t''$$

with A'', b'' computed from A, b, the equations that define the new variables, and using Fourier's method (see [8] for instance) for eliminating variable $t(f)$.

ii) As for $t(f)$, in a similar way, eliminate from the system obtained after step i), accounting for the relationships they imply, all variables corresponding to the transitions disabled when $t(f)$ is fired: these transitions are those enabled by M and not enabled

by $M(\cdot) - B(t(f), \cdot)$ i.e., before computing the new marking.

iii) Augment the system obtained after step b) with new variables, one associated with each transition newly enabled, and define these variables to belong to their static firing intervals. The newly enabled transitions are those not enabled by $M - B(t(f), \cdot)$ and enabled by M'.

It will be shown in the next section that this new system of inequalities may be written as

$$A' \cdot t' \geq b';$$

it has as many variables as there are transitions enabled by marking M' and its solution set defines D'.

Comments on steps in b):

1) Assuming that class C is the class associated with the firing of some sequence σ, then the solution set of the system found in b)-i) is the union of the firing intervals of the states belonging to class C, from which transition $t(f)$ is firable and restricted by $t(f) \leq t$ (others).
2) Considering component $t(f)$ as a parameter, the system expressed with the new variables $t''(j)$ after b)-i) step gives, for a given value θ_{fF} of the parameter, the possible vectors of firing times for all enabled transitions distinct from $t(f)$ with, as a new origin for times, the moment at which transition $t(f)$ is fired.

   Further, eliminating variable $t(f)$ gives a system whose solution set is the set of all possible vectors of firing times for these transitions, relative to the moment at which transition $t(f)$ is fired, this for all possible values of the firing time of $t(f)$.
3) The solution set of system derived from step b)-ii) is the projection, on the remaining dimensions, of the solution set of the system in step i). Relationships between variables that remain in the system are not and must not be changed by this projection.
4) The resulting solution set in iii) is the product of the set found at step ii) and of the static firing intervals of the newly enabled transitions. Notice that from this definition of firing, firing any self-loop transition implies that this self-loop transition will become a newly enabled transition every time it fires.

B. Formal Definition of the Behavior of Time Petri Nets

It will be shown here that, when firings are considered, the domains of state classes can be described as a set of inequalities of the form:

$$\alpha_i \leq t(i) \leq \beta_i, \quad \text{for all } i, \text{ for each } t(i) \text{ newly enabled or which remains enabled,}$$
$$t(j) - t(k) \leq \gamma_{jk}, \quad \text{for all } j,k \text{ with } j \neq k, \text{ for each pair } t(j), t(k) \text{ which remains enabled,}$$

where $t(m)$ represents both the mth transition enabled and the possible firing times associated with this mth transition enabled by the class.

1) Formal Definition of D:

Lemma 1: General form. The firing domains D of state classes for any T-Safe TPN can be expressed as solution sets of systems of inequalities of the following form:

$$\alpha_i \leq t(i) \leq \beta_i \quad \text{for all } i$$
$$t(j) - t(k) \leq \gamma_{jk} \quad \text{for all } j,k \quad k \neq j.$$

Proof: Let us prove that this is true for the initial class and that this form is conserved by transitions between classes.

Part A:

P1: Initial class.

P1-a: The initial marking of the initial class is the initial marking of the Petri net (the Petri net obtained from the TPN when all time pairs have been deleted).

P1-b: The initial domain is defined as a set of inequalities: $\alpha_i^S \leq t(i) \leq \beta_i^S$, for all $t(i)$, where $t(i)$ is associated with the ith transition enabled by the initial marking and α_i^S and β_i^S are the Static EFT and LFT of that transition, respectively (β_i^S may be unbounded).

Default values for γ_{jk} may be provided with $\gamma_{jk} = \beta_j^S - \alpha_k^S$; by definition, these inequalities are redundant and do not affect the solution set of the system $\alpha_i^S \leq t(i) \leq \beta_i^S$.

So, the initial firing domain fulfils the general form.

Part B: Computing the firing domain of a next Class, as defined in the firing rule, consists of three steps: let us show that they produce, from a system with the general form, new systems that also have this general form.

P2: Step 1: It consists of step (a) of the firing rule. Starting from a system with the form given above, let us show that Step 1 transforms as follows this system, $t(f)$ being the variable associated with the transition fired:

(OP1)
- αi becomes $\max(0, -\gamma_{fi}, \alpha_i - \beta_f)$ referred to as $\max(\alpha - 1 - 1, \alpha - 1 - 2)$
- β_i becomes $\min(\gamma_{if}, \beta_i - \alpha_f)$ referred to as $\min(\beta - 1 - 1, \beta - 1 - 2)$
- γ_{jk} becomes $\min(\gamma_{jk}, \beta_j - \alpha_k)$ referred to as $\min(\gamma - 1 - 1, \gamma - 1 - 2)$.

All inequalities containing variable $t(f)$ disappear from the system.

P2-a: By Definition, $\alpha \geq 0$, giving $\max(0)$.

P2-b: $t(f)$ appears in

$$\alpha_f \leq t(f) \leq \beta_f \tag{13}$$
$$t(i) - t(f) \leq \gamma_{if} \tag{14}$$
$$t(f) - t(i) \leq \gamma_{fi}. \tag{15}$$

$t(f)$ does not appear in

$$\alpha_i \leq t(i) \leq \beta_i \tag{16}$$
$$t(j) - t(k) \leq \gamma_{jk}. \tag{17}$$

From (16), using $t(i) = t(f) + t''(i)$, it comes: $\alpha_i - t(f) \leq t''(i) \leq \beta_i - t(f)$, and from (13): $\alpha_i - \beta_f \leq t''(i) \leq \beta_i - \alpha_f$, which gives $\alpha - 1 - 2$ and $\beta - 1 - 2$. From (14), it comes: $t''(i) + t(f) - t(f) \leq \gamma_{if}$, which gives $\beta - 1 - 1$. From (15): $t(f) - t''(i) - t(f) \leq \gamma_{fi}$, giving $t''(i) \geq -\gamma_{fi}$ and $\alpha - 1 - 1$. From (17): $t''(j) + t(f) - t''(k) - t(f) \leq \gamma_{jk}$, which gives $\gamma - 1 - 1$.

