JANUARY 1987 LIDS-P-1639

TIMED EVENT-GRAPH AND PERFORMANCE EVALUATION OF SYSTEMS*

by

Herve P. Hillion**

Alexander H. Levis***

ABSTRACT

Performance analysis of Timed Event-Graphs, including both deterministic and random models, is considered. First, a bound to the average firing rate in steady-state is given. This bound is computed using the critical circuits of the net, for which the average cycle time is maximal. The second result deals with an extended deterministic model, in which the transition processing times are a function of the number of firing repetitions. A fast and simple algorithm is described that determines the earliest firing schedule.

*This work was carried out at the MIT Laboratory for Information and Decision Systems with support provided by the Joint Directors of Laboratories under Contract No. N00014-85-K-0782 and by the Office of Naval Research under Contract No. N00014-84-K-0519.

**INRIA-LORRAINE, Chateau du Montet, rue du Doyen Roubault, 54500 Vandoeuvre, FRANCE, Tel: 83 55 44 98.

***Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA, 02139, USA, Tel: (617) 253-7262.

1. INTRODUCTION

Timed Petri Nets are used to analyse the dynamic behavior of systems with asynchronous and concurrent processing. So far, two models of Timed Petri Nets have been studied: deterministic models [1,2,3] in which the transition firing times are assumed to be fixed, and probabilistic models [4,5,6], where the firing times are specified by probabilistic distribution functions, generally assumed to be exponential distributions. In both cases, methods are developed to evaluate the steady-state performance. A more general approach is proposed in this paper. The transition firing times are considered to be dependent on the number of firing repetitions. The method can handle a sequence of successive firing times for every transition, assuming the firings to be repetitive over time. The only other necessary assumption is that an average firing time exist for each transition, regardless of any other assumption concerning the sequence of repetitive firings which may be either deterministic or random. The analysis is restricted, however, restricted to a special class of Petri Nets, namely Event-Graphs [7] and is focused on obtaining performance measures. The results presented in this paper have been used to evaluate the performance of Decisionmaking Organizations modeled by Petri Nets [8].

This paper is divided in five sections. In Section 2, the assumptions of the model, called Repetitive Timed Event-Graph, are presented. In Section 3, an upper bound to the average firing rate is computed, that only depends on the average firing times of the transitions. In Section 4, the deterministic case is considered and a fast and simple algorithm that determines the firing schedule is described. Finally, concluding remarks are given in the last section.

2. REPETITIVE TIMED EVENT-GRAPH

Recall first that an Event Graph [7] (also known as Marked Graph [10]) is a Petri Net [9] such that each place has exactly one input and one output transition. Given an initial distribution of tokens in the net (i.e., an initial marking), it is shown in [10] that an Event-Graph is live if and only if the number of tokens in every directed elementary circuit is strictly positive. We assume here that this condition is satisfied so that each transition can fire repeatedly any number of times.

In Timed Petri Nets, each transition t takes a "real" time $\mu(t)$ to fire. When a transition t is enabled, a firing can be initiated by removing one token from each of t's input places. The tokens remain in transition t for the firing execution during the time $\mu(t)$ and then the firing terminates by adding one token in each of t's output places.

Different models of Timed Petri Nets have been studied; in deterministic Timed Petri Nets [1.2] and Timed Event-Graphs [3], a positive (rational) number is assigned to each transition t of the net, which defines the firing time $\mu(t)$. In stochastic Petri Nets [4,5], the firing times are assumed to be random variables that are exponentially distributed.

In this paper, the transition firing times are not fixed, but may be different from one firing to the next. Therefore, a sequence of successive firing times $\{\mu_t(1),\mu_t(2),\ldots,\mu_t(k),\ldots\}$ (for any transition t) is constructed according to the number of firing repetitions. There is no assumption regarding this sequence, except that the following limit

$$\lim_{n \to \infty} \frac{\sum_{k=1}^{n} \mu_t(k)}{n} = \bar{\mu}_t \tag{1}$$

must exist and be finite (for any transition t). The limit, denoted by $\bar{\mu}_t$, determines the average (or mean) firing time of transition t. Two cases will be considered.

