

Timed Petri Nets in Modeling and Analysis of Manufacturing Systems

W.M. ZUBEREK

Department of Computer Science
Memorial University
St. John's, CANADA A1B 3X5
wlodek@cs.mun.ca

Abstract: Petri nets are formal models of systems which exhibit concurrent activities. As formal models, Petri nets are bipartite directed graphs, in which the two types of vertices represent, in a very general sense, conditions and events. In order to study performance aspects of systems modeled by Petri nets, the durations of modeled activities must also be taken into account. This can be done in different ways, resulting in different types of nets “with time”. In timed Petri nets, occurrence times are associated with events. For timed nets with constant occurrence times, if the net is covered by a family of simple cyclic subnets, basic performance characteristics can be obtained by structural analysis, without the exhaustive generation of the state space. The paper briefly recalls basic concepts of Petri nets and timed Petri nets, and then derives hierarchical Petri net models of manufacturing systems.

Key-Words: timed Petri nets, performance analysis, manufacturing systems, hierarchical modeling.

1 Introduction

Petri nets have been proposed (by Carl Adam Petri [24]) as a simple and convenient formalism for modeling systems that exhibit concurrent activities [22], [23], [26]. Communication networks, multiprocessor systems, manufacturing systems and distributed databases are simple examples of such systems. The popularity that Petri nets (and their numerous extensions and modifications) have been gaining is due to simple representation of concurrency and synchronization, i.e., those aspect of systems which cannot be expressed easily in traditional formalisms, developed for analysis of systems with sequential behavior.

Petri nets are bipartite directed graphs, in which the two types of vertices, called places and transitions, represent, in a very general sense, conditions and events.

An event can occur only when all conditions associated with it (and represented by arcs directed to the event) are satisfied. An occurrence of an event usually satisfies some other conditions, indicated by arcs directed from the event. In effect, an occurrence of one event causes some other event(s) to occur, and so on.

In order to study performance aspects of Petri net models, the durations of activities must also be taken into account. Several types of Petri nets “with time” have been proposed by assigning “occurrence times” (or “firing times”) to the transitions or places of a net. In *timed* nets [32], [35], firing times are associated with transitions, and 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). All firings of enabled transitions are initiated in the same instants of time in which the transitions become enabled. In *stochastic* (and *generalized stochastic*) Petri nets [3], [5], [21] and their many variants [2], [7], [8], (exponentially distributed) firing times are associated with transitions, but the tokens remain (for the occurrence time) in places, and the instantaneous “firings” occur at the end of occurrence times (so the “occurrence times” are actually “enabling times”). In *time* nets [1], [20] there is an interval associated with a transition, and the (instantaneous) occurrence must occur within this interval of time.

Analysis of net models can be based on their behavior (i.e., the set of reachable states) or on the structure of the net; the former is called *reachability analysis* and the latter – *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, obtained by structural methods, are quite satisfactory.

Timed Petri nets are discrete–event models which can be continuous–time (M–timed nets) or discrete–time (D–timed nets). Analysis of timed models by event–driven simulation of their behavior is yet another approach to performance analysis, which imposes very few restrictions on the class of analyzed models [38] (for example, both continuous–time and discrete–time elements can be used in the same model).

Manufacturing systems are also discrete–event systems in which events correspond to the initiation or termination of different activities (such as an operation of a machine, moving a part from one location to another, etc.). It cannot, therefore, be surprising that Petri nets have often been used to modeling of manufacturing systems [9], [10], [25], [34].

This paper first recalls the most important concepts of Petri nets and timed Petri nets in Section 2. Section 3 introduces models of manufacturing systems at the system level and uses structural analysis of derived models for their performance analysis. Section 4 refines previous models at the machine level. The proposed hierarchical approach is summarized in Section 5. Section 6 concludes the paper.

