AN ENUMERATIVE APPROACH FOR ANALYZING TIME PETRI NETS

Bernard BERTHOMIEU and Miguel MENASCHE*
Centre National de la Recherche Scientifique
Laboratoire d'Automatique et d'Analyse des Systemes
7, Avenue du Colonel Roche, 31400 TOULOUSE, FRANCE

abstract: This paper is concerned with specifying and proving correct systems in which time appears as a parameter. We model such systems via Merlin's Time Petri Nets. An enumerative analysis technique is introduced for these nets based on the computation of a set of state classes and a reachability relation on the set. State classes are defined in the text and an algorithm is provided for their enumeration. This enumerative approach allows us to derive a finite representation of their behavior for a large family of Time Petri Nets. The analysis method is illustrated by the analysis of a communication protocol.

1 INTRODUCTION

This paper analyses concurrent systems in which time appears as a quantifiable and continuous parameter. Communication protocols are among such systems: recovery mechanisms for losses of messages or network topology changes are usually implemented using time outs.

Several attemps have been made in the past for specifying and verifying such systems. Among the models developed, two were based on Petri Nets. They are known as Time Petri Nets [12] and Timed Petri Nets [13].

Merlin defined Time Petri Nets as Petri Nets in which two times, a and b, with $0 \leq a \leq b$ and b possibly unbounded, are associated with each transition. Times a and b, for transition t, are relative to the moment at which the transition was last enabled. Assuming that transition t has been last enabled at time θ. Then t may not fire before time $\theta + a$ and must fire before or at time $\theta + b$ unless it is disabled before then by the firing of another transition. Firing a transition takes no time to complete.

Using Time Petri Nets, Merlin investigated the recoverability problems in computer systems and the specification of communication protocols [11], [12].

Ramchandani's Timed Petri Nets are obtained from Petri Nets by associating a firing time with each transition of the net. The firing rule is further modified considering the time it takes to fire the transition, and that transitions fire as soon as enabled. Timed Petri Nets and related equivalent models have been used mainly for performance evaluation.

Merlin's Time Petri Nets were choosen for modeling our systems since these nets have been proven convenient for expressing most of the temporal constraints required, including duration. We developed an enumerative analysis technique for these nets that allows a reachability analysis similar to the well known method for Petri Nets and Vector Addition Systems [7]. This method permits computing a finite representation for the behavior of a large family of TPNs in terms of a set of state classes and a reachability relation on the set.

The method is explained in section 2. Time Petri Nets and their behavior are presented in section 1, section 3 focuses on analysis of some properties of Time Petri Nets, and some applications are discussed in section 4.

*On leave, current address is: Departamento de Engenharia Eletrica, Pontificia Universidade Catolica do Rio de Janeiro, Brazil.
This research has been partially supported by Centre National d'Etudes des Telecommunications, under contract 82 1B 169, 1982

2 TIME PETRI NETS, TERMINOLOGY AND BEHAVIOR

Time Petri Nets (or TPNs for short) are obtained from Petri Nets by associating two times with each transition. Let us call the smallest and the largest of these times for any transition its Static Earliest Firing Time (Static EFT for short) and Static Latest Firing Time (Static LFT for short), respectively. The Static Firing Interval of the transition will be the closed left bounded interval of times comprised between its Static EFT and LFT.

States in TPNs will be pairs S=(M,I) in which M is a marking and I is a a Firing Interval function. Function I associates with each enabled transition the time interval in which the transition is allowed to fire.

When the net is behaving, these intervals are generally different from the Static Firing Intervals, they will be simply refered to as (dynamic) Firing Intervals, and their bounds as (dynamic) EFTs and LFTs.

Firing a transition t, at a time θ , from a state $S = (M, I)$, is allowed iff both the following conditions hold:

(i) The transition is enabled;

(ii) Time θ is comprised (bounds included) between the EFT of transition t and the smallest of the LFTs among those of the transitions enabled.

The first condition is the usual one of enabledness for Petri Nets, the second results from the necessity of firing transitions in their firing interval.