First, in the non-deterministic case, the sequence of successive firing times can be regarded as possible outcomes of the random variable μ_t, with mean value $\bar{\mu}_t$ (regardless of the probability distribution function). For instance, μ_t can be a discrete random variable that takes on a finite set of possibles values $\{v_1,v_2,\ldots,v_j\}$ according to a certain probability distribution $\{\gamma_1,\ldots,\gamma_j\}$. In that case:

$$\bar{\mu}_t = \sum_{k=1}^{j} \gamma_k \, v_k$$

This assumption was used by the first author [8] to evaluate the performance of Decisionmaking Organizations modeled by Event-Graphs. In that model, the processing of a task can be performed by different algorithms, each having a fixed execution time. At each occurence of the task, an algorithm is selected, according to a fixed probability distribution: such a decision rule is called a decision switch [11] in the Petri Net model.

The second case is the deterministic case, when the sequence of successive firing times is assumed to be fixed (for all transitions). In Flexible Manufacturing Systems, for instance, transitions can model machines [12]. If it is assumed that a machine can process different types of jobs, according to a given sequencing of the jobs, then the sequence of successive processing times is fixed.

In order to study the performance of Event-Graphs, it is assumed in this paper that the Event-Graph model is strongly connected and live. These assumptions ensure that the transition firings can be repeated any number of times and that the net is bounded [13]. The following notation will be used:

$T = \{t_1,t_2,\ldots,t_m\}$ is the set of transitions

$\mu_i(n)$ denotes the n-th firing time of transition t_i (i.e., when t_i fires for the n-th time)

$S_i(n)$ denotes the instant at which the n-th firing initiation of transition t_i occurs.

The initial distribution of tokens (i.e., an initial marking) is given at $\tau = 0$ and is denoted by M^0. The dynamic behavior of the system will be described by the sequence $S_i(n)$ for $i=1,2,\ldots,m$ and $n=1,2,\ldots$, which will also be called the firing schedule. Since it is assumed that transitions fire as soon as they are enabled, the performance obtained is the maximum performances (with respect to time) and the schedule is the earliest firing schedule.

3. PERFORMANCE EVALUATION

In this section, an upper bound to the performance in steady-state is obtained. The performance measure considered is the average period π_i with which any transition t_i fires, i.e.,

$$\pi_i = \lim_{n \to \infty} \frac{\sum_{k=1}^{n} (S_i(k) - S_i(k-1))}{n} \tag{2}$$

Quite obviously, $1/\pi_i$ determines also the average firing rate of transition t_i. The average period π_i can also be written as:

$$\pi_i = \lim_{n \to \infty} \frac{S_i(n)}{n} \tag{3}$$

Since the Event-Graph is assumed to be strongly connected, the average period is the same for all transitions of the net and will be denoted by π. This is trivially deduced by the fact that the number of tokens in any directed elementary circuits is invariant with any transition firing [10]. A directed elementary circuit is a directed path that goes from one node (place or transition) back to itself and such that none of the nodes are repeated.

The average cycle time of any directed elementary dircuit ρ, denoted by $C(\rho)$, is defined as the sum of the average firing times of all transitions belonging to the circuit divided by the number of tokens in the circuit

$$C(\rho) = \frac{\sum_{i \varepsilon \rho} \bar{\mu}_i}{M} \tag{4}$$

Let C_{max} be the maximum over all directed elementary circuits of the average cycle times. C_{max} will be called the maximum average cycle time.

$$C_{max} = \max_{\rho} \; (C(\rho)) \tag{5}$$

Then the following result holds:

Theorem: The maximum average cycle time is a lower bound of the average period, i.e.,

$$\pi \geq C_{max} \tag{6}$$

This theorem generalizes the result obtained by Ramchandani [1] in which all transition firing times are constant. In order to prove this result, we consider any directed elementary circuit ρ and prove that $\pi \geq C(\rho)$.