P2-c: the new relationships between $t(j)$ and $t(k)$ come

through $t(f)$, as

$$\alpha_j \le t''(j) + t(f) \le \beta_j$$
$$\alpha_j - t''(j) \le t(f) \le \beta_j - t''(j)$$
$$\text{and } \alpha_k - t''(k) \le t(f) \le \beta_k - t''(k),$$
$$\text{giving } t''(k) - t''(j) \le \beta_k - \alpha_j \quad (\gamma - 1 - 2)$$
$$\text{and } t''(j) - t''(k) \le \beta_j - \alpha_k \quad (\gamma - 1 - 2).$$

Step 2: Eliminations correspond to step (b) of the firing rule and consist of successive eliminations in the system of the variables associated with the transitions disabled when $t(f)$ is fired. Each elimination, for instance of variable $t(e)$, corresponds to the following transformation of the system:

(OP2)
- αi becomes $\max(\alpha_i, \alpha_e - \gamma_{ei})$ or $\max(0, \alpha - 2 - 1, \alpha - 2 - 2)$
- β_i becomes $\min(\beta_i, \beta_e + \gamma_{ie})$ or $\min(\beta - 2 - 1, \beta - 2 - 2)$
- γ_{jk} becomes $\min(\gamma_{jk}, \gamma_{je} + \gamma_{ek})$ or $\min(\gamma - 2 - 1, \gamma - 2 - 2)$.

As before, $t(e)$ appears in

$$\alpha_e \le t(e) \le \beta_e \tag{18}$$
$$t(e) - t(k) \le \gamma_{ek} \tag{19}$$
$$t(j) - t(e) \le \gamma_{je} \tag{20}$$

and not in

$$\alpha_i \le t(i) \le \beta_i \tag{21}$$
$$t(j) - t(k) \le \gamma_{jk}. \tag{22}$$

Equations (21) and (22) give $\alpha - 2 - 1$, $\beta - 2 - 1$, and $\gamma - 2 - 1$; (19) and (20) give $\gamma - 2 - 2$; (20) and (18) give $\alpha - 2 - 2$ and $\beta - 2 - 2$.

All inequalities containing variable $t(e)$ disappear from the system.

Step 3: Corresponds to step (c) of the firing rule; it consists of successively introducing in the system the variables and inequalities relative to the newly enabled transitions. Each introduction of variable $t(n)$ for instance, corresponds to the following transformation of the system:

- For all i, j, k, distinct from n, α_i, β_i, and γ_{jk} do not change.
- A new variable $t(n)$, when introduced, is constrained by the inequalities:

$$\alpha_n^S \le t(n) \le \beta_n^S$$

in which α_n^S and β_n^S denote the Static EFT and LFT of the transition associated with $t(n)$, respectively.

(OP3)
- Further inequalities have to be provided for relationships between variable $t(n)$ and the others. These inequalities are

$$t(n) - t(k) \le \gamma_{nk} \qquad \text{for all } k, k \ne n$$
$$t(j) - t(n) \le \gamma_{jn} \qquad \text{for all } j, j \ne n.$$

Default values for γ_{nk} and γ_{jn} must be chosen such that these inequalities are redundant. This is achieved by selecting

$$\gamma_{nk} = \beta_n^S - \alpha_k$$
$$\gamma_{jn} = \beta_j - \alpha_n^S.$$

For achieving the proof of Lemma 1, it must be shown that firing a transition keeps the general form for the systems that define the firing domains. Firing a transition consists of an application of transformation (OP1) followed by a number of applications of (OP2) and a number of applications of (OP3). Each of these transformations individually keeps the general form, thus so does their combination. Q.E.D.

2) How Classes Express the Behavior of the TPN: Using the firing rule, a tree of classes can be built. Its root is the initial class and there is an arc labelled with t_i from class C to class C' if t_i is firable from class C and if its firing leads to class C'.

In this tree of classes, and this is of importance, each class has only a finite number of successors, at most one for each transition enabled by the marking of the class.

From the definition of the classes, any sequence of transitions firable in the TPN will be a path in the above tree of classes. Further, the existence of a path labelled ω between two classes C and C' of the tree must be interpreted as follows.

There exist two feasible firing schedules (σ, t_1) and (ω, t_2) such that

$$S_0 \xrightarrow{(\sigma,t_1)} S \xrightarrow{(\omega,t_2)} S'$$

where S belongs to class C and S' to class C'.

3) Checking State Classes for Equality: When the tree of classes of a TPN will have a bounded number of distinct nodes, a finite graph will be associated to the net. The graph is obtained by grouping equal classes of the tree into the same class. This graph will of course be called the reachability graph of the TPN.

Two state Classes C1 and C2 are defined to be equal iff both

$$\text{their markings are equal, } M1 = M2$$
$$\text{and their firing domains are equal, } D1 = D2.$$

Checking firing domains for equality requires some comments since domains are defined as solution sets of systems of linear inequalities.

In the general case, comparing for equality the solution sets of two systems of linear inequalities, with the same variables and having n and m inequalities respectively may be done by solving m systems with $(n+1)$ inequalities and n systems with $(m+1)$ inequalities with same variables as the systems to compare. This method is a straightforward application of the set equality, the solution set of a system of inequalities being the intersection of the solution sets of its constituting inequalities.

Although possible, this method is not efficient since every computed domain has to be compared pairwise with each previously computed domain. A better method is first to put the systems that define the domains into some canonical form, as soon as they are computed, and then comparing for identity the canonical forms of the systems. This canonical form for systems should have obviously the property that canonical forms of two systems are identical if and only if the solution sets of the systems are equal.

This problem will not be addressed in the general case because, as it has been seen, firing domains for TPN have been defined by systems of inequalities with at most two variables per inequality:

(GF)
$$\alpha_i \le t(i) \le \beta_i \qquad \text{for all } i;$$
$$t(j) - t(k) \le \gamma_{jk} \qquad \text{for all } j, k \text{ with } j \ne k$$

where $t(i)$, $t(j)$, and $t(k)$ are variables and α_i, β_i, and γ_{jk} are constants.