2 Timed Petri Nets

Petri nets are formal models of systems that exhibit parallel and concurrent activities [22], [26]. In Petri nets, these activities are represented by the so called *tokens* which can move within a (static) graph–like structure of the net. More formally, a marked place/transition Petri net \mathcal{M} is defined as $\mathcal{M} = (\mathcal{N}, m_0)$, where the structure \mathcal{N} is a bipartite directed graph, $\mathcal{N} = (P, T, A)$, with a set of places P , a set of transitions T , a set of directed arcs A connecting places with transitions and transitions with places, $A \subseteq T \times P \cup P \times T$, and the initial marking function m_0 which assigns nonnegative numbers of tokens to places of the net, $m_0 : P \rightarrow \{0, 1, \dots\}$. Marked nets can be equivalently defined as $\mathcal{M} = (P, T, A, m_0)$.

A place is shared if it is connected to more than one transition. A shared place p is free–choice if the sets of places connected by directed arcs to all transitions sharing p are identical. A net is free–choice if all its shared places are free–choice. A net is structurally (or statically) conflict–free if it does not contain shared places. A marked net is dynamically conflict–free if for any marking reachable from the initial marking, and for any shared place, at most one of transitions sharing this place is enabled. The models of manufacturing systems discussed in this paper are (statically and dynamically) conflict–free nets.

In timed nets [35], occurrence times are associated with transitions, and transition occurrences are real–time events, i.e., tokens are removed from input places at the beginning of the occurrence period, and they are deposited to the output places at the end of this period. All occurrences of enabled transitions are initiated in the same instants of time in which the transitions become enabled (although some enabled transitions cannot initiate their occurrences). If, during the occurrence period of a transition, the transition becomes enabled again, a new, independent occurrence can be initiated, which will overlap with the other occurrence(s). There is no limit on the number of simultaneous occurrences of the same transition (sometimes this is called infinite occurrence semantics). Similarly, if a transition is enabled “several times” (i.e., it remains enabled after initiating an occurrence), it may start several independent occurrences in the same time instant.

More formally, a conflict–free timed Petri net is a pair, $\mathcal{T} = (\mathcal{M}, f)$, where \mathcal{M} is a marked net and f is a timing function which assigns an (average) occurrence time to each transition of the net, $f : T \rightarrow \mathbf{R}^+$, where \mathbf{R}^+ is the set of nonnegative real numbers.

The occurrence times of transitions can be either deterministic or stochastic (i.e., described by some probability distribution function); in the first case, the corresponding timed nets are referred to as D–timed nets, in the second, for the (negative) exponential distribution of firing times, the nets are called M–timed nets (Markovian nets). In both cases, the concepts of state and state transitions have been formally defined and used in the derivation of different performance characteristics of the model [35]. Only D–timed Petri nets are used in this paper.

Each place/transition net $\mathcal{N} = (P, T, A)$ can be conveniently represented by a connectivity (or incidence) matrix $\mathbf{C} : P \times T \rightarrow \{-1, 0, 1\}$ in which places correspond to rows, transitions to columns, and the entries are defined as:

$$\forall p \in P \forall t \in T : \mathbf{C}[p, t] = \begin{cases} -1, & \text{if } (p, t) \in A \wedge (t, p) \notin A, \\ +1, & \text{if } (t, p) \in A \wedge (p, t) \notin A, \\ 0, & \text{otherwise.} \end{cases}$$

Connectivity matrices disregard *selfloops*, that is, pairs of arcs (p, t) and (t, p) . A pure net is defined as a net without selfloops [26].

A P–invariant (place invariant, sometimes also called S–invariant) of a net \mathcal{N} is any nonnegative, nonzero integer (column) vector I which is a solution of the matrix equation

$$\mathbf{C}^T \times I = 0,$$

where \mathbf{C}^T denotes the transpose of matrix \mathbf{C} . It follows immediately from this definition that if I_1 and I_2 are P-invariants of \mathcal{N} , then any linear (positive) combination of I_1 and I_2 is also a P-invariant of \mathcal{N} . A basic P-invariant of a net is defined as a P-invariant which does not contain simpler invariants.