Proof: Let ρ be any circuit of the net, that we denote, without loss in generality, by $\rho = (t_1 \; p_1 \; t_2 \ldots t_r \; p_r)$. Let M_i^0 denote the initial marking of place p_i. If we consider any two transitions (t_i, t_{i+1}) connected by place p_i, then for any positive integer n, we know that (see [1,3]).

$$S_{i+1}(n+M_i^0) \geq S_i(n) + \mu_i(n) \tag{7}$$

This means that at the instant:

$$\tau = S_i(n) + \mu_i(n)$$

transition t_i has "produced" exactly n tokens in place p_i (since t_i has fired exactly n times since the initial instant). Since there were initially M_i^0 tokens in this place, the total number of tokens available in p_i during the interval of time $[0,\tau]$, is precisely $n + M_i^0$. Now, each time that the firing of t_{i+1} was initiated, one token in p_i was "consumed". Accordingly, the number of firing initiations of t_{i+1} during the interval $[0,\tau]$ cannot be more than $(n + M_i^0)$:

$$S_{i+1}(n+M_i^0) \geq \tau = S_i(n) + \mu_i(n)$$

Suppose now that n is large enough so that $n > M_i^0$ for $i = 1,2,\dots,r$. Then (7) can be written as:

$$S_{i+1}(n) \geq S_i(n-M_i^0) + \mu_i(n-M_i^0) \quad \text{for} \quad i=1,2,\dots,r \tag{8}$$

If we apply recursively (8) from $i = r$ to $i = 1$, we obtain:

$$S_1(n) = S_{r+1}(n) \geq S_1(n-N_t) + (\sum_{i=1}^{r} \mu_i(n-M_i^0)) \tag{9}$$

where N_t denotes the total number of tokens in the circuit, as determined by:

$$N_t = \sum_{i=1}^{r} M_i^0$$

Let us now write n in the form:

$$n = K\, N_t + s$$
$$0 \leq s \leq N_t$$

Then, we deduce from (9):

$$S_1(n) \geq S_1(s) + \sum_{k=1}^{K} \left(\sum_{i=1}^{r} \mu_i(KN_t+s-M_i^0)\right) \tag{10}$$

Applying (10) for $s = 0,1,2,\dots,N_t-1$ and summing the inequalities yields:

$$\sum_{s=0}^{N_t-1} S_1(KN_t+s) \geq \sum_{s=0}^{N_t-1} S_1(s) + \sum_{s=0}^{N_t-1} \left(\sum_{k=1}^{K} \left(\sum_{i=1}^{r} \mu_i(kN_t+s-M_i^0)\right)\right) \tag{11}$$

Now:

$$\sum_{s=0}^{N_t-1} \left(\sum_{k=1}^{K} \left(\sum_{i=1}^{r} \mu_i(kN_t+s-M_i^0)\right)\right) = \sum_{i=1}^{r} \left(\sum_{q=0}^{KN_t-1} \mu_i(q+N_t-M_i^0)\right)$$

Finally, dividing (11) by KN_t and taking the limit when K goes to infinity, we obtain

(1) $$\lim_{K\to\infty} \frac{1}{KN_t} \left(\sum_{s=0}^{N_t-1} S_1(KN_t+s)\right) = \sum_{s=0}^{N_t-1} \left(\lim_{K\to\infty} \frac{S_1(KN_t+s)}{KN_t}\right) = N_t\ \pi$$

given (3)

(2) $$\lim_{K\to\infty} \frac{\sum_{s=0}^{N_t-1} S_1(s)}{KN_t} = \frac{1}{N_t} \sum_{s=0}^{N_t-1} \left(\lim_{K\to\infty} \frac{S_1(s)}{K}\right) = 0$$

$$\lim_{K\to\infty} \frac{1}{KN_t} \left(\sum_{i=1}^{r} \left(\sum_{q=0}^{KN_t-1} \mu_i(q+N_t-M_i^0)\right)\right) = \sum_{i=1}^{r} \left(\lim_{n\to\infty} \frac{1}{n} \left(\sum_{q=0}^{n-1} \mu_i(q+N_t-M_i^0)\right)\right)$$

$$= \sum_{i=1}^{r} \bar{\mu}_i \qquad \text{given (2)}$$

Hence:

$$N_t\ \pi \geq \sum_{i=1}^{r} \bar{\mu}_i$$

which implies:

$$\pi \geq \frac{\sum_{i=1}^{r} \bar{\mu}_i}{N_t} = C(\rho)$$

Q.E.D.

