

TIMED PETRI NET MODELS OF FLEXIBLE MANUFACTURING CELLS

W.M. Zuberek

W. Kubiak

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

Abstract

It is shown that a class of flexible manufacturing cells can be conveniently modeled and evaluated by timed Petri nets. For simple schedules, the modeling nets are covered by conflict-free invariant subnets, so the performance of the model is determined by the performance of its subnets. For composite schedules, the invariant subnets are free-choice, so a more elaborate approach to evaluating the performance of the model must be used.

1. INTRODUCTION

Flexible manufacturing systems are discrete-event systems which are composed of (i) a set of versatile machines, (ii) an automatic transportation system, and (iii) a decision-making system which determines what has to be done, when and where [SV89]. Machines are often grouped into manufacturing cells (or robotic cells), in which a robot performs sequences of pickup, move, load, unload and drop operations, transporting the manufactured parts from one machine of the cell to another [SSSBK92,Cl83]. The throughput of the cell depends on the sequence of robot activities as well as on the sequence in which different parts enter the cell [DH90]. The problem of maximizing the throughput of a robotic cell can thus be considered as a scheduling problem.

The behavior of flexible manufacturing systems is represented by 'events' and 'activities'; an event corresponds to a change of system's state while an activity corresponds to an operation performed by a machine. Different sets of activities determine the states of the system. In each state, several activities can occur concurrently, for example, several machines can perform their operations simultaneously and the robot can also transport a part. Petri nets [Re85,Mu89] provide a simple and convenient formalism for modeling systems of events which can occur concurrently, but there are constraints on precedence and frequency of such occurrences. In fact, one of the very first applications of Petri net models was to analyze production schemata [Ha72]. However, the complexity of real-life systems as well as rather difficult and not well-understood nature of concurrency seem to have caused a rather limited popularity of these models [Su85].

In order to study performance aspects of Petri net models, the duration of activities must also be taken into account and included into model specifications. Several types of Petri nets 'with time' have been proposed by assigning 'firing times' to the transitions or places of a net. In timed nets [Zu91], transition firings are 'real-time' events, i.e., tokens are removed from input places at the beginning of the firing period, and they are deposited to the output places at the end of this period (sometimes this is also called a 'three-phase' firing mechanism). The firing times may be

either deterministic or stochastic, i.e., described by some probability distribution function. In both cases the concepts of state and state transitions has been formally defined and used in derivation of different performance characteristics of the model.

This paper investigates timed Petri net models of flexible manufacturing cells. Two types of schedules are considered. For simple schedules, in which exactly one part enters and one leaves the cell in each cycle, the net model is composed of elementary conflict-free invariant subnets. The performance (i.e., the cycle time or the throughput [Ki90]) of the whole model is thus determined by the subnet with the maximum cycle time. For composite schedules, several parts enter and leave the cell in each cycle. Models of such schedules can be obtained by combining simple schedules in such a way that the liveness of the model is preserved. Invariant subnets of composite net models are free-choice nets, so their cycle times depend upon relative frequencies of transition firings. These relative frequencies of firings can be obtained from the T-invariants of the model.

The paper is composed of 3 main sections. Section 2 describes timed Petri net models of manufacturing cells with simple schedules. Models of composite schedules are introduced in Section 3, and Section 4 presents invariant analysis applied to net models of manufacturing cells.

2. MODELS OF MANUFACTURING CELLS

A simple manufacturing cell composed of three machines and a robot is sketched in Fig.1; the machines are denoted by M_1 , M_2 and M_3 , *In* represents a conveyor bringing the parts in while *Out* a conveyor for outgoing parts.

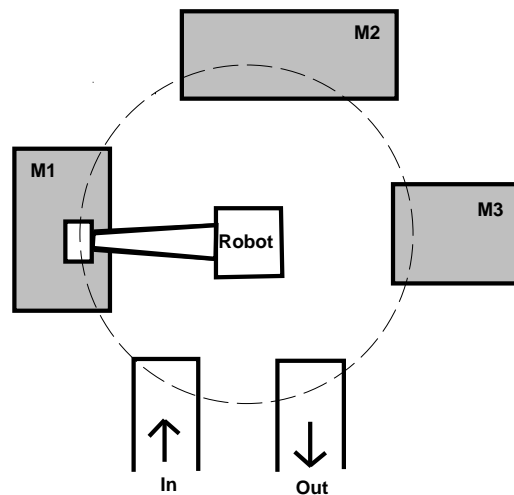


Fig.1. Layout of a three-machine cell.

