MODELLING AND CONTROL OF RESOURCE ALLOCATION SYSTEMS WITHIN DISCRETE EVENT SYSTEMS BY MEANS OF PETRI NETS – PART 1: INVARIANTS, SIPHONS AND TRAPS IN DEADLOCK AVOIDANCE

František ČAPKOVIČ

Institute of Informatics
Slovak Academy of Sciences
Dúbravská cesta 9
845 07 Bratislava, Slovakia
e-mail: `Frantisek.Capkovic@savba.sk`

Abstract. Solving the deadlocks avoidance problem in Resource Allocation Systems (RAS) in Discrete-Event Systems (DES) is a rife problem, especially in Flexible Manufacturing Systems (FMS), alias Automated Manufacturing Systems (AMS). Petri Nets (PN) are an effectual tool often used at this procedure. In principle, there are two basic approaches how to deal with deadlocks in RAS based on PN. They are listed and illustrated here. First of the approaches is realized by means of the supervisor based on P-invariants of PN, while the second one is realized by means of the supervisor based on PN siphons. While the first approach needs to know the reachability graph/tree (RG/RT) expressing the causality of the development of the PN model of RAS, in order to find (after its thorough analysis) the deadlocks, the second approach needs the thorough analysis of the PN model structure by means of finding siphons and traps. Next, both approaches will be applied on the same PN model of RAS and the effectiveness of the achievement of their results will be compared and evaluated. Several simple illustrative examples will be introduced. For the in-depth analysis of the problem of deadlock avoiding, next Part 2 of this paper is prepared, where the newest research will be introduced and illustrated on more complicated examples. If necessary (because of the limited length of particular papers), also the third part – Part 3, will be prepared.

Keywords: Automated manufacturing systems, control synthesis, deadlocks, deadlock avoidance, discrete-event systems, flexible manufacturing systems, modelling,

1 INTRODUCTION

This paper has the character of an overview paper. It is conceived as the first part of a two-part paper (maybe also a three-part, if it will be needed). Its main aim is to introduce and describe principled terms and two basic approaches to the deadlock avoidance in Resource Allocation Systems (RAS) in Discrete-Event Systems (DES) as well as to present simple illustrative examples. The intended Part 2 of this paper (possibly also the Part 3, if necessary), to solve more complicated cases of RAS using newer methods, as well as comparing the mentioned two approaches, will be submitted later.

Flexible Manufacturing Systems (FMS), lately also called Automated Manufacturing Systems (AMS), represent a class of DES. They consists of various resources like machine tools, robots, buffers, transport belts, automatically guided vehicles (AGV) and so on. The resources are usually shared by two or more subsystems of AMS/FMS. Because of a limited number of resources different kinds of problems arise during the system operation, especially deadlocks [46]. Deadlocks are undesirable and unfavorable because they disrupt the course of the technological process. Due to deadlocks, either the entire plant or some of its parts remain stagnate. In such a way the primal intention of the production cannot be achieved. Such a situation can be, especially from the practical view, understood as a very unsafe form of non-determinism. Consequently, the approaches how to deal with this are sought. The approaches employing Petri Nets (PN)-based models of AMS/FMS are often used [52]. Besides the approaches based on the analysis of PN reachability trees/graphs (RT/RG), the approaches based on utilizing PN siphons are more frequently used. Moreover, in the recent years siphon-based approaches even started to prevail.

DES are systems discrete in nature. Such a system remains in a real intact state until it is forced to change this state as a consequence of the occurrence of a discrete event. In this document, PN-based models of DES will be exclusively used.