The result obtained in Theorem 1 gives an upper bound to the average firing rate, since (6) can be written:

$$\frac{1}{\pi} \leq \frac{1}{C_{max}} \tag{12}$$

This bound can be computed quite easily, once all circuits are determined. A simple method for obtaining all circuits is described in [8], using an algorithm that determines the invariants of a net [14]. It is particularly interesting that the upper bound of the maximum performance only depends of the transition mean firing times (given the initial distribution of tokens), regardless of any other assumptions concerning the firing times (which may be either deterministic or random variables with any type of probability distribution). Recall that (12) is shown to be an equality (i.e., $\pi = C_{max}$) when the transition firing times are constant [1], and that the steady-state is then K-periodic [3].

In computing C_{max}, it is also interesting to determine the critical circuits, i.e., those circuits ρ for which $C(\rho) = C_{max}$. It turns out that only these circuits bound the average firing rate in steady-state. Accordingly, the critical circuits should be the ones to modify (in terms of transition firing times or number of tokens in the circuit) so as to improve the performance of a system.

We are now going to determine the firing schedule, assuming that the sequences of successive firing times are known, i.e., the firing process is deterministic.

4. DETERMINATION OF THE FIRING SCHEDULE

We present in this section a fast algorithm to compute the firing schedule, i.e., the sequence $(S_i(n))$ $i=1,\ldots,m$; $m=1,2,\ldots$, when the system is deterministic. An initial distribution of tokens is assumed to be given at $\tau = 0$ and the sequence of successive firing times $\mu_i(n)$ $n=1,2,\ldots$ is assumed to be fixed for any transition t_i. We first determine the order with which transitions fire.

4.1 Partial Firing Order

Given the structure of the net and the initial distribution of tokens, some transitions fire sequentially and other fire concurrently. In order to psecify how the firings occur in the process, we proceed as follows. We call marked places the places which contain one token. We first consider the set P_1 of places that contain at least one token, i.e., with an initial marking strictly positive, and take this set as the initial set of marked places. Then let T_1 be the set of transitions that are enabled by the places of P_1. Assume that all transitions of T_1 fire once and let P_2 be the new set of marked places. Let T_2 be the set of transitions that are enabled by the places of P_2 and which do not have already fired (i.e., do not belong to T_1). Let P_3 be the set of marked places once all transitions of T_2 have fired. In a similar way, we then consider the set T_3 of transitions that are enabled by P_3 and which do not have already fired (i.e., do not belong to $T_1 U T_2$) and so on. We repeat this operation s times until $T_{s+1}=0$. Because the net is a live Event-Graph, the sequence $T_1, T_2, \ldots, T_s$ so constructed verifies:

$$T = T_1 \text{ U } T_2 \ldots \text{U } T_s$$

where T denotes the set of all transitions of the net. This sequence determines therefore the partial firing order between the transition firings. In particular, the transitions belonging to T_i (i=1,...,s) are transitions that fire concurrently at the i-th step. For that reason, the sequence $T_1, T_2, \ldots, T_s$ will be called the sequence of concurrent transitions. It should be clear that this sequence is fully determined by the structure of the net, and the initial distribution of tokens.

4.2 Firing Schedule

For clarity, we now assume that the transitions are labeled according to the sequence $T_1, \ldots, T_s$, i.e.,

$T_1 = \{t_1, t_2, \ldots, t_{k1}\}$

$T_2 = \{t_{k1+1}, t_{k1+2}, \ldots, t_{k2}\}$

$\cdot \; \cdot \; \cdot$

$T_s = \{t_{ks+1}, \ldots, t_m\}$

where m denotes the total number of transitions in the net. Note that the order between the transitions belonging to T_i $(i=1,\ldots,s)$ does not matter, since the transitions fire concurrently.

