IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC'2000); Nashville, Tennessee, 8-11 Oct. 2000, pp.3021-3026.
Copyright © 2000 IEEE (DOI 10.1109/ICSMC.2000.884468).

Hierarchical Analysis of Manufacturing Systems Using Petri Nets

W.M. Zuberek
Department of Computer Science
Memorial University of Newfoundland
St.John's, Canada A1B 3X5

Abstract

Hierarchical analysis of manufacturing systems is performed in a top–down manner in which a general, approximate model is used to capture the main effects of components interconnections, while more detailed models of components provide the detailed information needed for the derivation of performance characteristics of the entire system. For Petri net models, this approach corresponds to stepwise refinements of models. Structural analysis, based on place invariants combined with simple net transformations, is used to obtain performance characteristics of the modeled systems.

1. Introduction

Modern manufacturing systems have to deal with requirements changing to small batch, larger product variety, production on demand with low lead times, with the ability to be "agile" [2]. This is in stark contrast to conventional manufacturing which has relied on economies of scale, and where changes were viewed as disruptions, detrimental to production. Computer–integrated manufacturing and flexible manufacturing practices are key components in the transition from conventional manufacturing to the "modern" manufacturing environments [5].

Modern manufacturing environments are, however, complex, expensive and difficult to operate. In order to eliminate design errors and oversights and take into account the effects of operational constraints, the design and actual implementation stages usually follow extensive experimentation and analysis studies of a model (or models) of the system.

A model is a formal, mathematical representation of the important features of the system. Conventional modeling tools, such as differential and difference equations, are not sufficient to deal with manufacturing systems which are event–driven, and can be asynchronous, with many concurrent activities interacting in complex ways. Petri nets [7, 9] are formal models developed specifically for modeling systems with concurrent, interacting activities. The structure of Petri nets is represented as a bipartite graph, i.e., a graph with two types of vertices connected by directed arcs; one type of vertices, called places, is used to represent conditions, while the other, called transitions, represents events. An event can occur only if all conditions associated with it are satisfied, but quite often several events can occur at the same time which is used for representation of concurrent activities. Petri nets are becoming quite popular models of manufacturing systems [3, 4, 8, 11].

In order to predict the performance of the modeled systems, the models must also represent the durations of activities. Several types of Petri nets "with time" have been proposed by assigning the timing information to different elements of Petri nets [1, 6, 12]. The models used in this paper are known as timed Petri nets [12], with the durations of activities assigned to transitions of the net.

Performance measures characterize the effectiveness of different configurations of manufacturing systems. The configuration of a manufacturing system is determined by the number and types of the machines, part routings, the number and type of material handling devices, capacities of locations and buffers and operating policies [5]. Performance measures determine the utilization of equipment, flow times, throughput rates, and other characteristics of a manufacturing system which are vital for evaluating the design and well as planning the operations.

Timed Petri nets can be analyzed by exhaustive analysis of the state spaces of such models (known as reachability analysis), or by using structural properties of models to predict their performance. If neither of these methods can be used, the models can be simulated to provide some basic characteristics of their behavior. The main deficiency of reachability analysis is that, in many models, the number of states grows exponentially with the size of the model (this is known as the "state explosion problem"), and this makes the approach impractical for large models. The fundamental deficiency of structural analysis is that it can be applied only to models with special properties. It appears, however, that many models of manufacturing systems satisfy these properties, and can be analyzed by structural methods.

Section 2 recalls basic concepts for timed nets and net invariants. Section 3 introduces models of manufacturing systems and their analysis, while Section 4 discusses the more detailed characterization of manufacturing cells, that complements the description of manufacturing systems presented in Section 3. A few concluding remarks are given in Section 5.

2. Timed Petri Nets

A place/transition net $\mathbf{N}$ is a triple $\mathbf{N} = (P, T, A)$ where P is a finite, nonempty set of places, T is a finite, nonempty set of transitions, and A is a set of directed arcs, $A \subseteq P \times T \cup T \times P$. It is usually assumed that for each transition t there is at least one place p such that $(p, t) \in A$.