1.1 Petri Net Structure

As to the structure, PN are bipartite directed graphs $\langle P, T, F, G \rangle$ with two kinds of nodes – places p_i, $i = 1, \ldots, n$, and transitions t_j, $j = 1, \ldots, m$, and also two kinds of edges being directed arcs – $f_{ij} \in \mathbb{Z}_{\geq 0}$, $i = 1, \ldots, n$, $j = 1, \ldots, m$, from places to transitions (where $\mathbb{Z}_{\geq 0}$ is the set of non-negative integers), and $g_{ji} \in \mathbb{Z}_{\geq 0}$,

$j = 1, \ldots, m$, $i = 1, \ldots, n$, from transitions to places. In other words, f_{ij} and g_{ji} represent weights of the directed arcs (i.e. their multiplicity).

Thus, $P = \{p_1, \ldots, p_n\}$ is a set of places; $T = \{t_1, \ldots, t_m\}$ is a set of transition; $P \cap T = \varnothing$, $P \cup T \neq \varnothing$, with $\varnothing$ being the empty set; $F = \{f_{ij}\}_{i=1,n; j=1,m}$ is the set of the directed arcs from places to transitions (i.e. $p_i \to t_j$), $F \subseteq P \times T$; $G = \{g_{ji}\}_{j=1,m; i=1,n}$ is the set of the directed arcs from transitions to places (i.e. $t_j \to p_i$), $G \subseteq T \times P$; the set $B \subseteq (P \times T) \cup (T \times P)$, $B \in \mathbb{Z}$, were $\mathbb{Z}$ is the set of integers.

When nonzero elements of F, G are solely of value 1 the arcs are called ordinary, when some of nonzero elements of F, G or all of them are greater than 1 the arcs are called weighted. The sets F, G, B can be represented, respectively, by the incidence matrices $\mathbf{F} \in \mathbb{Z}_{\geq 0}^{(n \times m)}$, $\mathbf{G} \in \mathbb{Z}_{\geq 0}^{(m \times n)}$, $\mathbf{B} \in \mathbb{Z}^{(n \times m)}$ of the directed arcs. In general, $\mathbb{Z}_{\geq 0}^{(a \times b)}$ represents the $(a \times b)$ matrix of non-negative integers and $\mathbb{Z}^{(a \times b)}$ is the $(a \times b)$ matrix of integers.

PN defined in such a way are called place/transition PN(P/T PN). PN are called *pure* when they do not contain self-loops. PN are called *ordinary* when all weights of their arcs are equal to one.

In general, PN places can be of three kinds:

1. operation places representing a progress in AMS/FMS;

2. fixed resources representing shared devices or elements (e.g. working tools);

3. variable resources representing e.g. availability of semi-products, parts, etc.

A transition $t \in T$ is enabled in marking M, denoted by $M[t\rangle$, if and only if (verbally expresed by *iff* or symbolically by $\Leftrightarrow$) $\forall p \in P$: $M(p) \geq F(p, t)$. Consequently, when t is *enabled* in M, then t may yield (after its firing) another marking M' where $\forall p \in P : M'(p) = M(p) - F(p, t) + G(t, p)$. This is denoted as $M[t\rangle M'$. It means, that the *enabled* transition may be fired and partake of the PN marking evolution. When t does not meet the above introduced condition, i.e. when $\forall p \in P$: $M(p) < F(p, t)$, it is *disabled*. Such t cannot be fired, i.e., it cannot share in the marking development.

There exist specific transitions:

1. the *source* transition is a transition without any input place – it is unconditionally enabled;

2. the *sink* transition is a transition without any output place – it consumes but does not create any tokens.

1.2 Petri Net Dynamics

Besides the graph structure, PN have also dynamics (the PN marking evolution). In an effort to make an analogy with the classical control theory, consider the PN marking to be the state vector of the system being the PN model. Thus, the model

can be expressed by the following constrained linear discrete integer system:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{B}.\mathbf{u}_k, \quad k = 0, 1, \ldots \tag{1}$$