Let us start from a system with the above form. In order to simplify the algorithms, it has been shown in [8] that a canonical form for this system can be defined as:

$$\text{(CF)} \quad \begin{aligned} &\alpha_i^* \le t(i) \le \beta_i^* && \text{for all } i; \\ &t(j) - t(k) \le \gamma_{jk}^* && \text{for all } (j,k) \text{ with } j \ne k \end{aligned}$$

where

α_i^* is the smallest possible value of variable $t(i)$
β_i^* the largest possible value of variable $t(i)$
γ_{jk}^* the largest possible value of the difference $t(j) - t(k)$.

The solution set of a given system (GF) can be associated with only one system of form (CF) since transforming into an equality any inequality in system (CF) gives an equality the solution set of which contains a part of the domain boundary.

Details of the method are not given here, but computing this canonical form from a system (GF) may be done by a series of eliminations and comparisons. About the complexity of computing the canonical form, it is conjectured that this can be done in polynomial time. Strong conviction for this comes from a result in [24] in which it is shown that solving systems of inequalities with at most two variables per inequalities can be done in polynomial time.

In order to have more efficient computations, state class may thus be characterized by its marking M, constants α_i^* and β_i^* for each enabled transition, and constant γ_{jk}^* for each distinct pair of enabled transitions. Computation of the canonical form may be put as an additional step to the firing rule; comparing classes for equality becomes comparing per equality strings of numbers.

IV. Some Properties of Time Petri Nets

Let us first introduce some terminology and notations.

- The set of markings a TPN can reach from its initial marking $M0$ will be denoted as $R(Mo)$.
- The reachability problem is whether or not a given marking belongs to $R(Mo)$.
- The boundedness problem is whether or not all markings in $R(Mo)$ are bounded, i.e.,

$$(\exists\, k \in N)\,(\forall\, M \in R(Mo))\,(\forall\, p \in P)\,(M(p) \le k).$$

- A TPN will be said T-bounded if there exists a natural number k such that none of its transitions may be enabled more than k times simultaneously by any reachable marking, i.e.,

$$(\exists\, k \in N)\,(\forall\, M \in R(Mo))\,(\forall\, t_i \in T)\,(\exists\, p \in P)\,(M(p) < (k+1) \cdot B(t_i,p)).$$

  It may be noted that boundedness implies T-boundedness and that the converse is not true.
- The property T-safe is the particular case of the above T-bounded property, with $k = 1$.

Theorem 1 recalls an undecidability result for TPN's.

A. Undecidability

Theorem 1: The reachability and boundedness problems for TPN's are undecidable.

Proof: A direct proof is produced in reference [21]. Others (indirect) proofs may be produced: it can be shown that TPN's can simulate inhibitor nets and priority Petri nets, and have equivalent reachability and boundedness problems. Since these problems are known undecidable for the two latter classes of nets, (see [22] for instance), it may be inferred that they are also undecidable for TPN's.

A straightforward consequence of Theorem 1 is that the finiteness of the set of classes for TPN's is undecidable since classes contain markings.

Two important and sufficient conditions for boundedness will be given later in this section, but some theorems are needed first.

B. Boundedness

Lemma 2: The constants α_i, β_i, and γ_{jk} of any domain computed from the initial system by the firing rule are linear combinations with integer coefficients of the static EFT's and LFT's associated with the transitions of the TPN, i.e., the following hold:

$$\begin{aligned}&(\forall_i)\,(\exists\, \lambda_1, \cdots, \lambda_{2n} \in Z)\\ &(\alpha_i = \lambda_1 \alpha_1^S + \cdots + \lambda_n \alpha_n^S + \lambda_{n+1}\beta_1^S \\ &\qquad + \cdots + \lambda_{2n}\beta_n^S)\end{aligned}$$

(and similarly for each β_i and γ_{jk}).

Proof: The proof is straightforward: Lemma 2 is true for the initial Class and this property holds when the three basic transformations OP1, OP2, and OP3 are individually applied. Q.E.D.

Lemma 3: The constants α_i, β_i, and γ_{jk}, for all i, j, k, of any domain computed from the initial class by the firing rule have the following bounds:

$$\begin{aligned} 0 \le \alpha_i \le \alpha_i^S &\quad \text{referred to as } (1) \le \alpha_i \le (2) \\ 0 \le \beta_i \le \beta_i^S &\quad \text{referred to as } (3) \le \beta_i \le (4) \\ -\alpha_k^S \le \gamma_{jk} \le \beta_j^S &\quad \text{referred to as } (5) \le \gamma_{jk} \le (6). \end{aligned}$$

Proof: They are satisfied by the initial state and trivially by OP3 and they remain satisfied when any of the transformations OP1 and OP2 apply.

(1) holds by Definition, so (3) holds.

(2) and (4) hold because: for OP1 α_i is computed by

$$\begin{aligned} &\alpha_i - \beta_f, && \text{with } \beta_f \ge 0 \text{ or by} \\ &-\gamma_{fi} = -\beta_f + \alpha_i = \alpha_i - \beta_f \\ &\beta_i = \min(\beta_i - \alpha_f, —), && \text{with } \alpha_k \ge 0; \\ &\text{for OP2 } \alpha_i = \alpha_e - \gamma_{ei} = \alpha_e - \beta_e + \alpha_i \\ &\qquad = \alpha_i - (\beta_e - \alpha_e) \le \alpha_i^S \\ &\beta_i = \min(\beta_i, —); \end{aligned}$$

(6) holds because

$$\begin{aligned} &\gamma_{jk} = \min(\beta_j - \alpha_k) \quad \text{and } \alpha_k \ge 0 \quad \text{or} \\ &\gamma_{jk} \text{ is min of } (\gamma_{jk}, \gamma_{je} + \gamma_{ek}); \end{aligned}$$

(5) holds because

$$\begin{aligned} &\gamma_{jk} = \beta_j - \alpha_k \quad \text{and } \beta_j \ge 0, \quad \text{so } \gamma_{jk} \ge -\alpha_k \\ &\gamma_{jk} = \gamma_{je} + \gamma_{ek} = (\beta_j - \alpha_e) + (\beta_e - \alpha_k) \\ &\qquad = (\beta_e - \alpha_e) + (\beta_j - \alpha_k), \quad \text{and as } (\beta_e - \alpha_e) \ge 0, \\ &\gamma_{jk} \ge (\beta_j - \alpha_k) \ge -\alpha_k. \end{aligned}$$

Furthermore, if the static LFT β_i^S is not bounded for some transition t_i, then β_i and γ_{ij} (for all j, $j \neq i$) are initially and will remain unbounded. Q.E.D.