For any transition t_j we now denote by $Inp(t_j)$ the set of all transitions which are the <u>input</u> transitions of all <u>input</u> places of t_j, as shown on Figure 1; $\{p_{i1}, p_{i2}, \ldots, p_{ir}\}$ are all input places of transition t_j and $\{t_{i1}, t_{i2}, \ldots, t_{ir}\}$ is the set of input transitions of each of these places. If $t_i \in Inp(t_j)$, we also denote by M^0_{ij} the initial marking of the unique place p_{ij} whose input transition is t_i and output transition t_j.

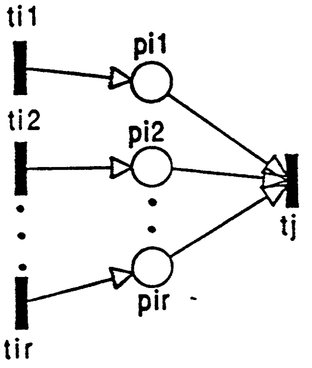

Figure 1. Definition of $Inp(t_j)$

For any transition t_j and any transition $t_i \in Inp(t_j)$ we have:

$$S_j(n+M^0_{ij}) \geq S_i(n) + \mu_i(n) \tag{13}$$

where n is any positive integer. The inequality follows directly from (7)(see previous section). Suppose that $n > M^0_{ij}$. Then (13) can be written as:

$$S_j(n) \geq S_i(n-M^0_{ij}) + \mu_i(n-M^0_{ij}) \tag{14}$$

For simplicity, let us assume $S_i(k) = 0$ and $\mu_i(k) = 0$ if $k \leq 0$ (for $i=1,2,\dots,m$), so that (14) is still satisfied if $n \leq M^0_{ij}$; this means that there are still tokens left in the place p_{ij} from the initial marking, so that the firing initiations of t_j do not depend on the firing terminations of t_i.

Given the assumption that the firings occur as soon as the transitions are enabled, we deduce from (14) the following relation:

$$S_j(n) = \max_{t_i \varepsilon \mathrm{Inp}(t_j)} (S_i(n-M^0_{ij}) + \mu_i(n-M^0_{ij})) \tag{15}$$

Equation (15) can now be used to compute the firing schedule by iteration on n, the number of firing repetitions. At the n-th iteration, we can compute $S_j(n)$ ($j=1,2,\dots,m$) once we know both $S_i(k)$ for $i=1,2,\dots,m$ and $k=1,2,\dots,n-1$ and $S_i(n)$ for $i=1,2,\dots,j-1$. Consider indeed any transition $t_i \ \varepsilon \ \mathrm{Inp}(t_j)$. If $M^0_{ij} > 0$, we then know $S_i(n-M^0_{ij})$ (computed at a previous interation). Now if $M^0_{ij} = 0$, it means that in the firing order t_i should necessarily fire before t_j. Therefore if $t_j \ \varepsilon \ T_1$ then, by the sequence $T_1, T_2, \dots, T_S$, it follows that $i < j$. Finally, if $t_j \ \varepsilon \ T_1$ then, by construction, M^0_{ij} is always strictly positive whenever $t_i \ \varepsilon \ \mathrm{Inp}(t_j)$.

The algorithm can now be described recursively in a very simple way:

```
{Initialization}
$p = \mathrm{Max}\ (M^0_{ij})$
for $k = -p$ to $0$ set $S_i(k) = 0$ and $\mu_i(k) = 0$  {for $i=1,\dots,m$}
$n = 1$
Repeat {main loop}
  For $j = 1$ to $m$ set
    $S_j(n) = \max_{t_i \varepsilon \mathrm{Inp}(t_j)} (S_i(n-M^0_{ij}) + \mu_i(n-M^0_{ij}))$
End
until $n = N$ (total number of firing repetitions).
```

Recall that there are two steps to complete in order to use the algorithm:

(1) Determine the firing order of the transitions, following the method described in Section 4;

(2) Determine the set $\mathrm{Inp}(t_j)$ for any transition t_j (deduced from the structure of the net).