Moreover, a net $\mathcal{N}_i = (P_i, T_i, A_i)$ is a P_i -implied subnet of a net $\mathcal{N} = (P, T, A)$, $P_i \subset P$, if:

- (1) $A_i = A \cap (P_i \times T \cup T \times P_i)$;
- (2) $T_i = \{t \in T \mid \exists p \in P_i : (p, t) \in A_i \vee (t, p) \in A_i\}$.

A subnet $\mathcal{N}_I = (P_I, T_I, A_I)$ implied by a P-invariant I is obtained from a net $\mathcal{N} = (P, T, A)$ by selecting all those places, for which the corresponding elements of I are nonzero, $P_I = \{p \in P \mid I(p) > 0\}$ (P_I is sometimes called the support of the invariant I), and taking all arcs incident with these places in net \mathcal{N} (part (1) of the definition), and then including all transitions which are associated with the included arcs (part (2) of the definition). It should be observed that all arcs incident with places in P_I belong to A_I but some arcs incident with transitions in T_I are usually left out.

There are efficient algorithms for finding all basic invariants of a net [17], [19].

Net invariants can be very useful in performance evaluation of net models. If a net is covered by a family of conflict-free cyclic subnets (i.e., if each place and each transition of a net belongs to one of subnets), the cycle time of the net, τ_0 , is equal to the maximum cycle time of the covering subnets [30]:

$$\tau_0 = \max(\tau_1, \tau_2, \dots, \tau_k)$$

where k is the number of subnets covering the original net, and each τ_i , $i = 1, \dots, k$, is the cycle time of the subnet i , equal to the sum of occurrence times associated with the transitions divided by the total number of tokens assigned to the subnet:

$$\tau_i = \frac{\sum_{t \in T_i} f(t)}{\sum_{p \in P_i} m(p)}.$$

In many cases, the number of basic P-invariants can be reduced by removing from the analyzed net all these elements which do not affect the performance of models [42].

3 Manufacturing Systems

Modeling and analysis of manufacturing systems is one of the most popular applications of Petri nets [9], [10], [25], [34], however, the use of Petri net models for performance analysis of such systems has received little attention in the past; deadlock detection and deadlock prevention seem to be the dominant aspects of application of Petri nets to manufacturing systems.

A large class of manufacturing systems can be represented as collections of manufacturing cells connected by transportation (and possibly storage) systems. Simple examples of such systems are shown in Fig.4.1, where “A” is a supply of unprocessed parts, and “D” is a storage for the final products.

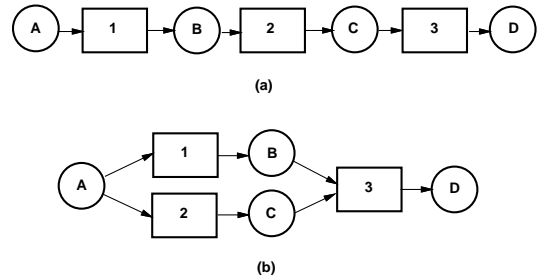


Fig.4.1. Outlines of simple manufacturing systems.

A Petri net model of the system from Fig.4.1(a) is shown in Fig.4.2. The three cells are represented by identical subnets, each of which contains one timed transition, t_i , which models the total operations performed by the corresponding cell (and which can be refined to show more details if needed); the two immediate transitions simply represent the operations of bringing a new part to a cell and removing a completed part from a cell. Places p_i , if marked, indicate that the corresponding cell is idle, waiting for another part.

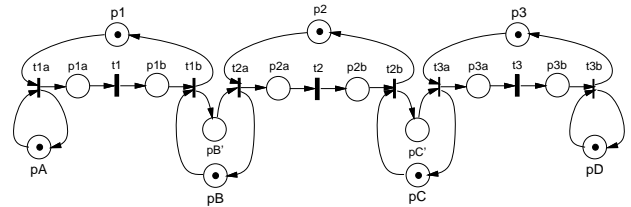


Fig.4.2. Petri net model of system from Fig.4.1(a).