Let (for simplicity) all the parts move from *In* to *M*₁, then from *M*₁ to *M*₂, from *M*₂ to *M*₃, and finally from *M*₃ to *Out*. The parts are moved between machines by the robot which follows a cyclical pattern of actions called its schedule. For ‘simple schedules’, exactly one (new) part enters and one leaves the cell in each cycle (and the leaving part may not be the same as the entering one). It has been shown [SSSBK92] that an *m*-machine cell has *m!* different simple schedules. The six simple schedules for a three-machine cell are shown in Fig.2(a) to 2(f).

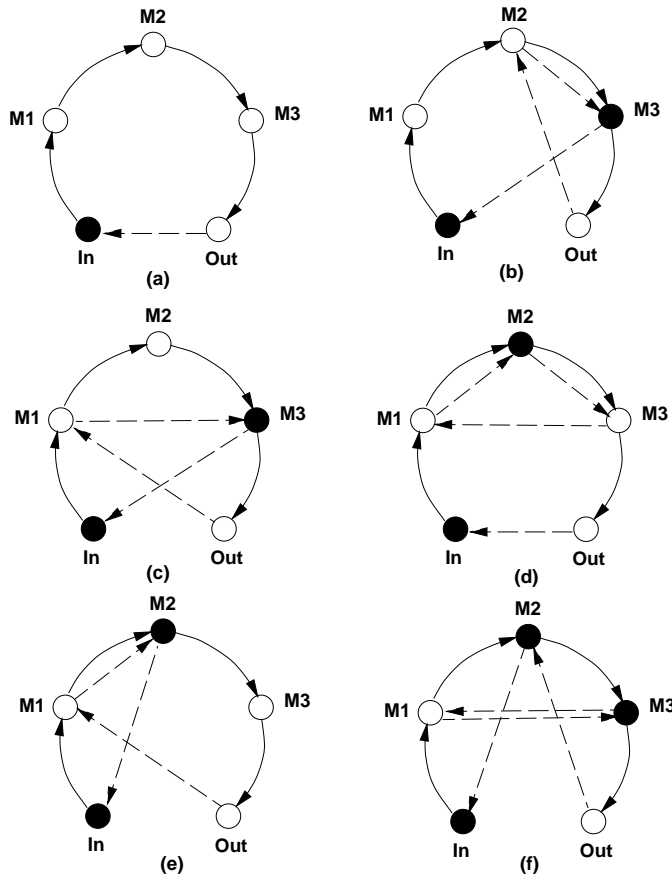


Fig. 2. Six simple schedules for a three-machine cell.

The schedules in Fig.2 are shown in the moments of time in which a part is picked from the input conveyor; the placement of parts in the cell is indicated by black circles; for example, for schedule D, whenever a part is picked from input, another part is being processed by machine *M*₂.

Denoting the robot moves from *X* to *Y* by $X \Rightarrow Y$ if the robot carries a part and by $X \rightarrow Y$ otherwise, the schedules from Fig.2 are as follows:

- A: $In \Rightarrow M_1 \Rightarrow M_2 \Rightarrow M_3 \Rightarrow Out \rightarrow In$
- B: $In \Rightarrow M_1 \Rightarrow M_2 \rightarrow M_3 \Rightarrow Out \rightarrow M_2 \Rightarrow M_3 \rightarrow In$
- C: $In \Rightarrow M_1 \rightarrow M_3 \Rightarrow Out \rightarrow M_1 \Rightarrow M_2 \Rightarrow M_3 \rightarrow In$
- D: $In \Rightarrow M_1 \rightarrow M_2 \Rightarrow M_3 \rightarrow M_1 \Rightarrow M_2 \rightarrow M_3 \Rightarrow Out \rightarrow In$
- E: $In \Rightarrow M_1 \rightarrow M_2 \Rightarrow M_3 \Rightarrow Out \rightarrow M_1 \Rightarrow M_2 \rightarrow In$
- F: $In \Rightarrow M_1 \rightarrow M_3 \Rightarrow Out \rightarrow M_2 \Rightarrow M_3 \rightarrow M_1 \Rightarrow M_2 \rightarrow In$

A timed Petri net model of the cell from Fig.1 with the schedule of Fig.2(a) is shown in Fig.3. In timed models, transitions represent operations and places ‘conditions’ (in the most general sense); the three machines (or rather machine operations) are represented by t_1, t_2 and t_3 , each transition with its input and output place (for ‘part loaded’ and ‘machine operation finished’ conditions). The

‘firing times’ associated with these transitions $f(t_1) = o_1, f(t_2) = o_2$ and $f(t_3) = o_3$ represent the (average) times of performing the operations on machines *M*₁, *M*₂ and *M*₃, respectively.