This algorithm has been used to compute the firing schedule of Decisionmaking Organizations, using a Timed Event-Graph model [8]. It should be noted that, in this approach, we do not need to describe the states of the system nor use graph theory (as is done in the algorithm proposed in [3] for constant firing times).

5. CONCLUSION

We have developed in this paper some techniques for analyzing real-time systems that can be modeled using Event-Graphs. The result presented in Section 3 generalizes the result obtained in [1] for constant transition firing times. It can be used for preliminary performance evaluation of a general system for which only the average task execution times are known. It holds, in particular, without the restrictive assumptions of the probabilistic models studied so far. We have also extended the deterministic case, allowing a different firing time at each repetition. The algorithm described in section 4 provides a simple way to obtain the precise firing schedule for this case without the need for simulation.

REFERENCES

[1] Ramchandani, C. (1974). "Analysis of asynchronous concurrent systems by Timed Petri Nets", Technical Report No. 120, Laboratory for Computer Science, MIT, Cambridge, MA.

[2] Sifakis, J. (1980). "Performance Evaluation of Systems using Nets", Net Theory and Applications, Lecture Notes in Computer Science, Springer-Verlag, Berlin, FRG, pp. 307-319.

[3] Chretienne, P. and Carlier, J. (1984). "Modeling scheduling problems with Timed Petri Nets," Advances in Petri Nets, 1984, Lecture Notes in Computer Science, No. 188, Springer-Verlag, Berlin, FRG, pp. 62-82.

[4] Zuberek, V. M. (1985). "M-Timed Petri Nets, priorities, preemptions, and performance evaluation of systems," Advances in Petri Nets, 1985, Lecture Notes in Computer Science, No. 222, Springer-Verlag, Berlin, FRG, pp. 478-498.

[5] Molloy, M.K., (1982). "Performance analysis using stochastic Petri Nets," IEEE Trans. on Computers, Vol. 31, No. 9, pp. 913-917.

[6] Wiley, R.P. (1986). "Performance analysis of stochastic Timed Petri Nets," Ph.D. Thesis, Report LIDS-TH-1525, Laboratory for Information and Decision Systems, MIT, Cambridge, MA.

[7] Brams, G.V. (1983). Reseaux de Petri: theorie et pratique, Masson, Paris, France.

[8] Hillion, H. P. (1986). "Performance evaluation of Decisionmaking Organizations using Timed Petri Nets," M.S. Thesis, Report LIDS-TH-1590, Laboratory for Information and Decision Systems, MIT, Cambridge, MA.

[9] Peterson, J. L. (1981). Petri Net theory and the modeling of systems, Prentie-Hall, Englewood Cliffs, NJ.

[10] Commoner, F., A. W. Holt, S. Even, and A. Pnueli, (1971). "Marked Directed Graphs," Journal of Computer and System Sciences, Vol. 5, No. 5, pp. 511-523.

[11] Tabak, D. and A. H. Levis (1985). "Petri Net representation of decision models," IEEE Trans. on Systems, Man and Cybernetics, Vol. 5, SMC-15, No. 6, pp. 812-818.

[12] Cohen, G., D. Dubois, J.P. Quadrat and M. Viot (1985). "A linear-system theoretic view of discrete event processes and its use for performance evaluation in manufacturing," IEEE Trans. on Automatic Control, Vol. Ac-30, No. 3, pp. 210-220.

[13] Sifakis, J. (1978). "Structural properties of Petri Nets," Mathematical Foundations of Computer Science, Lecture Notes in Computer Science, No. 64, Springer-Verlag, Berlin, FRG, pp. 474-483.

[14] Martinez, J. and M. Silva (1980). "A simple and fast algorithm to obtain all invariants of a generalized Petri Net," Lecture Notes in Computer Science, No. 52, Springer-Verlag, Berlin, pp. 302-310.