$$\mathbf{F}.\mathbf{u}_k \leq \mathbf{x}_k \tag{2}$$

where $\mathbf{x}_k = (\sigma_{p_1}^k, \ldots, \sigma_{p_n}^k)^T$ with $\sigma_{p_i}^k \in \mathbb{Z}_{\geq 0}$ is the state vector (marking) expressing the states of particular places (the number of tokens in p_i) in the step k; $\mathbf{u}_k = (\gamma_{t_1}^k, \ldots, \gamma_{t_m}^k)^T$ with $\gamma_{t_j}^k \in \{0, 1\}$ (where 0 means the disabled t_j, while 1 means the enabled t_j) is the control vector in the step k; $\mathbf{F}$ (frequently being named as **Pre**), $\mathbf{G}^T$ (frequently being named as **Post**) are, respectively, the incidence matrices corresponding to sets F, G; $\mathbf{B} = (\mathbf{G}^T - \mathbf{F})$ is the incidence matrix being the structural matrix of the system (1)–(2); $\mathbf{x}_0$ is the initial state vector (initial marking). Thus, the state vector $\mathbf{x}_k$ in (1) corresponds to $M(p)$ mentioned above.

In the following, we will use the symbol N for the PN introduced above, and the term marking of PN places as an alternative to the state of the places (i.e. the number of tokens placed in them). In other words, under PN we will understand $(N, \mathbf{x}_0)$. Under the symbol $\mathcal{R}$ we will mean the set of reachable states including the initial state $\mathbf{x}_0$. Sometimes $\mathcal{R}$ will be expressed by a matrix $\mathbf{X}_r \in \mathbb{Z}_{\geq 0}^{(n \times N_v)}$, whose columns are particular reachable state vectors with N_v being the number of the reachable state vectors (including the initial state $\mathbf{x}_0$). The columns of $\mathbf{X}_r$ represent the particular nodes of RT corresponding to the PN in question.

2 PRELIMINARIES

Let us introduce here the basic terms representing important terminology and properties of the PN models of AMS/FMS, which will be used in this paper.

2.1 Siphons, Traps, Deadlocks, Invariants, Repetitive and Characteristic Vectors

Definition of Siphons and Traps. There exist many papers where siphons and traps are defined – see e.g. [72, 84, 58, 60, 61, 92, 93, 54] and many newer ones mostly written by Chinese authors [63, 8, 25, 55, 56, 57, 59, 62, 64, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40]. Siphons and traps are basic net structures of PN [84] which allow important views on the behaviour of the modelled system as well as on some implications on this behaviour.

For ordinary PN the definitions of siphons and traps are defined in many papers – see e.g [72, 17, 78, 84, 1] and many other new ones mentioned above – as follows. A nonempty subset $S \subset P$ in N is called a siphon if $^\bullet S \subseteq S^\bullet$, i.e., if every transition having an output place in S has an input place in S. In [72] and several other older works the siphon is even identified with deadlock. However, at present it is an obsolete understanding. A nonempty subset $Q \subset P$ in an ordinary PN is called a trap if $Q^\bullet \subseteq {}^\bullet Q$ i.e., if every transition having an input place in Q has an output place in Q.

The illustration of the simple siphon and the simple trap is introduced in Figure 1.
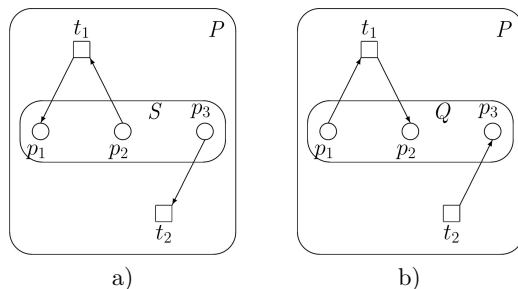

Figure 1. The a) siphon S and b) the trap Q inside a net N with the set of places P

As we can see, here $^\bullet S = \{t_1\}$, $S^\bullet = \{t_1, t_2\}$, $^\bullet S \subseteq S^\bullet$ while $Q^\bullet = \{t_1\}$, $^\bullet Q = \{t_1, t_2\}$, $Q^\bullet \subseteq {}^\bullet Q$. The number of tokens in the siphon S remains the same by firing t_1 and decreases by firing t_2. The number of tokens in the trap Q remains the same by firing t_1, but increases by firing t_2.