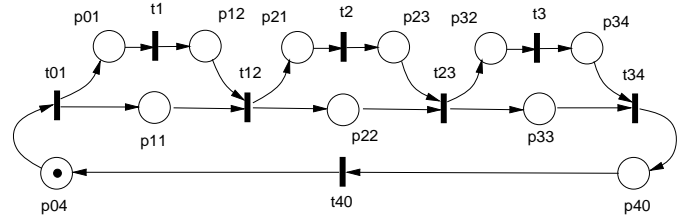


Fig.3. Petri net model of schedule A (Fig.2(a)).

The robot actions are represented by a path $t_{01}, t_{12}, t_{23}, t_{34}, t_{40}$, with the following interpretation; the ‘execution’ times are given in parentheses assuming that ‘*a*’ denotes the (average) pickup time, ‘*b*’ the (average) unload time, ‘*c*’ the (average) load time, ‘*d*’ the (average) drop time and ‘*e*’ the average ‘travel’ time between adjacent machines (assuming, for simplicity, that it is the same for all adjacent machines, and also the same for *M*₃ to *Out*, *Out* to *In* and *In* to *M*₁ moves):

- t_{01} – pick a part from *In*, move to *M*₁ and load ($a + c + e$)
- t_{12} – unload *M*₁, move to *M*₂ and load ($b + c + e$)
- t_{23} – unload *M*₂, move to *M*₃ and load ($b + c + e$)
- t_{34} – unload *M*₃, move to *Out* and drop ($b + d + e$)
- t_{40} – move from *Out* to *In* (e)

Furthermore, it is assumed that there is always an available part in *In* and that *Out* removes manufactured parts very quickly, so *In* and *Out* are not shown although they can easily be added to the model.

It can be observed that the net in Fig.3 is conflict-free because all places have exactly one incoming and one outgoing arc. All invariant subnets of such a conflict-free net are also conflict-free nets.

A model for the robot’s strategy of Fig.2(b) is shown in Fig.4, where t_1, t_2 and t_3 represent the machine operations, as in Fig.3, and the remaining transitions correspond to robot actions:

- t_{01} – pick a part from *In*, move to *M*₁ and load ($a + c + e$)
- t_{12} – unload *M*₁, move to *M*₂ and load ($b + c + e$)
- t_{23} – unload *M*₂, move to *M*₃ and load ($b + c + e$)
- t_{30} – move from *M*₃ to *In* ($2e$)
- t_{32} – move from *M*₂ to *M*₃ (e)
- t_{34} – unload *M*₃, move to *Out* and drop ($b + d + e$)
- t_{42} – move from *Out* to *M*₂ ($2e$)

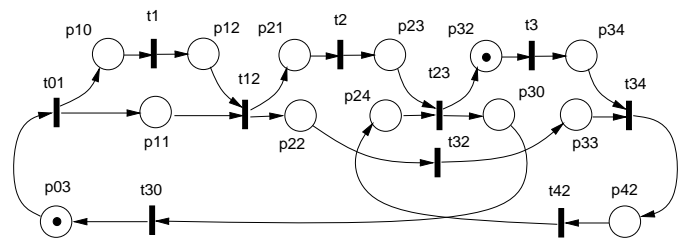


Fig.4. Petri net model of schedule B (Fig.2(b)).

Again, the net shown in Fig.4 is conflict-free. Similar models can easily be derived for the other robot’s strategies.

3. COMPOSITE SCHEDULES

Schedules in which several parts enter/leave the cell are called ‘composite schedules’. Composite schedules are obtained by merging simple schedules in a “consistent way”, i.e., in a way which preserves the liveness property of the model.