Lemma 4: Let A and B be two bounded constants and $q_1, \cdots, q_n$ be a finite set of rational constants. There is only a bounded number of linear combinations of numbers $q_1, \cdots, q_n$, with integer coefficients, between numbers A and B, i.e., the number of rational numbers X such that

$$X = \lambda_1 q_1 + \cdots + \lambda_n q_n$$
$$\text{and } \lambda_1, \cdots, \lambda_n \in Z$$
$$\text{and } q_1, \cdots, q_n \in Q$$
$$\text{and } A \leq X \leq B \text{ is bounded.}$$

Proof: Let d be the common denominator of rational numbers $q_1, \cdots, q_n$ and Qi the product $d.qi$. The problem above is equivalent to proving that there are only a bounded number of combinations of integers $Q_1, \cdots, Q_n$, with integer coefficients, between the bounds $d.A$ and $d.B$, which is true. Q.E.D.

Lemma 5: If a TPN is T-bounded, then the set of all the firing domains D and of its state classes is finite.

Proof: The proof is carried out for the T-safe case and then extended, with an adequate interpretation, to the T-bounded case.

The possible α_i, β_i, and γ_{jk} for systems that define the state classes are linear combinations with integer coefficients of the static EFT's and LFT's (from Lemmas 1 and 2), they are either unbounded (and in this case remain unbounded) or have upper and lower bounds (Lemma 3).

Further, let us take one given marking Mi. As by definition, the Static EFT's and LFT's are rational numbers, using Lemma 4, only a bounded number of domains related to Mi, distinct combinations of α_i, β_i, and γ_{jk}, can be computed. As the number of markings is bounded, so is the number of classes, pairs (marking, domain).

This proves Lemma 5 for the T-Safe case.

For T-bounded nets, remember that the possibly many variables associated with a given transition are independent. Firing a transition is then firing one of its occurrences (all possible alternatives must be taken) and disabling applies to one or several occurrences of the same transition (all combinations for remaining enabled occurrences must be considered).

The firing rule needs not to be modified: the only difference with the T-safe case is that there are more variables in the system that there are enabled transitions: an n-enabled transition will appear n times in t. But, as the net is T-bounded, the total number of variables is bounded and, using a similar proof that for the T-safe case, only a bounded number of firing domains may be computed for the TPN. Q.E.D.

Remark: Restricting the static EFT's and LFT's of transitions to be rational numbers (instead of real numbers in Merlin's definition) is essential. Lemma 5 does not hold if they are real numbers. For example, the net shown in Fig. 4(a) is bounded when markings are considered (it has only one marking where both places have one token) but has an unbounded number of domains and so of state classes [Fig. 4(b)] because π is not a rational number.

Theorem 2 below addresses the finiteness of the state Classes.

Theorem 2: A TPN has a bounded number of state Classes if and only if it is bounded.

Proof: If it is bounded, then it is T-bounded and it has a bounded number of markings (boundedness property) and a bounded number of state Classes (Lemma 5). If it is not bounded, then it has an unbounded number of markings and classes since classes are pairs (marking, firing domain). Q.E.D.

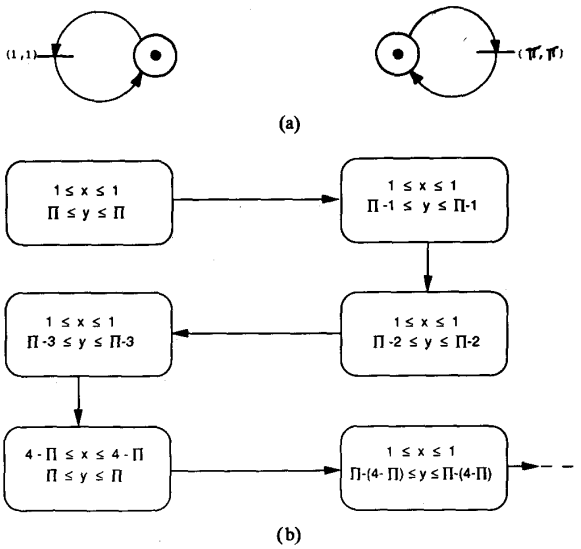


Fig. 4. An unbounded time Petri net having a bounded underlying Petri net.

So, any sufficient condition for the boundedness property provides a sufficient condition for the finiteness of the set of classes. Two sufficient conditions of interest are presented in the sequel.

In particular, it is important for the coming theorems to emphasize that, as the number of domains is bounded, then an unbounded TPN has an unbounded number of markings.

The notation $M' \geq M$ will be used in the following theorems; it is defined as

$$(\forall\, p \in P)\,(M'(p) \geq M(p)) \quad \text{and}$$
$$(\exists\, p \in P)\,(M'(p) > M(p)).$$

Lemma 6: If a T-bounded TPN is not bounded, then a firing sequence of unbounded length, going through a sequence ω of states classes in which all classes are pairwise different, is firable from the initial class.

Proof: From Theorem 2 it comes that an unbounded TPN has an unbounded number of reachable classes. Since each state class only has a bounded number of successors, k for each k-enabled transition, then, by definition of the firing rule, its set of Classes must necessarily contain such a sequence ω. Q.E.D.

Theorem 3: Sufficient condition 1 (SC1).