Firing t at the time θ from a state $S = (M, I)$ leads to a new state $S' = (M', I')$, computed as follows:

1) The new marking M' for each place is defined for any place p, as in Petri Nets, as:

$$M'(p) = M(p) - B(t, p) + F(t, p)$$

where B and F are the Backward and Forward Incidence Functions of the net, respectively;

2) The new firing intervals I' for transitions are computed as follows:

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

b) For all transitions k enabled by marking M and not in conflict with t, then

$$\max(0, EFT_k - \theta), LFT_k - \theta$$

where EFT_k and LFT_k denote the lower and upper bounds of interval I for transition k, respectively;

c) All other transitions have their interval set to their Static Firing Interval.

In other words, the transitions not enabled by the new marking M' receive empty intervals; the transitions that remained enabled while transition t was firing have their intervals shifted towards the origin of times of the value θ of the time at which transition ft fired (restricted to non negative values); the remaining transitions (those enabled by M' and either in conflict with t for M, or not enabled by M) have their intervals set to their Static Firing Intervals. Time θ is relative to the moment at which state S has been reached.

For simplicity, we consider in this paper only Time Petri Nets such that none of their transitions may be enabled several times "simultaneously" by any marking, i.e. TPNs such that for any marking M and any transition t, some place p is such that $M(p) < 2.B(t, p)$. This restriction may be dropped by defining some meaning for multiple enabledness in these nets; some solutions are presented in [4, 9] but lack of space prohibits discussing these here.

The firing rule above defines a reachability relation among states of Time Petri Nets. Firing sequences may be defined for TPNs, as they are for Petri Nets, as sequences of successively firable transitions. A Firing Schedule will be a firable sequence of pairs (transition, time). The behavior of a TPN is characterized by the set of states reachable from its initial state or, alternatively, by the set of firing schedules feasible from its initial state.

Representing the behavior of a TPN by its set of states, as the behavior of a Petri Net is represented by its set of reachable markings, is generally not possible. This is because, as the time is continuous and as transitions may fire at any time in their allowed intervals, the states have in general an unbounded number of successors.

Before defining state classes, let us give a more convenient formalism for the states. A TPN state may be seen as a pair (M, D), in which M is a marking and D is a set of vectors called the Firing

Domain. Vectors in the Firing Domain have one component per transition enabled. Projection i of the domain is the interval currently associated with the i^{th} transition enabled. These domains may expressed as solution sets of some systems of inequalities, with variables one to one associated with the transitions enabled.

The initial State S0 of the net fig. 1 below, for instance, may be defined as the pair composed of the initial marking of the net and the domain D0 defined as follows:

$M0 : p1(1), p2(2)$

$D0$: Solution set of:

$4 \leq t_1 \leq 9$

Where variable t_1 is associated with transition $t1$, which is the sole transition enabled for $M0$.

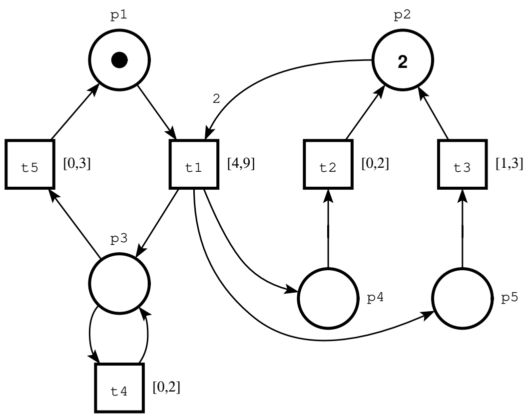


Figure 1: A Time Petri net

Firing transition $t1$ from the initial state, at a relative time θ_1 (comprised in the interval $[4, 9]$) leads to state $S1 = (M1, D1)$, with:

$M1 : p3(1), p4(1), p5(1)$

$D1$: Solution set of:

$0 \leq t_2 \leq 2$
$1 \leq t_3 \leq 3$
$0 \leq t_4 \leq 2$
$0 \leq t_5 \leq 3$

Time θ_1 does not appear in this system since no transition remained enabled while $t1$ was firing.