Consequences of Siphons and Traps.   In general,

1. the siphon behaviour is such that if it has no token in a state (marking) of N, then it remains without any token in each successor state. Siphons represent a very important structural concept of PN. When all places in a siphon have no token, all transitions connecting with the siphon cannot be firable any more. Siphons are widely used to analyze PN liveness, and also to prevent deadlocks in PN models of DES. The terms liveness and deadlocks are introduced in the next paragraph;

2. the trap behaviour is such that if it has at least one token in a state (marking) of N, then it remains marked under each successor state. It was proved in [72] that the union of two siphons (traps) is again a siphon (trap).

The evident resume is that siphons are sets of places which, if become empty of tokens, they will always remain empty for all reachable markings of the net, while traps are sets of places which, if become marked, will always remain marked for all reachable markings of the net.

Traps can also be useful in combination with place invariants (see Subsection 2.2) to recapture information lost in the incidence matrix due to the cancellation of self-loop arcs.

Deadlocks and Their Relation with Siphons and Traps.   In some literary sources – see e.g. [76] – is shown that each reachable marking of PN enables at least one transition. In doing so it means that each siphon S of PN contains as a subset

an initially marked trap. It was proved by the same author in [77] that a totally deadlocked ordinary Petri net contains at least one empty siphon.

If every non-empty siphon of PN includes a (sufficiently) marked trap then (see e.g. [84], but also many other authors) no dead marking is reachable. This is very important finding.

A siphon (trap) is named to be *minimal* if it does not contain any other siphon (trap). Minimal siphons provide a sufficient condition for the non-existence of deadlocks.

A *strict minimal* siphon [27] is a siphon containing neither other siphon nor a trap except itself. The sum of token numbers in S is denoted by $M(S)$, where $M(S) = \sum_{p \in S} M(p)$. A subset $S \subseteq P$ is marked by M if $M(S) > 0$. A siphon is under-marked if $\nexists t \in S^{\bullet}$ which can fire.

The *proper* siphon is the siphon when the set of its predecessors is strictly included in the set of its successors. It was shown in [11, 1] that in a deadlocked PN model all unmarked places form a siphon. Thus, the siphon-based approach for deadlocks detection checks if the net contains a proper siphon that can become unmarked by some firing sequence. A proper siphon does not become unmarked if it contains an initially marked trap.

Deadlocks and Liveness.   The problem of deadlocks and their effective resolution was studied for the first time in the 60's [15, 16], in context of the multi-threaded computation or multi-programming (emerging at that time). Of course, since that time the deadlock theory, and especially the deadlock avoidance one, has been intensively developed – see e.g. [46, 78, 79, 80, 81, 75, 74, 73, 82, 86, 87] and especially [63, 8, 25, 55, 56, 57, 59, 62, 64, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 92, 93, 94, 91, 88, 89, 65, 66, 67, 68, 69, 58, 60, 61, 47, 48, 49, 50, 51, 52, 53].

Simply said, the deadlock is the state of N if all transitions $t \in T$ are disabled. N is named to be deadlock-free when none reachable state $\mathbf{x} \in \mathcal{R}$ is a deadlock. Thus, the deadlock is a state of N when DES modelled by PN comes to a state in which no further changes are possible. In other words a deadlock is [70] a subset of places which, if none of them is marked at the beginning of the Petri net activity, will remain unmarked in all subsequent evolution. It can also be said that a subset of places $I \subset P$ is a deadlock *iff* (if and only if) each transition which is input transition of a place in I is also output transition of a place in I.

It was said in [26] that if M_0 is initial marking of N then in (N, M_0) is a deadlock only on the condition that $t \in T : M_0[t\rangle$ is never found.