A T-bounded TPN is bounded if no pair of state classes $C = (M, D)$ and $C' = (M', D')$ are reachable from its initial state class and are such that

$$\text{(SC1)} \quad \text{i) } C' \text{ is reachable from } C.$$
$$\text{ii) } M'(p) \geq M(p).$$

Proof: Let us assume the TPN be unbounded and consider the unbounded sequence ω used in Lemma 6. Since the net is T-bounded, it admits only a bounded number of firing variables and so of firing domains and, as classes are pairs (marking, domain), the unbounded sequence ω must contain an unbounded subsequence ω' in which all markings of the classes are pairwise different. Further, as in [23], such an unbounded subsequence

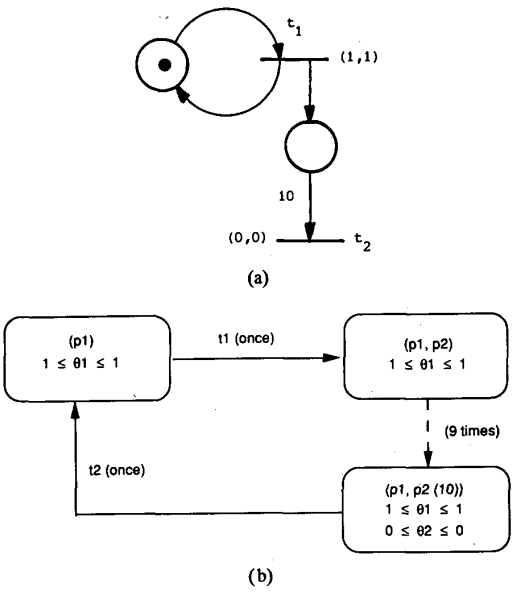


Fig. 5. Bounded TPN recognized as bounded.

ω' necessarily contains two classes $S'_i = (M, D)$ and $S'_i + n = (M', D')$ with $M' \geq M$. Q.E.D.

So, SC1 is sufficient for boundedness. However it is not necessary. Sufficient condition 1 fails for the TPN represented in Fig. 5(a); it is fulfilled though this net only admits a finite number of state classes [Fig. 5(b)]. The reason is that for TPN's, if a firing sequence is firable from a marking M, it does not implies that the same sequence is firable from a marking M' with $M' \geq M$, for this is depending on firing constraints.

It may be noticed that SC1 holds for any TPN such that the underlying Petri net, the net which is obtained from the TPN by removing the time labels from all transitions, is bounded. This does not implies SC1 to be worthless since it also holds for some TPN's that do not satisfy this property.

Nevertheless, SC1 is too weak in the general case.

Theorem 4: Sufficient condition 2 (SC2).

A TPN is bounded if no pair of state classes $C = (M, D)$ and $C' = (M', D')$, reachable from the initial state class, fulfils the following conditions:

i) C' is reachable from C.
ii) $M'(p) \geq M(p)$.
iii) $D' = D$.
iv) $\forall p \in \{p \in P \mid M'(P) > M(P)\}$, then $(M(p) > \max B(t_i, p))$.

Proof: Let us assume that the TPN is unbounded and consider the sequence ω' used in the proof of Theorem 3. As in [23], since the number of distinct subsets of P is bounded, this sequence ω' must necessarily contain an unbounded subsequence $\omega'' = (M_i, D_i)$ with $i \in N$ such that there must exist in this sequence some pair of classes

$$C''_i = (M, D) \quad \text{and} \quad C''_{i+n} = (M', D)$$

with the same domain D and

$$M'(P) > M(P)$$

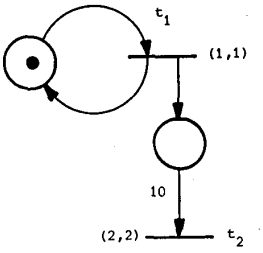


Fig. 6. Bounded TPN not recognized as bounded.

where

$$\forall p \in \{p \in P \mid M'(P) > M(P)\}, \quad (M(p) > k(p))$$

where k is any mapping associating with any place p an integer $k(p)$.

Because firing can remove at the maximum $\max(B(t_i, p))$ tokens, k may be chosen to be greater than all the related transition weights for any place p, $k(p) = \max(B(t_i, p))$. So, SC2 is a sufficient condition for boundedness. Q.E.D.

However, the condition is not necessary: SC2 fails for the net given in Fig. 6 though this net admits only 13 Classes. In the net Fig. 6, firing ten times transition t_1 leads to a state class with marking $p_1(1)$, $p_2(10)$ and firing intervals $(1, 1)$, $(2, 2)$ for transitions t_1 and t_2, respectively. From this class, transition t_1 may be fired two times, leading to two classes C and C', successively. Classes C and C' satisfy the three conditions expressed in Theorem 4 and thus SC2 fails to prove this net bounded. Indeed, the net is bounded: from class C', firing once more transition $t1$ is not allowed, and firing $t2$ leads to a previously enumerated class.

Probably, using time differences or time distances between some pertinent transitions would help to define a still better sufficient condition.

Using Theorem 4, the enumeration of classes stops when a firing sequence is found such that: it does not decrease the marking of any place, and all places it increases the marking, when starting the sequence, have markings greater than some bounds $k(p)$. Though bounds $k(p)$ may be arbitrarily chosen, it seems to be wise to choose them as—or related to—the maximum weight among those of the outgoing arcs from the places. The idea behind this choice is that if the marking of place P reaches that value, then it could enable some transition t_i that, when firing, may decrease the marking of place P of the maximum value: $\max(B(t_i, p))$. This is obviously not always the case, due to the timing constraints or to the fact that some transitions may be dead.

It may be noticed that SC2 permits one to prove boundedness of the net represented in Fig. 5. Also, it is clear from Theorems 3 and 4 that

$$\text{SC2} \Rightarrow \text{SC1}.$$

Finally, when TPN's are bounded, it becomes possible, using the graph of classes, to prove properties that characterize their correct behaviors. Further, liveness properties, similar to those defined for Petri nets, can be defined for TPN's and for bounded TPN's, and can be proved using the graph of state classes. The state class graph can also be used as a usual state transition system for verification, for instance by checking temporal logic assertions on it or by using reduction and observational equivalence.

An APL experimental computer software package, TINA, has been developed for analyzing TPN's. It allows a user controlled enumerative check for boundedness: sufficient condition 2 is used together with conditions expressed by the designer, based upon his intuitive understanding of the behavior of the net. Typical user defined conditions are upper bounds for the markings of some places or relationships between markings of several places. Some industrial tools are under development.

This technique has been proved to be adequate for all applications treated so far. The state class graph is the only way that presently allows one to verify the correctness of explicit time values in dense time dependent systems.