Firing transition $t2$, at a time θ_2 (in the interval $[0, 2]$) from state $S1$ above leads to the following state $S2 = (M2, D2)$:

$M2 : p2(1), p3(1), p5(1)$

$D2$: Solution set of:

$\max(0, 1 - \theta_2) \leq t_3 \leq 3 - \theta_2$
$0 \leq t_4 \leq 2 - \theta_2$
$0 \leq t_5 \leq 3 - \theta_2$

As time is continuous, θ_2 may take any value among the infinitly many contained in the interval $[0, 2]$. An infinity of different states may be reached by firing transition $t2$ from state $S1$.

3 STATE CLASSES AND THE ENUMERATION METHOD

State Classes

Rather than considering the state reached from the initial state by firing some feasible schedule, we will consider, for each firing sequence s, the set of all states reachable from the initial state by firing schedules with this firing sequence s. This set of states will be called the State Class associated with the firing sequence s.

More formally, the class associated with a firing sequence s is defined as the pair (M, D) in which M is the marking reached from the initial marking by firing sequence s and D is the union of all firing domains of states reachable from the initial state by firing schedules with firing sequence s.

Domains in the classes may be expressed as solution sets of systems of linear inequalities, with one variable per transition enabled by the marking of the class. Classes are pairs $C = (M, D)$, in which M is a marking and D is a firing domain defined as the solution set of some system of linear inequalities:

$A.\underline{t} \leq \underline{b}$

in which A is a matrix, $\underline{b}$ is a vector and variable $\underline{t_i}$ corresponds to the i^{th} transition enabled by marking M.

In practice, we are interested in computing recursively the set of classes, i.e. deriving the class associated with sequence $s.t$ from the class associated with sequence s, the initial class being defined as the class containing only the initial state. This is provided by the following firing rule.

The firing rule for state classes

A transition t is firable from a class $C = (M, D)$ iff both the following conditions hold:

(i) t is enabled;

(ii) There is in the domain D a vector in which the component corresponding to transition t is not greater than any other component.

The first condition is the usual one of enabledness. Domains in the state classes express a set of intervals for each transition enabled and, eventually, relationships between the firing times of these transitions. The second condition expresses that transition t is fired in its allowed interval, and that it is fired the first among the enabled transitions.

In the form of inequalities, D being the solution set of some system $A.\underline{t} \leq \underline{b}$ and t being the i^{th} transition enabled, this second condition is true iff the following system of inequalities is consistent:

$$A.\underline{t} \leq \underline{b}$$
$$\underline{t}_i \leq \underline{t}_j, \text{ for all variables } \underline{t}_j,\ j \neq i.$$

Computation of the successor class $C' = (M', D')$ is carried out as follows:

1) Compute new marking M' as in Petri Nets;

2) Compute new domain D' in four steps:

a) Augment the system $A.\underline{t} \leq \underline{b}$ with the above firability conditions for transition t;

b) Eliminate from this system the variables associated with transitions in conflict with t; these transitions are those enabled by M and not enabled by $M - B(t, -)$, where $B(t, -)$ is the Backward Incidence vector of transition t;

c) In this reduced system, express each remaining variable $\underline{t}_j$, with $j \neq i$, as the sum of variable $\underline{t}_i$ and a new variable $\underline{t}'_j$, and eliminate from the system all old variables, including $\underline{t}_i$;

d) In this new system, add one variable for each newly enabled transition, constrained to belong to the Static Firing Interval of the transition it is associated with. Newly enabled transitions are those enabled by M' and not enabled by $M - B(t, -)$.

Step a) corresponds to selecting from the starting domain all vectors such that the component corresponding to transition t is not greater than any other, i.e. transition t fires the first among the enabled transitions. Elimination of the variables at step b) does not affect the firing intervals of the remaining variables in the system, nor their relationships; elimination corresponds to a projection of the domain. The solution set of the system found at step c) may be seen as the firing domain for the transitions that remained enabled while t was firing, expressed with the time at which transition t fires as the new time origin. Step d) simply introduces intervals for the newly enabled transitions, equal to their respective static firing intervals.