Place p_A represents the source of unprocessed parts, and it is assumed that there is always sufficient supply of these parts; this is the reason that, in Fig.4.2, whenever a part is taken from “A” (transition t_{1a}), a part is also “returned” to p_A , so there is another part “ready”, when needed. Similarly, it is assumed that storage “D” can always accept another completed product. If these assumptions are not realistic, the model needs to be

expanded to take these additional constraints into account.

The connections between the cells are represented by buffers “B” and “C” with capacity 1; this capacity is indicated by the initial markings of places p_B and p_C . If a different capacity of these buffers is needed, the initial marking of these two places needs to be changed accordingly.

Fig.4.3 shows a Petri net model corresponding to the outline from Fig.4.1(b). This model differs in two aspects from that in Fig.4.2; the storage “A” is connected to both stages “1” and “2” in Fig.4.3, and the connection with stage “3” is different because of a different flow of parts in the system.

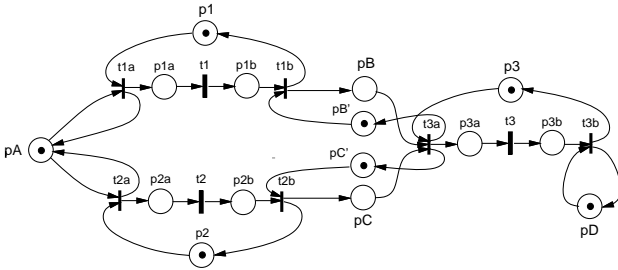


Fig.4.3. Petri net model of system from Fig.4.1(b).

The models shown in Fig.4.2 and Fig.4.3 are composed of simple cyclic subnets, so the structural approach can be used for their analysis.

The net shown in Fig.4.2 has 5 simple P-invariants, which imply subnets with the following subsets of transitions (these subnets correspond to the cyclic subnets which can easily be identified in Fig.4.2); three subnets corresponding to the three cells, and two subnets corresponding to the two buffers:

inv	t_{1a}	t_1	t_{1b}	t_{2a}	t_2	t_{2b}	t_{3a}	t_3	t_{3b}
1	1	1	1	0	0	0	0	0	0
2	0	0	1	1	0	0	0	0	0
3	0	0	0	1	1	1	0	0	0
4	0	0	0	0	0	1	1	0	0
5	0	0	0	0	0	0	1	1	1

The cycle time is thus:

$$\tau_0 = \max(\tau_1, \tau_2, \tau_3, \tau_4, \tau_5)$$

where $\tau_i, i = 1, \dots, 5$, are cycle times of the subnets (all subnets contain just one token):

$$\begin{aligned} \tau_1 &= f(t_{1a}) + f(t_1) + f(t_{1b}), \\ \tau_2 &= f(t_{1b}) + f(t_{2a}), \\ \tau_3 &= f(t_{2a}) + f(t_2) + f(t_{2b}), \\ \tau_4 &= f(t_{2b}) + f(t_{3a}), \\ \tau_5 &= f(t_{3a}) + f(t_3) + f(t_{3b}). \end{aligned}$$

The net shown in Fig.4.3 also has 5 simple P-invariants with the following sets of transitions of the subnets implied by these invariants:

inv	t_{1a}	t_1	t_{1b}	t_{2a}	t_2	t_{2b}	t_{3a}	t_3	t_{3b}
1	1	1	1	0	0	0	0	0	0
2	0	0	1	0	0	0	1	0	0
3	0	0	0	1	1	1	0	0	0
4	0	0	0	0	0	1	1	0	0
5	0	0	0	0	0	0	1	1	1

The difference with respect to the previous model (Fig.4.2) is only in subnet (2), for which the cycle time now is:

$$\tau_2 = f(t_{1b}) + f(t_{3a}).$$

The times of storing and retrieving parts $f(t_{1a})$, $f(t_{1b})$, etc., can be estimated on the basis of physical measurements; the times $f(t_1)$, $f(t_2)$ and $f(t_3)$ are usually derived from a more detailed analysis of the corresponding manufacturing cells.