V. Example: The Alternating Bit Protocol

As communication protocols [18] make important use of timing values in their specifications (e.g., recovery from losses of messages is implemented using time-outs), TPN's constitute a suitable approach for verifying that these time-outs are correctly set. The alternating bit protocol [2] will provide an introductory example to the analysis method.

This protocol transmits messages between two entities, allowing only one message in transit at a time, over an unreliable transmission medium. Hypotheses on the behavior of the transmission medium are that messages or acknowledgments may be lost in transit.

Recovery from losses is done using a time-out and retransmitting: each sender records the time at which it sends a message and if an acknowledgment of its delivery does not return within a given time, the message is retransmitted.

This selected mechanism must be sufficient for recovering from losses and for preventing the acceptance of duplicate messages: upon reception of a message, the receiver must be able to decide whether this message is a new message or a duplicate. To solve this problem, messages are numbered, prior to transmission, with modulo-2 sequence numbers and, for every packet received, an acknowledgment is sent that carries the sequence number of the received packet.

Fig. 7 gives a TPN model for such a protocol. Losses of messages and acknowledgments are represented in the net simply as transitions with no output places; they do not need any artificial mechanism for relating lost messages and messages retransmitted.

Of course, estimates for the duration of all elementary actions of the protocol must be provided. In the net represented in Fig. 7, retransmissions of messages occur at a time comprised between 5 and 6 units after the message has been sent. Equal estimates (between 0 and 1) are given for losses and receptions of messages and acknowledgments. Also, no time constraint, i.e., the intervals $(0, \infty)$, are given for sending the numbered messages.

The graph of classes for this net is represented in Fig. 8; 16 classes have been computed. It is clear from the state classes in Fig. 8 that only one message or acknowledgment will be pending at a time and that the transmission medium never holds more than one message or acknowledgment (all places in the net hold at most one token, for any marking). This ensures that the retransmission time-out is correctly set. Furthermore, no duplicate message may be released (transitions $t7$ and $t10$ alternate in all paths of the graph) and the transfer of messages actually occurs (the net is live, the liveness property being defined for TPN's as for Petri nets). The graph given in Fig. 8 can be used as any usual state transition system for verification. Note that the underlying Petri net has a deadlock.

The meanings of the transitions are as follows.

- $t1$ Send Packet 0
- $t2$ Resend Packet 0
- $t3$ Receive Ack 0
- $t4$ Send Packet 1
- $t5$ Resend Packet 1
- $t6$ Receive Ack 1
- $t7$ Receive and Release Packet 0
- $t8$ Send Ack 0
- $t9$ Receive and Reject Packet 0
- $t10$ Receive and Release Packet 1
- $t11$ Send Ack 1
- $t12$ Receive and Reject Packet 1
- $t13$ Lose Packet 0
- $t14$ Lose Ack 0
- $t15$ Lose Packet 1
- $t16$ Lose Ack 1

The Classes which are obtained for this net are as follows.

Class 0

$$M = p1, p5$$
$$D: \quad 0 \le t1$$

Class 1

$$M = p2, p5, p9$$
$$D: \quad 5 \le t2 \le 6$$
$$0 \le t7 \le 1$$
$$0 \le t13 \le 1$$

Class 2

$$M = p2, p6$$
$$D: \quad 4 \le t2 \le 6$$
$$0 \le t8 \le 2$$

Class 3

$$M = p2, p7, p10$$
$$D: \quad 2 \le t2 \le 6$$
$$0 \le t3 \le 1$$
$$0 \le t14 \le 1$$

Class 12

$$M = p2, p7$$
$$D: \quad 1 \le t2 \le 6$$

Class 13

$$M = p2, p7, p9$$
$$D: \quad 5 \le t2 \le 6$$
$$0 \le t9 \le 1$$
$$0 \le t13 \le 1$$

Class 14

$$M = p2, p7$$
$$D: \quad 4 \le t2 \le 6$$

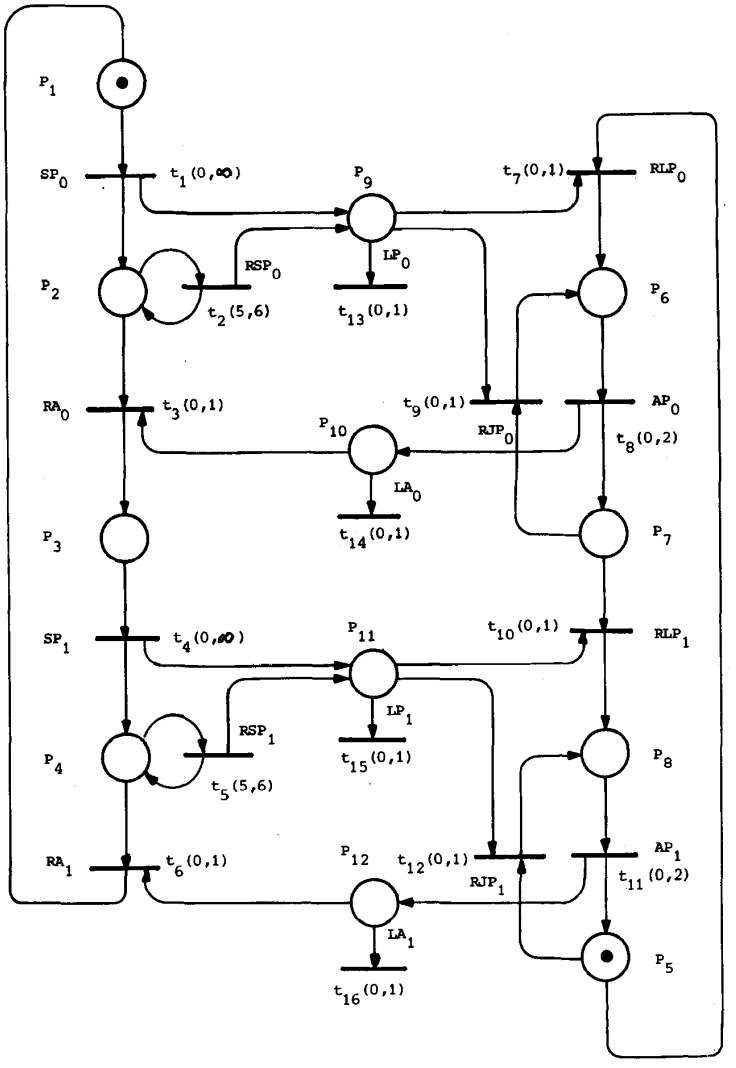


Fig. 7. A simple protocol loosing messages and including time-out values.