A marked Petri net $\mathbf{M}$ is a pair $\mathbf{M} = (\mathbf{N}, m_0)$ where $\mathbf{N}$ is a Petri net, $\mathbf{N} = (P, T, A)$, and m_0 is an initial marking function, $m_0 : P \to \{0, 1, ...\}$ which assigns a (nonnegative) number of tokens to each place of the net. Let any function $m : P \to \{0, 1, ...\}$ be called a marking in a net $\mathbf{N} = (P, T, A)$.

A transition t is enabled by a marking m iff m assigns at least one token to each input place of t. Every transition enabled by a marking m can fire (or 'occur'). When a transition fires, a single token is removed from each of its input places and a single token is added to each of its output places. This determines a new marking in a net, a new set of enabled transitions, and so on. The set of all markings that can be derived from the initial marking is called the set of reachable markings. If this set if finite, the net is bounded.

A place p is shared iff it is an input place for more than one transition. A net is free–choice if the input sets of all transitions sharing the same place are identical. A net is structurally or statically conflict–free is it does not contain shared places. A marked net is dynamically conflict–free if for any marking in the set of reachable markings, and for any shared place, at most one of transitions sharing this place is enabled.

Each place/transition net $\mathbf{N} = (P, T, A)$ can conveniently be represented by a connectivity (or incidence) matrix $\mathbf{C} : P \times T \to \{-1, 0, +1\}$ in which places correspond to rows, transitions to columns, and the entries $\mathbf{C}[p, t]$, $p \in P$, $t \in T$, represent the directed arcs:

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

A P–invariant (place invariant) of a net $\mathbf{N}$ is any integer positive (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 $\mathbf{N}$, then also any linear (positive) combination of I_1 and I_2 is a P–invariant of $\mathbf{N}$.

A basic P–invariant of a net is defined as a P–invariant which does not contain simpler invariants. All basic P–invariants I are binary vectors [Re85], $I : P \to \{0, 1\}$.

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

$$(1)\ A_i = A \cap (P_i \times T \cup T \times P_i),$$

$$(2)\ T_i = \{t \in T \mid (p,t) \in A_i \vee (t,p) \in A_i\}.$$

It should be observed that in a pure net $\mathbf{N}$ (a net is pure if it does not contain self–loops $(p, t), (t, p)$), each P–invariant I determines a P_I-implied subnet of $\mathbf{N}$, where $P_I = \{p \in P \mid I(p) > 0\}$; P_I is sometimes called the support of the invariant I. All nonzero elements of I select rows of $\mathbf{C}$, and each selected row i corresponds to a place p_i with all its input (+1) and all output (–1) arcs associated with it.

Finding basic invariants is a 'classical' problem of linear algebra, and there are known algorithms to solve this problem efficiently.

In timed Petri nets [12] each transition takes a 'real time' to fire, i.e., there is a "firing time" associated with each transition of a net which determines the duration of transition's firings.

A conflict–free timed Petri net $\mathbf{T}$ is a pair $\mathbf{T} = (\mathbf{M}, f)$ where $\mathbf{M}$ is a conflict–free marked Petri net, $\mathbf{M} = (\mathbf{N}, m_0)$, $\mathbf{N} = (P, T, A)$, and f is a firing time function which assigns a nonnegative (average) firing time $f(t)$ to each transition t of the net, $f : T \to \mathbf{R}^{\oplus}$, and $\mathbf{R}^{\oplus}$ denotes the set of nonnegative real numbers.