4 Manufacturing Cells

Each manufacturing cell typically contains a number of versatile machines, $M1, \dots, M_k$, an input and output conveyor (or other transportation system), and a robot which moves the manufactured or assembled parts from one machine to another, and also from the input conveyor to the first machine and from the last machine to the output conveyor.

An outline of a simple manufacturing cell with 4 machines is shown in Fig.4.4.

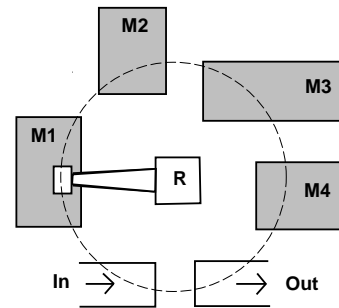


Fig.4.4. An outline of a 4-machine manufacturing cell.

A sequence of operations performed (cyclically) by the robot is called a *schedule*. It is known that there are $m!$ different schedules for a cell with m machines [29]. The best schedule is the one which maximizes the throughput (or minimizes the cycle time) of a cell. For a given cell, all schedules can be systematically derived, as Petri net models, and evaluated using P-invariants [36].

Fig.4.5 shows a Petri net model of the simplest, sequential schedule for a 4-machine cell.

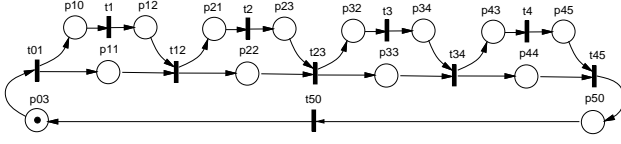


Fig.4.5. Net model of schedule “1234”.

The model is composed of four sections modeling the machines of the cell, each section composed of a transition t_i , $i = 1, 2, 3, 4$, and two places, one representing the condition that the part has been loaded, so the machine can begin its operation, and the other indicating that the machine’s operation has been completed, so the part can be moved by the robot to another machine or the output conveyor.

The sequence of robot’s operations is described by the following sequence of transitions:

t_i	robot’s operations
t_{01}	pick a part from <i>In</i> , move to M_1 and load
t_{12}	unload M_1 , move to M_2 and load
t_{23}	unload M_2 , move to M_3 and load
t_{34}	unload M_3 , move to M_4 and load
t_{45}	unload M_3 , move to <i>Out</i> and drop
t_{50}	move from <i>Out</i> to <i>In</i>

The model shown in Fig.4.5 contains several parallel paths which can be simplified without affecting the performance of the model [42]; all places p_{ii} , $i = 1, 2, 3, 4$, with their arcs can be removed, creating the simple cyclic model shown in Fig.4.6.

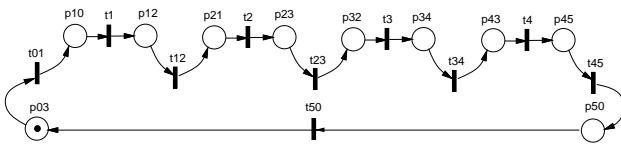


Fig.4.6. Simplified net model of schedule “1234”.