Comparing classes for equality

Two classes will be defined equal iff both their markings are equal and their firing domains are equal. Comparing for equality the set of solutions of two systems of linear inequalities is generally costly but, as shown in the sequel, this comparison can be done efficiently in our particular case.

Normal Form Lemma: Firing domains of the state classes of T-Safe Time Petri Nets may be expressed as solution sets of systems of inequalities with the following general form:

$$a_i \leq \underline{t}_i \leq b_i, \text{ for all } i;$$
$$\underline{t}_j - \underline{t}_k \leq c_{jk}, \text{ for all } j,\ k \text{ with } j \neq k$$

where $\underline{t}_i$ is the variable associated with the i^{th} transition enabled by the marking of the class and a_i, b_i and c_{jk} are constants (some b_i or c_{jk} may be unbounded).

The proof is straightforward: initial domains satisfy this property and this general form is kept by the four transformations that together constitute the firing rule. QED

These systems admit canonical forms defined as:

$$a_i^* \leq \underline{t}_i \leq b_i^*, \text{ for all } i;$$
$$\underline{t}_j - \underline{t}_k \leq c_{jk}^*, \text{ for all } j,\ k \text{ with } j \neq k$$

Where a_i^*, b_i^* and c_{jk}^* are the smallest possible value for variable $\underline{t}_i$, the largest possible value for variable $\underline{t}_i$ and the largest possible difference between the values of variables $\underline{t}_j$ and $\underline{t}_k$, respectively.

A result in [1] implies that these canonical forms can be computed in polynomial time. Once the canonical forms for domains are derived, comparing domains for equality reduces to comparing their canonical forms for identity.

The graph of state classes

The reachability relation defined by the firing rule allows building a tree of state classes as follows: its root is the initial class and there is an arc labelled with transition t, going from class C to class C', iff transition t is firable from C and firing it leads to class C'. The graph of state classes is obtained from the tree by merging equal classes.

It is clear from the definition of the state classes that any sequence of transitions firable from the initial state will be a path in the above tree of classes. Further, existence of a path labelled s from the initial class to a class C implies that a firing schedule with firing sequence s is feasible from the initial state.

Let us illustrate the firing rule for classes by some firings in the net of fig. 1: The initial class $C0$ contains only the initial state $S0$ (given at the end of section 1). Firing transition $t1$ from this class at a relative time comprised in the interval $[4, 9]$ leads to a class $C1 = (M1, D1)$ equal to the state $S1$ previously computed since no transition remained enabled while $t1$ fired. Firing transition $t2$ from class $C1$ leads to a class $C2 = (M2, D2)$ with:

$$M2 : p2(1), p3(1), p5(1)$$

$$D2 : \text{Solution set of:}$$

$$\begin{aligned} &0 \leq t_3 \leq 3 \\ &0 \leq t_4 \leq 2 \\ &0 \leq t_5 \leq 3 \\ &t_4 - t_3 \leq 1 \\ &t_5 - t_3 \leq 2 \end{aligned}$$

System $D2$ gives the possible firing times of transitions $t3$, $t4$ and $t5$ with time origin being the moment at which transition $t2$ fired. Note that the firing times of transitions $t3$, $t4$ and $t5$ are tied together. The firing rule allows the enumeration of 12 state classes for this net.

Our intent is to use the graph of classes of a TPN for representing and analyzing its behavior. It follows from the definition of the classes that each can only have a bounded number of successors. So, for the graph of classes to have a bounded number of nodes, it suffices that the net does not admit an infinite length firing sequence going through a sequence of all different state classes. This property is investigated in the next section.

4 SOME PROPERTIES OF TIME PETRI NETS

Let us denote $R(M0)$ the set of markings of a TPN that can be reached from its initial marking. The Reachability problem is whether or not a given marking belongs to $R(M0)$ and the Boundedness problem is whether or not all markings in $R(M0)$ are bounded, i.e. are such that all of their components are smaller than some integer constant k.