Fig.5 shows a timed Petri net model of a composite schedule obtained by merging schedules A and B from Fig.2; the central part represents the operations of machines M_1 , M_2 and M_3 , the lower part models the schedule B from Fig.2(b) (and Fig.4), and the upper part represents the schedule A from Fig.2(a). The “transitions” from schedule A to B and B to A can be performed only in those “states” of the schedules for which the distribution of parts in all machines is identical (for Fig.5, this distribution is: one part in M_2 and one part in M_3).

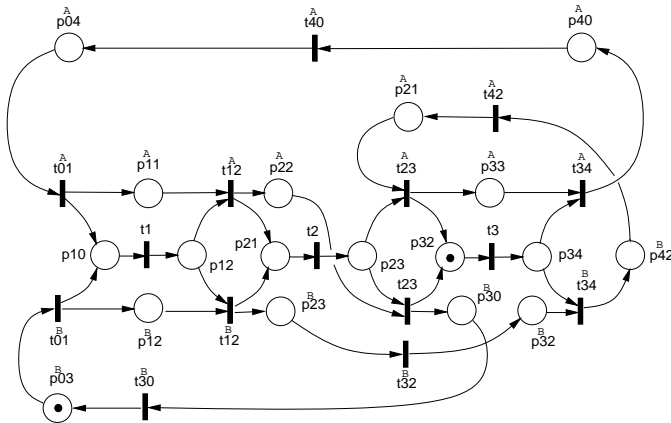


Fig.5. Petri net model of a composite schedule A+B.

It can be observed that the net from Fig.5 contains several shared places (p_{12} , p_{23} and p_{34}) which are created by combining simple schedules.

The numbers of different possible compositions for all pairs of simple schedules are as follows:

	A	B	C	D	E	F
A	*	2	3	1	2	0
B	2	*	3	3	2	2
C	3	3	*	2	2	1
D	1	3	2	*	3	3
E	2	2	2	3	*	2
F	0	2	1	3	2	*

so, the simple schedule A can be combined with B in two different ways, but it cannot be combined with the schedule F.

It should be observed that more complex composite schedules, for example B+C+F or B+C+B+E, etc., can be derived in the same way, by consistent merging of simple schedules.

4. MODEL EVALUATION

Analysis of net models can be based on their behavior (i.e., the space of reachable states) or on the structure of the net; the first case is called reachability analysis and the second, structural analysis. Invariant analysis seems to be the most popular example of the structural approach. Structural methods eliminate the derivation of the state space, so they avoid the ‘state explosion’ problem of reachability analysis, but they cannot provide as much information as the reachability approach does. Quite often, however, all the detailed results of reachability analysis are not really needed,

and more synthetic performance measures, that can be provided by structural methods, are quite satisfactory [Hi89].

Invariant analysis [Re85,MS82,KJ87] decomposes the net model into a number of simpler subnets and derives the properties of the model from properties of its components. It appears that timed Petri net models of manufacturing cells can easily be decomposed into simple nets which determine the performance of the whole model.

For example, the net shown in Fig.3 has eight basic invariant subnets, corresponding to the following subsets of transitions:

	t_1	t_2	t_3	t_{01}	t_{12}	t_{23}	t_{34}	t_{40}
1	0	0	0	1	1	1	1	1
2	0	0	1	1	1	1	1	1
3	0	1	0	1	1	1	1	1
4	0	1	1	1	1	1	1	1
5	1	0	0	1	1	1	1	1
6	1	0	1	1	1	1	1	1
7	1	1	0	1	1	1	1	1
8	1	1	1	1	1	1	1	1

Since all invariant subnets are elementary nets (all transitions and all places have a single incoming and outgoing arc), the cycle time of the net is equal to the largest cycle time of invariant subnets. Since one of the subnets includes all transitions (subnet 8), its cycle time determines the cycle time of the net, so the cycle time is equal to the sum of times assigned to the transitions:

$$T_c^A = o_1 + o_2 + o_3 + a + 3b + 3c + d + 5e$$

The net in Fig.4 has five basic invariants, but none of the invariants contains all transitions:

P-inv	t_1	t_2	t_3	t_{01}	t_{12}	t_{23}	t_{30}	t_{32}	t_{34}	t_{42}
1	0	0	0	1	1	1	1	1	1	1
2	0	0	1	0	0	1	0	0	1	1
3	0	1	0	1	1	1	1	0	0	0
4	1	0	0	1	1	1	1	1	1	1
5	1	1	0	1	1	1	1	0	0	0