Class 15

$$M = p2, p5$$
$$D: \quad 4 \leq t2 \leq 6$$

The other classes are obtained from the above and Fig. 8 by renaming places and transitions for the value 1 of the alternating bit.

Another nontrivial real example, a virtual ring protocol [1], is considered in [17].

VI. Conclusion

This paper has presented an approach for analyzing TPN's which permits one to check the properties of systems in which some values of time must be specified explicitly, as in communication protocols, real-time designs, and high performance architectures.

No necessary and sufficient condition can be stated for the boundedness property. To weaken this limitation, appropriate sufficient conditions have been given and conditions stronger than those stated here must be developed. Also, more specific and semantic checks could be developed in order to be able to stop the enumeration as early as possible if the behavior of the net is not the one expected.

Alternative analysis techniques for TPN's, such as structural analysis techniques, using place or transition invariants [26] developed for Petri nets can be extended to TPN's as in [30].

As enumerative approaches for analyzing Petri nets can produce large sets of classes, even when the net is bounded, Petri nets experts must be able to create nets with manageable numbers of classes when this can be done. This has been possible up to now and the analysis technique for TPN introduced here proved to be quite useful in many examples.

Acknowledgment

The authors would like to thank Prof. M. Menasche, P.U.C., Rio de Janeiro, for his contribution to this work.

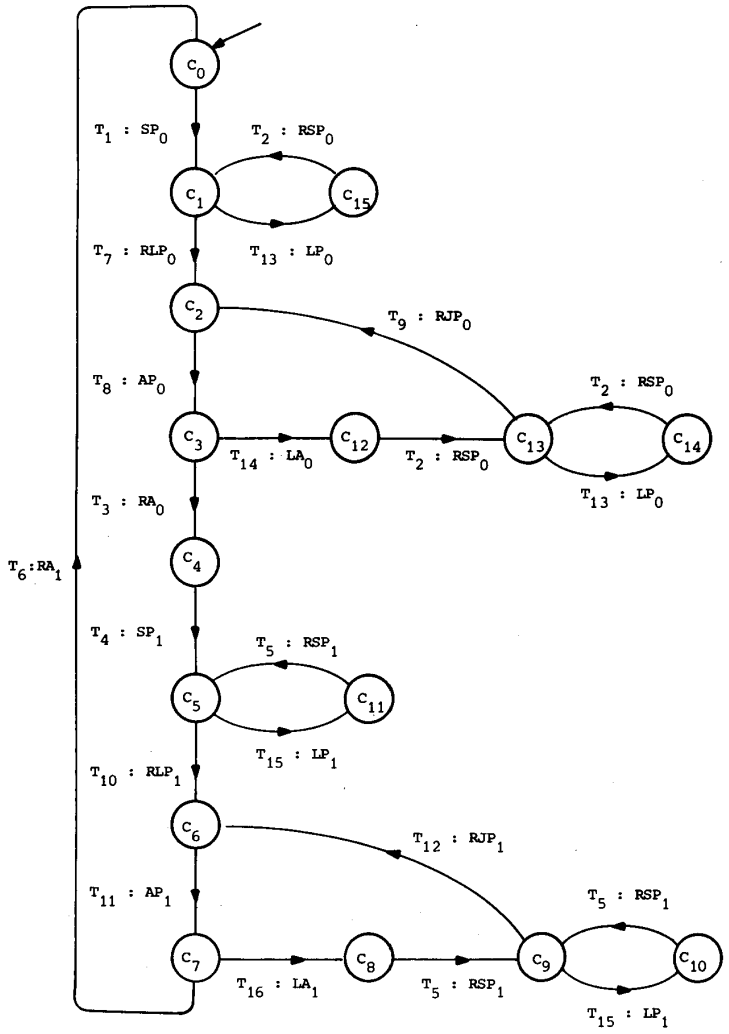


Fig. 8. The class graph for the TPN given in Fig. 7.

References

[1] J. M. Ayache, J. P. Couriat, and M. Diaz, "REBUS, A fault-tolerant distributed system for industrial real-time control," *IEEE Trans. Comput.*, vol. C-31, no. 7, July 1982.

[2] K. A. Bartlett, R. A. Scantlebury, and P. T. Wilkinson, "A note on reliable full-duplex transmission over half-duplex link," *Commun. ACM*, vol. 12, no. 5, May 1969.

[3] B. Berthomieu and M. Menasche, "A state enumeration approach for analyzing time Petri nets," in *Proc. 3rd European Workshop Applications and Theory of Petri Nets*, Varenna, Italy, Sept. 1982.

[4] ——, "An enumerative approach for analyzing time Petri nets," in *Proc. IFIP Cong. 1983*, Paris, Sept. 1983.

[5] P. Caspi and N. Halbwachs, "Analyse approchée du comportement asymptotique de systèmes temporisés," Lab. d'Informatique et de Mathématiques Appliquées de Grenoble, Rep. RR 322, Sept. 1982 (in French).

[6] N. D. Jones, L. H. Landweber, and Y. E. Lien, "Complexity of some problems in Petri nets," *Theoret. Comput. Sci.*, vol. 4, 1977.

[7] R. M. Karp and R. E. Miller, "Parallel program schemata," *J. Comput. Syst. Sci.*, vol. 3, 1969.

[8] M. Menasche, "Analyse des réseaux de Petri temporisés et application aux systèmes distribués," Thesis, Univ. Paul Sabatier, Toulouse, France, Nov. 1982 (in French).

[9] P. Merlin and D. J. Faber, "Recoverability of communication protocols," *IEEE Trans. Commun.*, vol. COM-24, no. 9, Sept. 1976.

[10] P. Merlin, "A study of the recoverability of computer system," Thesis, Dep. Comput. Sci., Univ. California, Irvine, 1974.

[11] C. Ramchandani, "Analysis of asynchronous concurrent systems by timed Petri nets," Massachusetts Inst. Technol., Project MAC, Tech. Rep. 120, Feb. 1974.