Let us recall also that TPNs considered in this paper do not admit multiply enabled transitions. TPNs with this property are called T-Safe in the sequel. T-Safeness is easy to check incrementaly when enumerating classes. Theorem 1 below recalls an undecidability result for Time Petri Nets:

Theorem 1: The Reachability and Boundedness problems for Time Petri Nets are undecidable.

A direct proof is produced in reference [6]. Others proofs may be given: it may be shown that TPNs can simulate Inhibitors Nets or Priority Nets and have equivalent Reachability and Boundedness problems. Since these problems are known undecidable for these nets, it may be infered that they are also undecidable for Time Petri Nets.

A straightforward consequence of theorem 1 is that the finiteness of the set of classes of a TPN is undecidable, since classes are pairs (marking, domain). The following theorem 2 will help for stating sufficient conditions for this property.

Theorem 2: If Static EFTs and LFTs for all transitions are choosen among rational numbers, then the number of state classes of a T-Safe TPN is bounded if and only if the net is Bounded.

The proof is rather lengthy and will be only sketched here, more rigorous proofs may be found in references [4] and [9]. First, it must be shown that if Static EFTs and LFTs are choosen among rational numbers then the constants a_i^*, b_j^* and c_{jk}^* in the canonical forms of the domains are rational numbers and are either unbounded (and remain so by the firing rule) or admit upper and lower bounds in all domains that depend only on the Static EFTs and LFTs of the transitions. This implies that only a bounded number of distinct values may be computed for these constants. Further, the number of variables in these systems is boun-ded, since TPNs considered are T-Safe. Thus, only a bounded number of non equivalent systems of inequalities that characterize domains may be computed using the firing rule. Last, if the net is Bounded and T-Safe, then it admits both a bounded number of markings (consequence of the Boundedness property) and a bounded number of non equal domains. QED

The restriction that Static EFTs and LFTs be rational numbers was not made in [12]. This restriction, essential here, does not induce in practice any limitation. Assuming this restriction, any sufficient condition for Boundedness will provide a sufficient condition for the finiteness of the set of state classes of the net, and conversely. Some of these are investigated in the sequel.

The necessary and sufficient condition for Boundedness of Petri Nets and Vector Addition Systems [7] provides a sufficient condition for Boundedness of TPNs. This condition allows proving bounded a large class of TPNs but has been proven too weak for the applications we have in mind. The following is stronger.

Theorem 3: A T-Safe Time Petri Net is Bounded if no pair of state classes $C = (M, D)$ and $C' = (M', D')$ reachable from its initial state class are such that:

- **(i)** C' is reachable from C;
- **(ii)** $M' \geq M$ and $M' \neq M$;
- **(iii)** $D' = D$;

An unbounded TPN necessarily admits an infinite length firing sequence , going through a sequence s of all different state classes. Since the net is T-Safe, it admits only a bounded number of distinct firing domains and, as classes are pairs (marking, domain), the unbounded sequence s must contain an infinite length subsequence s' in which all markings are different and all domains are equal. Any pair of classes in this sequence satisfies (iii). Further, using [7], this sequence s' will necessarily contain two classes C and C' satisfying (i) with their markings satisfying (ii). QED

This condition, augmented with user defined semantic conditions for stopping enumeration as soon as possible if the behavior of the net is not as expected, have been proven adequate for most of the meaningfull examples we have treated so far. Stronger sufficient conditions are discussed in [4] and [9].

When a TPN has been proven bounded, its graph of state classes allows checking the specific properties that characterize its correct behavior, somewhat the same way as properties of Petri Nets are investigated using the reachability analysis. Further, liveness properties similar to those defined for Petri Nets may be defined for TPNs and, for bounded TPNs, proved using the graph of state classes.