In general, deadlocks occur in DES (especially AMS/FMS) when processes, which want to run (and should run), hold insufficient resources, as a result of which the system comes to a standstill. This is the acute problem which should be solved by means of PN based RAS.

A N is said to be live, more precisely M_0 is said to be a live marking for N, if (no matter what state (marking) has been reached from M_0) it is possible to ultimately fire any transition of N by progressing through further firing sequence. A live PN
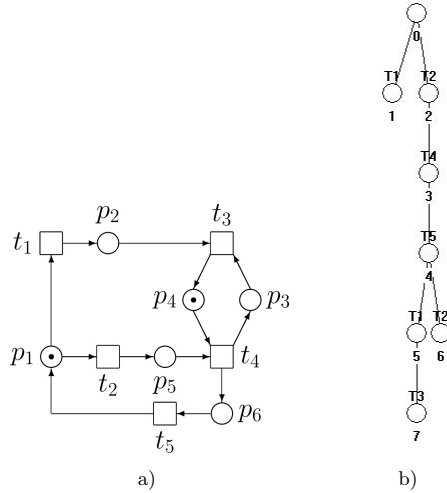
guarantees [72] deadlock-free operation, no matter what firing sequence is chosen. More details about five kinds of liveness can be found in [72]. Namely, a transition t in (N, M_0) is said to be:

1. *dead* (lived on the level $0 - L0$ live) – if it never be fired in any firing sequence in $L(M_0)$;

2. *potentially fired* ($L1$ live) – if it can be fired at least once in some firing sequence in $L(M_0)$;

3. $L2$ live – when for $k > 0$ it is fired at least k-times in $L(M_0)$;

4. $L3$ live – if it appears infinitely often in some firing sequence in $L(M_0)$;

5. $L4$ live or live if it is $L1$ live for every marking M in $\mathcal{R}(M_0)$.

The simple example of the live PN together with its reachability tree (RT) is given in Figure 2 while the simple example of the nonlive PN together with its reachability tree (RT) is given in Figure 3.

Figure 2. a) The live PN and b) its RT

As we can see in Figure 2 b), no state discontinues the course of the modelled process. On the other hand, in Figure 3 b) we can see that the states $\mathbf{x}_1 = (0\,1\,0\,1\,0\,0)^T$, $\mathbf{x}_6 = (0\,0\,1\,0\,1\,0)^T$ and $\mathbf{x}_7 = (0\,0\,0\,1\,0\,0)^T$ do this. These states are deadlocks.

Figure 3. a) The nonlive PN and b) its RT

Simply said, a transition t of N is said to be live *iff* for all reachable states (markings) $\mathbf{x}_r \in \mathcal{R}$ there exists a sequence of transition firings which results in a marking in which t is enabled. The N is said to be live if all its transitions are live. Liveness of PN implies absence of deadlocks in the modelled DES. When RT has a node (vertex) without a successor, then PN is not live. As to RT, the Koenig lemma [14] is also useful. It says: "Let RT be a tree of finite degree (i.e., every vertex has a finite number of successors) and with an infinite number of vertices. Then RT has an infinite branch."

In PN theory siphons and traps have been introduced [11] to characterize deadlocks of PN. Simultaneously, they can help us at finding deadlock avoidance methods.

Other details about behavioral properties of liveness are summarized in [18].

How to deal with the problem caused by the deadlock, that is a cardinal question. There exist three basic approaches for RAS (see e.g. [19, 74, 73]) how to deal with deadlocks and problems pertinent to them:

1. deadlock detection and recovery – i.e., to detect deadlock occurrences and restore the systems operations with recovery procedures;

2. deadlock prevention – i.e., to prevent circular wait conditions using offline strategies, mutual exclusion,

3. deadlock avoidance – i.e., to prevent deadlock situation applying online policy control of resource allocation.

In doing so there are two principles how to create related methods, namely by means of digraphs or by means of PN. Their comparison can be found in [20].

In this paper, solely the PN-based approaches will be used.