[12] A. Danthine, "Protocol representation with finite-state models," *IEEE Trans. Commun.*, vol. COM-28, no. 4, Apr. 1980.

[13] M. Diaz, "Modeling and analysis of communication and cooperation protocols using Petri net based models," *Comput. Networks*, Dec. 1982.

[14] J. Sifakis, *Use of Petri Nets for Performance Evaluation in Measuring, Modelling, and Evaluating Computer Systems.* Amsterdam, The Netherlands: North-Holland, 1977, pp. 75–93.

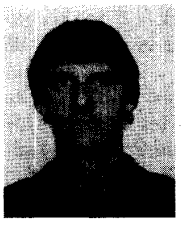
[15] W. M. Zuberek, "Timed Petri nets and preliminary performance evaluation," in *Proc. 7th Annu. Symp. Computer Architecture*, May 6–8, 1980, pp. 88–96.

[16] P. Chretienne, "Some results on the control of timed Petri nets," in *Proc. 2nd Workshop Applications and Theory of Petri Nets*, Bad Honnef, Sept. 1981.

[17] M. Menasche and B. Berthomieu, "Time Petri nets for analyzing and verifying time dependent communication protocols," in *Protocol Specification, Testing and Verification, III*, H. Rudin and C. H. West, Eds. Amsterdam, The Netherlands: North-Holland, 1983.

[18] M. Diaz and P. Azema, "Petri net based models for the specification and validation of protocols," in *Advances in Petri Nets (Lec-*

ture Notes in Computer Science, Vol. 188). New York: Springer-Verlag, 1985.
[19] P. Caspi and N. Halbwachs, "An approach to real time systems modelling," in *Proc. 2nd Workshop Applications and Theory of Petri Nets,* Bad Honnef, Sept. 1981.
[20] G. B. Dantzig, *Linear Programming and Extensions.* Princeton, NJ: Princeton University Press, 1963.
[21] N. D. Jones, L. H. Landweber, and Y. E. Lien, "Complexity of some problems in Petri nets," *Theoret. Comput. Sci.,* vol. 4, pp. 277–299, 1977.
[22] M. Hack, "Petri net languages," Computation Structures Group, Massachusetts Inst. Technol., Project MAC, Memo 124, June 1975.
[23] R. M. Karp and R. E. Miller, "Parallel program schemata," *J. Comput. Syst. Sci.,* vol. 3, pp. 147–195, 1969.
[24] B. Aspvall and Y. Shiloach, "A polynomial time algorithm for solving systems of linear inequalities with two variables for inequality," in *Proc. 20th Annu. Symp. Foundations of Computer Sciences,* Oct. 1979, pp. 205–217.
[25] R. R. Razouk and C. V. Phelps, "Performance analysis using time Petri nets," in *Proc. 4th IFIP Protocol Specification, Testing and Verification,* Y. Yemini *et al.,* Eds. Amsterdam, The Netherlands: North-Holland, 1985.
[26] K. Lautenbach and H. A. Schmidt, "Use of Petri nets for proving correctness of concurrent process systems," in *Proc. IFIP Congr. 1974.* Amsterdam, The Netherlands: North-Holland, 1974, pp. 187–191.
[27] S. Ghosh, "Some comments on timed Petri nets," *Journées sur les Réseaux de Petri,* AFCET, Paris, 1977.
[28] J. Dufau, "Un outil pour la vérification des protocoles décrits par réseaux de Petri," Thèse de Docteur-Ingénieur, Univ. Paul Sabatier, Toulouse, France, Jan. 1984 (in French).
[29] M. A. Holliday and M. K. Vernon, "A generalized timed Petri net model for performance analysis," *IEEE Trans. Software Eng.,* vol. SE-13, no. 12, pp. 1297–1310, Dec. 1987.
[30] J. L. Roux, "Modélisation et analyse des systèmes distribués par les réseaux de Petri temporels," Thèse de Docteur-Ingénieur INSA, Toulouse, France, Dec. 1985 (in French).



Bernard Berthomieu was born in Mazamet, France, on March 17, 1952. He received the Docteur Ing. degree from the University of Toulouse III, Toulouse, France, in 1979.

He has been with the Centre National de la Recherche Scientifique (CNRS) since 1981, and currently works at the Laboratoire d'Automatique et d'Analyse des Systèmes du CNRS, Toulouse. From 1978 to 1984, he was involved in research in the area of Petri net theory. His research activities currently focus on design, semantics, and implementation of parallel programming languages; he leads the LCS project in this area.

Michel Diaz (M'78) received the Doctorat es Sciences degree from the University of Toulouse, Toulouse, France, in 1969.

He is currently Directeur de Recherche at the Centre National de la Recherche Scientifique (CNRS) and leads the Research Group "Software and Tools for Communicating Systems" at the Laboratoire d'Automatique et d'Analyse des Systèmes du CNRS, Toulouse. Since 1969, he has worked on the development of formal methodologies, techniques, and tools for designing distributed systems based, in particular, on extended Petri nets, Estelle, LOTOS, and temporal logic. In 1989 and 1990, he spent nine months as a visiting staff member at the University of Delaware at Newark and at the University of California at Berkeley. He has written more than 120 technical publications and is the editor or co-editor of three North-Holland volumes on developing and using formal description techniques for the specification, testing, verification, and implementation of protocols in distributed systems.

Dr. Diaz is a member of the IFIP and AFCET and is a Technical Editor for *IEEE Communications Magazine.* From 1984 to 1988, he was the prime manager of the SEDOS project (Software Environments for the Design of Open distributed Systems, in which the formal techniques Estelle and LOTOS have been developed) within the ESPRIT program of the CEC. He was a member of the Program Committee of the European Workshop on Applications and Theory of Petri Nets, the IEEE Symposium on Fault Tolerant Computing, the IFIP International Symposium on Local Communication Systems LAN PABX, the IFIP Symposium on Protocol Specification, Testing and Verification, and the ICDCS International Conference on Distributed Computing Systems. He also served as Chairman of the Program Committee for the IFIP Congress on Protocol Specification, Testing and Verification, the European Workshop on Application and Theory of Petri Nets, and the International Conference on Distributed Computing Systems in the area of software engineering.