If a conflict–free timed net is covered by its simple invariants, i.e., if each element of the net belongs to one of its simple P–invariants, the cycle time τ_0 of the net is determined by the maximum cycle time of subnets implied by the simple P–invariants:

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

where k is the number of simple P–invariants of the net, and each τ_i is the cycle time of a subnet implied by P–invariant i:

$$\tau_i \;=\; \frac{\sum_{t \in T_i} f(t)}{\sum_{p \in P_i} m_0(p)}.$$

3. Models of Manufacturing Systems

Manufacturing systems are often composed of machines grouped into manufacturing (or robotic) cells connected by transportation systems with some storage (or buffering) facilities. Fig.1 outlines two very simple manufacturing system with rectangles representing manufacturing cells, and circles – storage facilities. Case (a) in Fig.1 is a model of sequential processing, in which each manufactured part passes sequentially through the three stages “1”, “2” and “3” although usually all stages perform their operations concurrently, on different parts. In case (b), the first two stages perform their operations on two different parts which are then assembled into a final product in stage “3”.

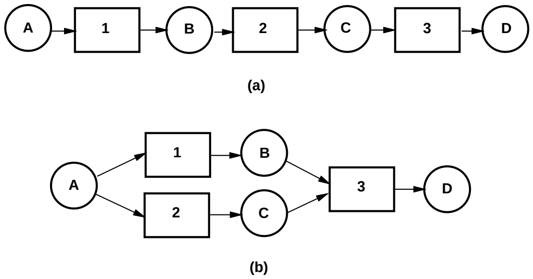

Fig.1. Examples of simple manufacturing systems.

A Petri net model of the outline form Fig.1(a) is shown in Fig.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 corresponding cell; the two immediate transitions simply represent the operations of bringing a new part to a cell and removing a complete part from a cell. Places p_i, if marked, indicate that the corresponding cell is idle, and waiting for another part.

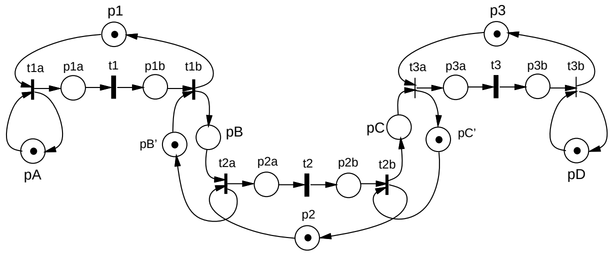

Fig.2. Petri net model of system outlined in Fig.1(a).

Place p_A represents the source of “raw” parts, and it is assumed that there is always sufficient suply of these parts. This is the reason that, in Fig.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 requirements into account.

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

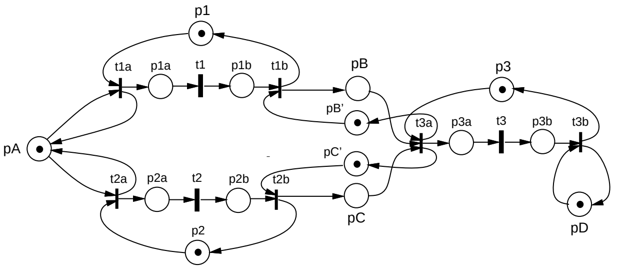

Fig.3. Petri net model of system outlined in Fig.1(b).

The net shown in Fig.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.2):

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 τ_i, $i = 1, .., 5$, are cycle times of the subnets:

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

The net shown in Fig.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.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 derived from more detailed analysis of corresponding manufacturing cells.

4. Manufacturing Cells

A manufacturing cell typically contains a number of versatile machines, M1, ...,Mk, an input and output conveyor, 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.

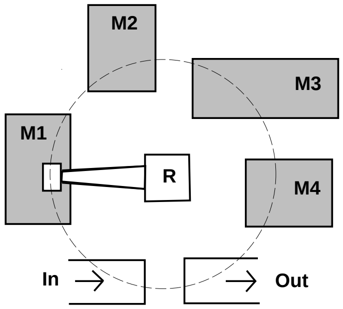

Fig.4. An outline of a 4–machine manufacturing cell.

A sequence of operations performed (cyclically) by the robot is called a schedule for a cell. It is known that there are $m!$ schedules for a cell with m machines [10]. The best schedule is the one which maximizes the throughput (or minimizes the cycle time) of a cell. It is also known [13] that the schedules can be systematically derived, represented as Petri net models, and evaluated by using P–invariants.

Fig.5 shows a Petri net model of the simplest, sequential schedule for a 4–machine cell. 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.