2.2 Invariants, Repetitive and Characteristic Vectors

Invariants. There are two kinds of PN invariants: T-invariants and P-invariants. It is well known – see e.g. [72] – that the T-invariant of PN is defined as the $(m \times 1)$ vector $\mathbf{w}$ for which $\mathbf{B}.\mathbf{w} = \mathbf{0}$, $\mathbf{w} \neq \mathbf{0}$, where $\mathbf{0}$ is the $(n \times 1)$ vector of zeros, while the P-invariant is defined as the $(n \times 1)$ vector $\mathbf{y}$ for which $\mathbf{B}^T.\mathbf{y} = \mathbf{0}$, $\mathbf{y} \neq \mathbf{0}$, where $\mathbf{0}$ is the $(m \times 1)$ vector of zeros. When we want to compute so called *proper* invariants we have to ask $\mathbf{w} > \mathbf{0}$ and $\mathbf{y} > \mathbf{0}$, respectively. What is important is that:

1. The T-invariant, if it exists at all, will give the number of times different transitions should be fired in order that a particular marking may be reproducible.

2. From the definition of P-invariants ($\mathbf{y}^T.\mathbf{B} = \mathbf{0}$, i.e. $\mathbf{y}^T.\mathbf{x}_k \overset{!}{=} \mathbf{y}^T.\mathbf{x}_0$) it follows that for all reachable markings $\mathbf{x}_k \in \mathcal{R}$, the weighted sum of tokens is a constant. Let $\mathbf{I}$ is a P-invariant. The set $P_I \subset P$ is called the *support* of $\mathbf{I}$ [45] *iff* $P_I = \{p \in P \mid \mathbf{I}(p) \neq \mathbf{0}\}$. $\mathbf{I}$ is called *non-negative iff* $\mathbf{I} \geq \mathbf{0}$. $\mathbf{I} \gneqq \mathbf{0}$ is called minimal *iff* there exists no P-invariant $\mathbf{I}' \gneqq \mathbf{0}$ with $\mathbf{I}' \lneqq \mathbf{I}$. Here, the symbol $\gneqq$ means *greater-than but not equal* and the symbol $\lneqq$ means *less-than but not equal*.

For the example of PN given in Figure 2 invariants (Figure 2 a)) and proper invariants (Figure 2 b)) are as follows:

$$\mathbf{I}_1 = (0, 0, 0, 1, 1, 1, 0, 0)^T,$$

$$^1\mathbf{I}_{prop} = (0, 0, 0, 1, 1, 1, 0, 0)^T,$$

$$\mathbf{I}_2 = (1, 1, 1, 0, 0, 0, 0, 0)^T,$$

$$^2\mathbf{I}_{prop} = (1, 0, 1, 0, 1, 0, 1, 1)^T,$$

$$\mathbf{I}_3 = (0, -1, 0, 0, 1, 0, 1, 1)^T,$$

$$^3\mathbf{I}_{prop} = (1, 1, 1, 0, 0, 0, 0, 0)^T.$$

Consequently, when we use the proper invariants being considered to be invariants, the supports are the following:

$$P_{I_1} = \{p_4, p_5, p_6\},$$

$$P_{I_2} = \{p_1, p_3, p_5, p_7, p_8\},$$

$$P_{I_3} = \{p_1, p_2, p_3\},$$

while for PN displayed in Figure 3 there exists only one invariant, one proper invariant equal to this invariant, and one support invariant as follows:

$$\mathbf{I}_1 = (0, 0, 1, 1, 0, 0)^T,$$

$$^1\mathbf{I}_{prop} = (0, 0, 1, 1, 0, 0)^T,$$

$$P_{I_1} = \{p_3, p_4\}.$$

Simply said, P-invariants are the sets of places whose weighted token sum remains constant for all possible markings, while T-invariants are the sets of firings that will cause a cycle in the state space, meaning the comeback to the original state (markings). The set of nodes corresponding to non-zero entries of an invariant is called the support of this invariant $\mathbf{I}$, written as $\mathrm{supp}(\mathbf{I})$. An invariant $\mathbf{I}$ is called minimal if $\nexists\, \mathbf{I}' : \mathrm{supp}(\mathbf{I}') \subset \mathrm{supp}(\mathbf{I})$, i.e., its support does not contain the support of any other invariant $\mathbf{I}'$, and the greatest common divisor of all non-zero entries of $\mathbf{I}$ is 1.