The cycle time of the model shown in Fig.4.6 is simply:

$$\tau_0^{(1)} = f(t_{01}) + f(t_1) + f(t_{12}) + f(t_2) + f(t_{23}) + f(t_3) + f(t_{34}) + f(t_4) + f(t_{45}) + f(t_{50}).$$

This cycle time can easily be expressed in terms of elementary operations (and their durations) performed by the robot. Assuming that:

u denotes the (average) part pickup time,
 v – the (average) machine unload time,
 w – the (average) machine load time,
 x – the (average) part drop time and
 y – the average ‘travel’ time between two adjacent machines (to simplify the description, it is assumed that this time is the same for all adjacent machines, and also the same for M_4 to *Out*, *Out* to *In* and *In* to M_1 moves),

the operations associated with transitions have the following (average) executions times:

t_i	$f(t_i)$
t_{01}	$u + w + y$
t_{12}	$v + w + y$
t_{23}	$v + w + y$
t_{34}	$v + w + y$
t_{45}	$v + x + y$
t_{50}	y

The cycle time, assuming that the (average) operation times of machines M_1 to M_4 are denoted by o_1 to o_4 , is:

$$\tau_0^{(1)} = o_1 + o_2 + o_3 + o_4 + u + 4v + 4w + x + 6y.$$

A different schedule, with two concurrent activities, is shown in Fig.4.7; the initial marking of place p_{32} indicates that, when the next part is being picked from the input conveyor, the previous part is loaded on machine M_3 and will be processed concurrently with the new part loaded on machine M_1 .

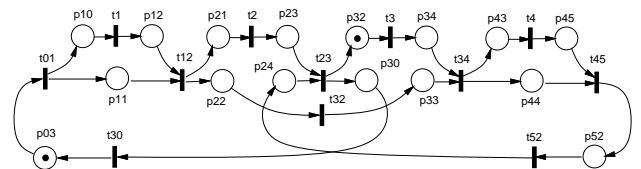


Fig.4.7. Net model of schedule “1243”.

In this case, the sequence of robot’s operations, with their execution times, is as follows:

t_i	$f(t_i)$
t_{01}	$u + w + y$
t_{12}	$v + w + y$
t_{32}	y
t_{34}	$v + w + y$
t_{45}	$v + x + y$
t_{52}	$3y$
t_{23}	$v + w + y$
t_{30}	$3y$

Similarly as before, the model can be simplified by removing places p_{11} and p_{44} as well as the arcs incident with them. The resulting net is shown in Fig.4.8.

are obtained in symbolic form, which provides very efficient analysis of specific configurations, described by sets of numerical parameters.

Hierarchical modeling by Petri nets is described in greater detail in [36], [37], [39], [40], while modeling and analysis of manufacturing cells is presented in [41].

6 Concluding Remarks

The presented approach uses timed Petri nets for hierarchical modeling of manufacturing systems. Structural analysis of derived models is used to estimate the throughput and the cycle time of the system. The results are obtained in a general symbolic form, so several variants of the analyzed system, characterized by different sets of parameters, can be evaluated without repeating the structural analysis.

An attractive aspect of the presented approach is that it can be used for a partially specified system in which only certain components are described in greater detail while the remaining ones are rather general. Such an approach would be typical in improvement studies, in which the “critical” part of the system is represented in detail in order to show the contributions of its elements, while the other parts, which are not critical to the performance of the whole systems, can be dealt with at a less detailed level.

Several simplifying assumptions were made in order to make the presentation as straightforward to follow as possible. Many of these assumptions can easily be removed by simple modifications of the model. For example, it was assumed that the input container always has sufficient supply of the parts so it does not affect the performance of the systems. If this may not be the case, a simple addition is needed, as shown in [39] in which an additional buffering structure is added at the input to the model.

Because of the complexity of real-life net models, high-level Petri nets are becoming increasingly popular in practical applications of Petri nets [28], [33]. Compositionality of models, usually expressed by process algebras, often with temporal enhancements for performance analysis, is expected to provide elegant formal methods for complex realistic applications [13], [16], [31].

Finally, there is an increasing number of monographs on Petri nets and their applications, so the popular Peterson’s book [23] and the Reisig’s monograph [26] are now supplemented by several books on application of Petri nets to manufacturing systems [9], [10], [25], [34], to workflow management systems [11], on modeling using stochastic Petri nets [4], [6], [12],

[18], and on colored Petri nets and their applications [14], [15], [27].