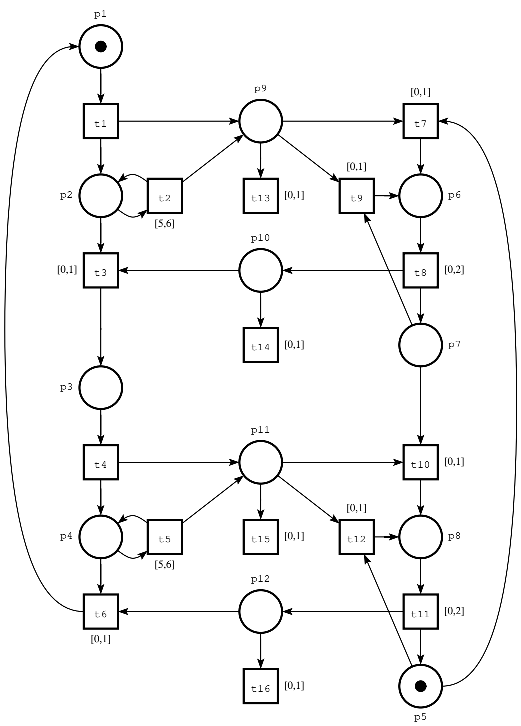
5 EXAMPLE, ANALYSIS OF TIME DEPENDENT COMMUNICATION PROTOCOLS

As mentionned earlier, communication protocols make a wide use of timing constraints in their specifications: recovery mechanisms for losses of messages are usually implemented using time outs. Time Petri Nets constitute a suitable tool for verifying that these time-outs are correctly set. The Alternating Bit Protocol will provide an illustrative example for the analysis method presented in section 2 and 3.

This protocol is a stop and wait data transfer protocol. Before sending a new message, the sender waits for the acknowledgement of the last message it sent. Hypotheses on the behavior of the transmission medium are that messages or their acknowledgements may be lost or damaged while in transit.

A mechanism is provided for recovering from these losses: a time-out is set when a message is sent and if its acknowledgment does not arrive in time the message is retransmitted. This basic mechanism is sufficient for recovering from losses of messages but does not prevent from duplication-free reception: if an acknowledgment is lost, the receiver is unable to decide whether the next message it receives is a new message or another copy of the last message it received. To solve this problem, messages are numbered prior to transmission with modulo-2 sequence numbers and acknowledgments refer explicitely to these numbers.

Fig. 2 below shows a TPN for the Alternating Bit Protocol. For simplicity, damaged messages are considered as lost. Notice that losses of messages and acknowledgements are simply represented with transitions that have no output places; there is no need here for any artificial mechanism relating lost messages to messages retransmitted.



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

Figure 2: A TPN for the Alternating Bit Protocol.

Estimates for the durations of all elementary actions of the protocols have been provided. Retransmission of a message occurs at a time comprised between 5 and 6 units after the last copy of the message has been sent. Equal estimates (between 0 and 1) are given for losses and receptions of messages and acknowledgments. No constraints, i.e. the intervals $[0, infty[$, are given for transmission of the first copies of the messages.

The net fig. 2. has been analysed with the help of an experimental computer package we developed. This net has been proven bounded, it admits sixteen classes. It is clear from these classes that only one message or acknowledgment will be in transit at a time (places $p9$, $p10$, $p11$ and $p12$ hold at most one token, in any marking). This assures that the retransmission time out is correctly set. Further, no duplicate message may be delivered (transitions $t7$ and $t10$ alternate in all paths of the graph) and the transfer of messages can actually occur (the net is live).

Among other communication systems we have analyzed is a bus allocation protocol, called REBUS, taken from [2]. REBUS is an experimental fault tolerant distributed system designed for real time control applications. The typical REBUS configuration is a set of functional units running application tasks linked together by a hardware bus through which they communicate. Units are organized in a virtual ring independently of the physical organization. Control of the bus is successively given to all units in the ring in a circular fashion. A broadcast message is used for transmitting the privilege.

Fault hypothesis are that messages may be damaged or lost, when transmitted or when received, and that units may become permanently "deaf" or permanently "dumb". Further, it is assumed that any fault is recovered before another occurs. Recovery of the faults is done using a watch dog based mechanism and private messages.