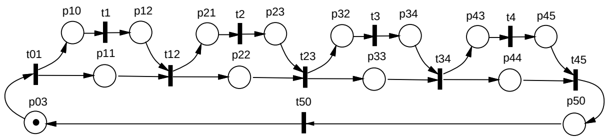

Fig.5. Net model of schedule 1234.

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 In, 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 Out and drop
t_{50}	move from Out to In

The model shown in Fig.5 contains several parallel paths which can be simplified without affecting the performance of the model [14]; all places p_{ii}, $i = 1, 2, 3, 4$, with their arcs can be removed, creating the simple cyclic model shown in Fig.6.

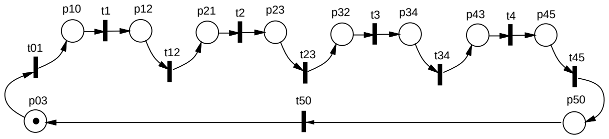

Fig.6. Simplified net model of schedule 1234.

The cycle time of the model shown in Fig.6 is:

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

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) pickup time,
v – the (average) unload time,
w – the (average) load time,
x – the (average) 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 $M4$ to Out, Out to In and In to $M1$ 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 M1 to M4 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.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 M3 and will be processed concurrently with the new part loaded on machine M1.

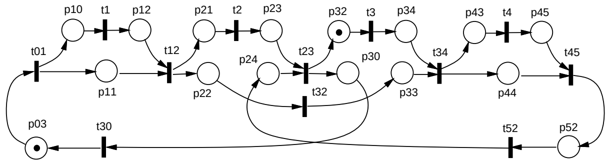

Fig.7. Net model of schedule 1243.

The sequence of robot's operations and their execution times, in this case, 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} with the arcs incident with them. The resulting net is shown in Fig.8.

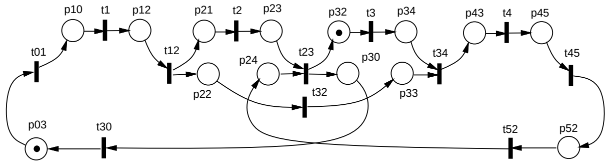

Fig.8. Simplified net model of schedule 1243.

The net in Fig.8 has 3 simple P–invariants with the following sets of transitions of subnets implied by these P–invariants:

inv.	*transitions*
1	$t_{01}, t_1, t_{12}, t_2, t_{23}, t_{30}$
2	$t_{23}, t_3, t_{34}, t_4, t_{45}, t_{52}$
3	$t_{01}, t_1, t_{12}, t_{32}, t_{34}, t_4, t_{45}, t_{52}, t_{23}, t_{30}$

so the cycle time is:

$$\tau_0^{(2)} \;=\; \max(\tau_1^{(2)}, \tau_2^{(2)}, \tau_3^{(2)})$$

where:

$$\begin{aligned}\tau_1^{(2)} &= o_1+o_2+u+2v+3w+6y,\\ \tau_2^{(2)} &= o_3+o_4+3v+2w+x+6y,\\ \tau_3^{(2)} &= o_1+o_4+u+3v+3w+x+12y.\end{aligned}$$

The schedule with the maximum concurrency is shown of Fig.9; in this case the three machines, M2, M3 and M4, are loaded with (previous) parts when a new part is picked and loaded on machine M1.

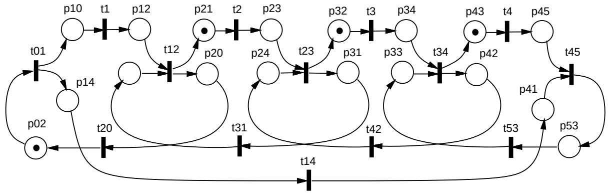

Fig.9. Net model of schedule 1432.