Acknowledgment

The Natural Sciences and Engineering Research Council of Canada partially supported this research through grant RGPIN-8222.

References

- [1] Aalst van der, W.M.P., “Interval timed colored Petri nets”; in: **Advances in Petri nets 1993** (Lecture Notes in Computer Science 674), pp.126–147, Springer-Verlag 1993.
- [2] Ajmone Marsan, M., Balbo, G., Bobbio, A., Chiola, G., Conte, G., Cumani, A., “The effect of execution policies on the semantics and analysis of stochastic Petri nets”; *IEEE Trans. on Software Engineering*, vol.15, no.7, pp.832–846, 1989.
- [3] Ajmone Marsan, M., Balbo, G., Conte, G., “The early days of GSPNs”; in: **Performance Evaluation: Origins and Directions** (Lecture Notes in Computer Science 1769), pp.505–512, Springer-Verlag 2000.
- [4] Ajmone Marsan, M., Balbo, G., Conte, G., Donatelli, S., Franceschinis, G., **Modeling with Generalized Stochastic Petri Nets**; J. Wiley & Sons 1995.
- [5] Ajmone Marsan, M., Conte, G., Balbo, G., “A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems”; *ACM Trans. on Computer Systems*, vol.2, no.2, pp.93–122, 1984.
- [6] Bause, F., Kritzinger, P.S., **Stochastic Petri Nets – and Introduction to the Theory**, Vieweg Verlag 1996.
- [7] Bobbio, A., Puliafito, A., Telek, M., Trivedi, K.S., “Recent developments in non-Markovian stochastic Petri nets”; *Journal of Circuits, Systems, and Computers*, vol.8, no.1, pp.119–158, 1998.
- [8] Chiola, G., Ajmone Marsan, M., Balbo, G., Conte, G., “Generalized stochastic Petri nets: a definition at the net level and its implications”; *IEEE Trans. on Software Engineering*, vol.19, no.2, pp.89–107, 1993.
- [9] Desrochers, A.A., Al-Jaar, R.Y., **Applications of Petri Nets in Manufacturing Systems**; IEEE Press 1995.
- [10] DiCesare, F., Harhalakis, G., Proth, J.M., Silva, M., Vernadat, F.B., **Practice of Petri Nets in Manufacturing**; Chapman and Hall 1993.
- [11] Girault, C., Valk, R., **Petri Nets for Systems Engineering**; Springer-Verlag 2002.
- [12] Haas, P.J., **Stochastic Petri Nets**; Springer-Verlag 2002.