It can be observed that the invariant (1) is a subset of (4), and (3) is a subset of (5). Consequently, the cycle time in this case is equal to:

$$T_c^B = \max(T_2, T_4, T_5),$$

where T_i is the cycle time of the invariant subnet i :

$$\begin{aligned} T_2 &= o_3 + 2b + c + d + 4e \\ T_4 &= o_1 + a + 3b + 3c + d + 9e \\ T_5 &= o_1 + o_2 + a + 2b + 3c + 5e \end{aligned}$$

For composite schedules the cycle times of schedules can be determined in a similar way. The net shown in Fig.5 has five invariants:

P-inv	subsets of transitions
1	$t_{01}^A, t_{12}^A, t_{23}^A, t_{34}^A, t_{40}^A, t_{01}^B, t_{12}^B, t_{23}^B, t_{32}^B, t_{34}^B, t_{30}^B, t_{42}^B$
2	$t_3, t_{01}^A, t_{12}^A, t_{34}^A, t_{40}^A, t_{23}^B, t_{34}^B, t_{42}^B$
3	$t_2, t_{01}^A, t_{12}^A, t_{23}^A, t_{34}^A, t_{40}^A, t_{01}^B, t_{12}^B, t_{23}^B, t_{30}^B$
4	$t_1, t_{01}^A, t_{12}^A, t_{23}^A, t_{34}^A, t_{40}^A, t_{01}^B, t_{12}^B, t_{23}^B, t_{32}^B, t_{34}^B, t_{30}^B, t_{42}^B$
5	$t_1, t_2, t_{01}^A, t_{12}^A, t_{23}^A, t_{34}^A, t_{40}^A, t_{01}^B, t_{12}^B, t_{23}^B, t_{30}^B$

Four of these invariant subnets are free-choice nets, as, for example, the one shown in Fig.6 (it corresponds to the invariant 5).

To find the cycle time of the net, the relative frequencies of non-deterministic “free-choices” must be determined from the (deterministic) behavior of the model. These relative frequencies of transition firings can be obtained from the net’s T-invariant [Re85]; since the net is deterministic, there is only one T-invariant. For the net shown in Fig.5, this T-invariant assigns the multiplicity 2 to t_1 , t_2 and t_3 , and multiplicity 1 to all other transitions (this T-invariant describes the composite cycle in which t_1 , t_2 and t_3 are fired twice and all other transitions only once). The relative frequencies are used as multipliers to determine the cycle times of invariant subnets and the cycle time of a net.

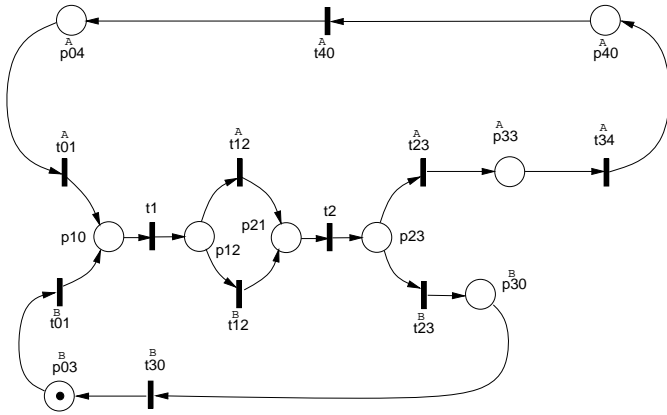


Fig.6. Free-choice invariant subnet of schedule A+B.

Since the invariant (1) is a subset of (4), and (3) a subset of (5), the cycle time is:

$$T_c^{A+B} = \max(T_2, T_4, T_5)$$

where T_2 , T_4 and T_5 are determined by the corresponding invariants (the ‘weights’ of transitions t_1 , t_2 and t_3 , determined by the T-invariant, are equal to two and for all other transitions they are equal to 1):

$$\begin{aligned} T_2 &= 2o_3 + a + 4b + 3c + 2d + 8e \\ T_4 &= 2o_1 + 2a + 6b + 2c + 2d + 14e \\ T_5 &= 2o_1 + 2o_2 + 2a + 5b + 6c + d + 10e \end{aligned}$$

Other composite schedules can be modeled and analyzed similarly.