This bus allocation protocol has been specified and proven correct in [2], using Petri nets and structural analysis methods. A specification and a proof of correctness using TPNs and the analysis method introduced here is detailled in [10]. The interest of the exercise was in its significant complexity. For keeping manageable the number of state classes, we used a superposition like analysis method, analyzing separately the effects of each possible fault on the behavior of the system instead of building a global net with all possible faults represented in it.

6 CONCLUSION

The reachability analysis technique presented here has been proven adequate for most of the meaningfull examples we experimented with. However, its limitations must be kept in mind.

A first limitation is that no necessary and sufficient condition for boundedness can be devised for Time Petri Nets, and proving this property is necessary for acheiving the reachability analysis. Little can be done to overcome this limitation, except devising stronger sufficient conditions.

The second limitation, also typical of the reachability analysis technique for usual Petri Nets, is that, even if bounded, the number of state classes of a Time Petri Net may be very large. Care must be taken, with a careful proof methodology, to keep the number of state classes manageable.

Alternative methods, avoiding partially or totally the enumeration phase, such as reduction methods or structural analysis [3, 8] are being investigated. Also being investigated is extending the field of application of the method towards performance analysis.

REFERENCES

[1] B. ASPVALL, Y. SHILOACH, "A Polynomial Time Algorithm for Solving Systems of Linear Inequalities with two Variables per Inequality", 20th Annual Symp. on Foundations of Computer Sciences, Oct. 1979, 205-217.

[2] J. M. AYACHE, J. P. COURTIAT, M. DIAZ, "REBUS, A Fault-Tolerant Distributed System for Industrial Real-Time Control", IEEE Trans. on Computers, vol. C-31, no 7, July 1982, 637-647.

[3] P. AZEMA, B. BERTHOMIEU, P. DECITRE, "The Design and Validation by Petri Nets of a Mechanism for the Invocation of Remote Servers", IFIP Congress 1980, Melbourne, Australia, North Holland Pub. Co., 1980, 599-604.

[4] B. BERTHOMIEU, M. MENASCHE, "A State Enumeration Approach for Analyzing Time Petri Nets", 3rd European Workshop on Applications and theory of Petri Nets, Varenna, Italy, Sept. 1982, 27-65.

[5] P. CASPI, N. HALBWACHS, "Analyse Approchee du Comportement Asymptotique de Systemes Temporises", Laboratoire d'Informatique et de Mathematiques Appliquees de Grenoble, RR 322, Sept. 1982.

[6] N.D. JONES, L.H. LANDWEBER, Y.E. LIEN, "Complexity of some problems in Petri Nets", Theoretical Computer Science 4, 1977, 277-299.

[7] R. M. KARP, R. E. MILLER, "Parallel Program Schemata", Journal of Computer and System Sciences 3, 1969, 147-195.

[8] K. LAUTENBACH, H.A. SCHMIDT, "Use of Petri Nets for Proving Correctness of Concurrent Systems", IFIP Congress 1974, Amsterdam, Netherlands, North Holland Pub. Co., 1974, 187-191.

[9] M. MENASCHE, "Analyse des Reseaux de Petri Temporises et Application aux Systemes Distribues", These Dr.Ing., Universite Paul Sabatier, Toulouse, Nov. 1982.

[10] M. MENASCHE, B. BERTHOMIEU, "Time Petri Nets for Analyzing and Verifying Time Dependent Communication Protocols", 3rd IFIP/WG6.1 International Workshop on Protocol Specification, Testing and Verification, Zurich, Switzerland, May 1983.

[11] P. MERLIN, D. J. FARBER, "Recoverability of Communication Protocols", IEEE Trans. on Communications, vol. COM-24, no. 9, Sept. 1976, 1036-1043.

[12] P. MERLIN, "A Study of the Recoverability of Computer Systems", Ph.D. Thesis, Univ. Of California, Irvine, 1974.

[13] C. RAMCHANDANI, "Analysis of Asynchronous Concurrent Systems by Timed Petri Nets", Project MAC, TR 120, Massachussets Institute of Technology, Feb. 1974.