- [13] Hillston, J., **A Compositional Approach to Performance Modeling**; Cambridge University Press 1996.
- [14] Jensen, K., **Colored Petri Nets – Basic Concepts, Analysis Methods and Practical Use**, vol.1; Springer-Verlag 1992.
- [15] Jensen, K., **Colored Petri Nets – Basic Concepts, Analysis Methods and Practical Use**, vol.2; Springer-Verlag 1995.
- [16] Koutny, M., “A compositional model of time Petri nets”; in: **Application and Theory of Petri Nets 2000** (Lecture Notes in Computer Science 1825), pp.303–322, Springer-Verlag 2000.
- [17] Krueckeberg, F., Jaxy, M., “Mathematical methods for calculating invariants in Petri nets”; in: **Advances in Petri Nets 1987** (Lecture Notes in Computer Science 266), pp.104–131, Springer-Verlag 1987.
- [18] Lindemann, C., **Performance Modeling with Deterministic and Stochastic Petri Nets**; Wiley and Sons 1998.
- [19] 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.
- [20] Merlin, P.M., Farber, D.J., “Recoverability of communication protocols – implications of a theoretical study”; *IEEE Trans. on Communications*, vol.24, no.9, pp.1036–1049, 1976.
- [21] Molloy, M.K., “Performance analysis using stochastic Petri nets”; *IEEE Trans. on Computers*, vol.31, no.9, pp.913–917, 1982.
- [22] Murata, T., “Petri nets: properties, analysis and applications”; *Proceedings of IEEE*, vol.77, no.4, pp.541–580, 1989.
- [23] Peterson, J.L., **Petri Net Theory and the Modeling of Systems**; Prentice-Hall 1981.
- [24] Petri, C.A., “Kommunikation mit Automaten”; Ph.D. Dissertation, University of Bonn, Bonn, Germany 1962; also: Memorandum MAC-M-212, Project MAC, Massachusetts Institute of Technology, Cambridge, MA.
- [25] Proth, J.M., Xie, X., **Petri Nets**; Wiley & Sons 1996.
- [26] Reisig, W., **Petri Nets - an Introduction** (EATCS Monographs on Theoretical Computer Science 4); Springer-Verlag 1985.
- [27] Reisig, W., **Elements of Distributed Algorithms – Modeling and Analysis with Petri Nets**; Springer-Verlag 1999.
- [28] Rokyta, P., Fengler, W., Hummel, T., “Electronic system design automation using high level Petri nets”; in: **Hardware Design and Petri Nets**, pp.193–204, Kluwer Academic Publ. 2000.
- [29] 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.
- [30] Sifakis, J., “Structural properties of Petri nets”; in: **Mathematical Foundations of Computer Science 1978** (Lecture Notes in Computer Science 64), pp.474–483, Springer-Verlag 1978.
- [31] Sifakis, J., “The compositional specification of timed systems – a tutorial”; in: **Computer Aided Verification** (Lecture Notes in Computer Science 1633), pp.2–7, Springer-Verlag 1999.
- [32] Wang, J., **Timed Petri nets**; Kluwer Academic Publ. 1998.
- [33] Wu, Z., “CEM/T net, a high level Petri net for FMS modeling”; *International Journal of Intelligent Control Systems*, vol.3, no.3, pp.377–387, 1999.
- [34] Zhou, M-C., **Petri Nets in Flexible and Agile Automation**; Kluwer Academic Publishers 1995.
- [35] 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.
- [36] Zuberek, W.M., “Hierarchical derivation of schedules for manufacturing cells”; *Proc. 9-th Symp. on Information Control in Manufacturing (INCOM-98)*, Nancy-Metz, France, pp.423–428, 1998.
- [37] Zuberek, W.M., “Stepwise refinements of net models and their place invariants”; *Proc. 8-th Int. Workshop on Petri Nets and Performance Models (PNPM’99)*, Zaragoza, Spain, pp.92–101, 1999.
- [38] Zuberek, W.M., “Discrete-event simulation of timed Petri net models”; *Proc. 33-rd Annual Simulation Symposium*; Washington, D.C., pp.91–98, 2000.
- [39] Zuberek, W.M., “Petri nets in hierarchical modeling of manufacturing systems”; *Proc. IFAC Conf. on Control System Design (CSD’2000)*, Bratislava, Slovakia, pp.287–292, 2000.
- [40] Zuberek, W.M., Bluemke, I., “Hierarchies of place/transition refinements in Petri nets”; *Proc. 5-th IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA’96)*, Kauai, Hawaii, pp.355–360, 1996.
- [41] Zuberek, W.M., Kubiak, W., “Timed Petri nets in modeling and analysis of simple schedules for manufacturing cells”; *Computers and Mathematics with Applications*, vol.37, no.12–12, pp.191–206, 1999.
- [42] Zuberek, W.M., Zuberek, M.S., “Transformations of timed Petri nets and performance analysis”; *Proc. 33-rd Midwest Symp. on Circuit and Systems* (Special Session on Petri Net Models), Calgary, Alberta, pp.774–778, 1990.