5. CONCLUDING REMARKS

It has been shown that timed Petri nets can conveniently model (at least a class of) flexible manufacturing cells. For simple schedules, the modeling nets are composed of conflict-free nets, so the cycle time (or throughput) can easily be determined from cycle times (or throughputs) of the invariant subnets. In fact, a parameterized analytical solution can easily be derived, with times of basic operations (like “load”, “unload”, “move”) as parameters. For models of composite schedules, the invariant subnets are free-choice nets, so relative frequencies of firing conflicting transitions are needed to find cycle times and throughputs. These relative frequencies can be obtained from T-invariants of the modeling net.

The evaluation of the net models can be used for the determination of optimal schedules. For example, for the manufacturing cell of Fig.1, there are six different simple schedules (models of two of

them are shown in Fig.3 and Fig.4); the optimal schedule in this case is the schedule with minimal cycle time:

$$T_{opt} = \min(T_c^A, T_c^B, \dots, T_c^F)$$

The optimization procedure (based on systematic analysis of feasible schedules) can be automated; the number of composite schedules can grow very quickly with the number of simple components, however, this large number can be significantly reduced if specific values of parameters are taken into account.

A number of simplifying assumptions were used in the discussion presented in previous sections, e.g., all parts are identical, robot ‘travel times’ are equal, etc. It should be noted, however, that all these assumptions were made to simplify the discussion and they can easily be removed by straightforward modification of the presented approach.

Acknowledgement

The Natural Sciences and Engineering Research Council of Canada partially supported this research through Grants A8222 and OGPIN020.

References

[Cl83] Claybourne, B.H.: “Scheduling robots in flexible manufacturing cells”; CME Automation, vol.30, no.5, pp.36–40, 1983.

[DH90] Dixon, C., Hill, S.D.: “Work-cell cycle-time analysis in a flexible manufacturing system”; Proc. Pacific Conf. on Manufacturing, Sydney–Melbourne, Australia, vol.1, pp.182–189, 1990.

[Ha72] Hack, M.: “Analysis of production schemata by Petri nets”; Project MAC Technical Report TR–94, 1972.

[Hi89] Hillion, H.P.: “Timed Petri nets and application to multi-stage production system”; in: Advances in Petri Nets 1989 (Lecture Notes in Computer Science 424); pp. 281–305, Springer Verlag 1989.

[Ki90] King, P.J.B.: “Computer and communication systems performance modeling”; Prentice–Hall 1990.

[KJ87] Krueckeberg, F., Jaxy, M.: “Mathematical methods for calculating invariants in Petri nets”; in: “Advances in Petri Nets 1987” (Lecture Notes in Computer Science 266), G. Rozenberg (ed.), pp.104–131, Springer Verlag 1987.

[MS82] Martinez, J., Silva, M.: “Simple and fast algorithm to obtain all invariants of a generalized Petri net”; in: “Applications and Theory of Petri Nets” (Informatik Fachberichte 52); pp.301–310, Springer Verlag 1982.

[Mu89] Murata, T.: “Petri nets: properties, analysis and applications”; Proceedings of IEEE, vol.77, no.4, pp.541–580, 1989.

[Re85] Reisig, W.: “Petri nets - an introduction” (EATCS Monographs on Theoretical Computer Science 4); Springer Verlag 1985.

[SSSBK92] Sethi, S.P., Sriskandarajah, C., Sorger, G., Blazewicz, J., Kubiak, W.: “Sequencing of parts and robot moves in a robotic cell”; Int. Journal of Flexible Manufacturing Systems, vol.4, pp.331–358, 1992.

[SV89] Silva, M., Valette, R.: “Petri nets and flexible manufacturing”; in: “Advances in Petri nets 1989” (Lecture Notes in Computer Science 424), pp. 374–417, Springer Verlag 1989.

[Su85] Suri, R.: “An overview of evaluative models for flexible manufacturing systems”; Annals of Operations Research, vol.3, no.1, pp.3–21, 1985.

[Zu91] Zuberek, W.M.: “Timed Petri nets – definitions, properties and applications”; Microelectronics and Reliability (Special Issue on Petri Nets and Related Graph Models), vol.31, no.4, pp.627–644, 1991.