There is only one possible sequence of robot's operations for this model, and it is described by the following transitions (and their "execution times"):

t_i	$f(t_i)$
t_{01}	$u+w+y$
t_{14}	$3y$
t_{45}	$v+x+y$
t_{53}	$2y$
t_{34}	$v+w+y$
t_{42}	$2y$
t_{23}	$v+w+y$
t_{31}	$2y$
t_{12}	$v+w+y$
t_{20}	$2y$

The net shown in Fig.9 has 5 P–invariants which imply subnets with the following sets of transitions:

inv.	*transitions*
1	$t_{01}, t_1, t_{12}, t_{20}$
2	$t_{12}, t_2, t_{23}, t_{31}$
3	$t_{23}, t_3, t_{23}, t_{42}$
4	$t_{34}, t_4, t_{45}, t_{53}$
5	$t_{01}, t_{14}, t_{45}, t_{53}, t_{34}, t_{42}, t_{23}, t_{31}, t_{12}, t_{20}$

so the cycle time of this model is:

$$\tau_0^{(3)} \;=\; \max(\tau_1^{(3)}, \tau_2^{(3)}, \tau_3^{(3)}, \tau_4^{(3)}, \tau_5^{(3)})$$

where:

$$\begin{aligned}
\tau_1^{(3)} &= o_1 + u + v + 2w + 4y, \\
\tau_2^{(3)} &= o_2 + 2v + 2w + 4y, \\
\tau_3^{(3)} &= o_3 + 2v + 2w + 4y, \\
\tau_4^{(3)} &= o_4 + 2v + w + x + 4y, \\
\tau_5^{(3)} &= u + 4v + 4w + x + 16y.
\end{aligned}$$

The derived cycle times of manufacturing cells can be used in the model of manufacturing system (Section 3), replacing the operations times of the cells; in particular, if the model shown in Fig.5 is representing cell "1" in Fig.1, the model shown in Fig.7 – cell "2" in Fig.1, and model shown in Fig.9 – cell "3", the cycle time for the manufacturing system is equal:

$$\tau_0 \;=\; \max(\tau_0^{(1)}, \tau_0^{(2)}, \tau_0^{(3)})$$

where the terms $\tau_0^{(i)}$, $i = 1, 2, 3$, are defined above.

5. Concluding Remarks

A systematic approach to representation and analysis of a large class of concurrent systems (an in particular, manufacturing systems) is proposed. The approach is based on stepwise refinement of timed Petri net models, and structural analysis used for performance evaluation of derived models. The results are derived in symbolic form, which provides very efficient analysis of specific configurations, described by sets of numerical parameters.

The hierarchical approach presented in this paper first analyzes the performance of the manufacturing system at the abstract level of cells and storage elements between cells, and then considers the cells one at a time. For complex manufacturing systems, even more structured approach can be justified. Instead of dealing with all the cells at the same time, an additional level of subsystems can be introduced, and first the performance of the system can be expressed in terms of subsystems, then performance of subsystems in terms of cells, and finally, performance of cells in terms of individual machines and their interconnections.

References

[1] 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.

[2] Ayres, R.U., Butcher, D.C., "The flexible factory revisited"; *American Scientist*, vol.81, no.5, pp.448–459, 1993.

[3] Desrochers, A.A., Al-Jaar, R.Y., *Applications of Petri nets in manufacturing systems*; IEEE Press 1995.

[4] DiCesare, F., Harhalakis, G., Proth, J.M., Silva, M., Vernadat, F.B., *Practice of Petri nets in manufacturing*; Chapman & Hall 1993.

[5] Joshi, S.B., Smith, J.S., "Computer control of flexible manufacturing systems – research and development"; Chapman & Hall 1994.

[6] 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.

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

[8] Proth, J.M., Hillion, H.P., "Mathematical tools in production management" (Competitive Methods in Operations Research and Data Analysis); Plenum Press 1990.

[9] Reisig, W., *Petri nets - an introduction* (EATCS Monographs on Theoretical Computer Science 4); Springer–Verlag 1985.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] Zuberek, W.M., Zuberek, M.S.: "Transformations of timed Petri nets and performance analysis"; Proc. 33-rd Midwest Symp. on Circuit and Systems, Calgary, Alberta, pp.774–778, 1990.