Characteristic Vectors. Let, in general, the P-vector means a vector expressing states of places (number of tokens inside them) – in (1)–(2) it is the state vector $\mathbf{x}$ – and the T-vector means a vector expressing states of transitions (enabled, disabled) – in (1)–(2) it is the control vector $\mathbf{u}$.

Let $S \subseteq P$ be a subset of places of N. The $(n \times 1)$ vector $^S\boldsymbol{\sigma}$ is called [58, 51, 91] the *characteristic P-vector* of S if $\forall p \in S : {}^S\sigma_p = 1$; otherwise $^S\sigma_p = 0$. The $(m \times 1)$ vector $^S\boldsymbol{\gamma} = \mathbf{B}^T.{}^S\boldsymbol{\sigma}$ is called the *characteristic T-vector* of S.

The physical interpretation of the T-vector of a subset of places is the following:

1. $^S\gamma(t) > 0$ means that $^S\gamma(t)$ tokens are put into S when the transition t fires;

2. $^S\gamma(t) = 0$ means that the number of tokens in S does not change after t fires;
   (iii) $^S\gamma(t) < 0$ implies that $|^S\gamma(t)|$ tokens are removed from S when t fires.

Repetitive Vectors. In [72] the term *repetitiveness* was also introduced. Namely, N is said to be (partially) *repetitive* if there exists a marking $\mathbf{x}_0$ and a firing sequence $\mathcal{U} = \{t_a,\, t_b,\, t_c \dots\}$ from $\mathbf{x}_0$, i.e. $\mathbf{x}_0[t_a\rangle\mathbf{x}_1[t_b\rangle\mathbf{x}_2[t_c\rangle \dots$, such that every (some) transition occurs infinitely often in $\mathcal{U}$. It was proved there that N is (partially) repetitive *iff* there exists an $(m \times 1)$ vector $\mathbf{q}$ of positive (non-negative) integers such that $\mathbf{B.q} \geq \mathbf{0}$, $\mathbf{q} \neq \mathbf{0}$. Such vector $\mathbf{q}$ is named as the *repetitive vector*.

A repetitive vector $\mathbf{q}$ is reachable [23] *iff* there exists a reachable state (marking) $\mathbf{x} \in \mathcal{R}$ that allows firing a sequence $\mathcal{U}$ whose corresponding characteristic T-vector is $\mathbf{q}$.

The System Evolution. The (1)–(2) represent the discrete event system – DES. For completeness' sake it is necessary to introduce also the procedure of the marking development in PN (i.e. the system evolution).

Let us develop the system (1) from $\mathbf{x}_0$ to $\mathbf{x}_q$

$$\mathbf{x}_1 \;=\; \mathbf{x}_0 + \mathbf{B.u}_0 \tag{3}$$

$$\mathbf{x}_2 \;=\; \mathbf{x}_1 + \mathbf{B.u}_1 = \mathbf{x}_0 + \mathbf{B.u}_0 + \mathbf{B.u}_1 = \mathbf{x}_0 + \mathbf{B.}(\mathbf{u}_0 + \mathbf{u}_1) \tag{4}$$

$$\cdots \tag{5}$$

$$\mathbf{x}_q \;=\; \mathbf{x}_0 + \mathbf{B.}(\mathbf{u}_0 + \mathbf{u}_1 + \cdots + \mathbf{u}_{q-1}) = \mathbf{x}_0 + \mathbf{B.}\sum_{i=0}^{q-1}\mathbf{u}_i \tag{6}$$

By the way, the vector represented the sum in (6) is called (in general) the *Parikh's vector*. Its entries show how many times the particular transitions are fired during the system evolution. Denote this vector in our case as $\mathcal{P}_q$. Now, develop analogically the system from $\mathbf{x}_q$ to $\mathbf{x}_r$

$$\mathbf{x}_{q+1} \;=\; \mathbf{x}_q + \mathbf{B}.\mathbf{u}_q \tag{7}$$

$$\mathbf{x}_{q+2} \;=\; \mathbf{x}_q + \mathbf{B}.\mathbf{u}_q + \mathbf{B}.\mathbf{u}_{q+1} = \mathbf{x}_q + \mathbf{B}.(\mathbf{u}_q + \mathbf{u}_{q+1}) \tag{8}$$

$$\cdots \tag{9}$$

$$\mathbf{x}_r \;=\; \mathbf{x}_q + \mathbf{B}.(\mathbf{u}_q + \mathbf{u}_{q+1} + \cdots + \mathbf{u}_{r-1}) = \mathbf{x}_q + \mathbf{B}.\sum_{i=q}^{r-1} \mathbf{u}_i \tag{10}$$

Denote the vector represented the sum in (10) as $\mathcal{P}_r$. Hence,

$$\mathbf{x}_r = \mathbf{x}_0 + \mathbf{B}.(\mathcal{P}_q + \mathcal{P}_r). \tag{11}$$

The Parikh's vectors $\mathcal{P}_q$, $\mathcal{P}_r$ represent, respectively, not only the firing sequences $\mathcal{U}_1$ (from $\mathbf{x}_0$ to $\mathbf{x}_q$) and $\mathcal{U}_2$ (from $\mathbf{x}_q$ to $\mathbf{x}_r$) but also how many times particular transitions are fired during the developments (6) and (10). From the point of view of the state $\mathbf{x}_q$ we can speak about the input firing sequence $\mathcal{U}_1$ and output firing sequence $\mathcal{U}_2$ or about $\mathcal{P}_q$ and $\mathcal{P}_r$, respectively.

A live PN guarantees deadlock-free operation, no matter what firing sequence is chosen. Moreover, equations introduced above represent the analytical expression of the principle of causality in PN.

2.3 Controllability Conditions for PN vs. Invariants and Siphons

An ordinary net N is said to be completely controllable if any marking is reachable from any other marking. Details how invariants and siphons make possible to control DES represented by PN models are introduced in following sections.

In case of control synthesis based on P-invariants, the permissive controller (supervisor) have to fulfill some conditions imposed on mutual relations among states of particular PN markings (states) in order to avoid deadlock identified by means of the thorough analysis of RT (RG).

In case of siphon-based approach the thorough structural analysis of PN is performed. Then, the properties of found siphons are utilized at the control synthesis. Although this topic was opened long ago [72, 1, 2] it is still very live – see recent contributions [7, 24, 25, 51, 9, 10, 63, 8, 25, 55, 56, 57, 59, 62, 64, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40]. The basis of such an approach to control consists in avoiding of emptying (unmarking) of places creating the siphons. A siphon in ordinary PN is said to be controlled [24] if it cannot be empted (unmarked) at any reachable marking.

If PN is generalized, owing to the weights of arcs, the non-emptyability of a siphon is not sufficient for the absence of dead transitions, and the controllability of a siphon is much more complex.

Elementary and Dependent Siphons.   The overview and definitions of different kind of siphons in PN are given in [58]. The motivation to propose the concept of *elementary* siphons is to control *dependent* siphons by explicitly controlling their elementary siphons only.

Elementary siphons play an important role in the development of deadlock prevention approaches, that lead to structurally simple supervisors enforcing liveness, based on monitors.

The set $\Pi_E = \{S_\alpha, S_\beta, \ldots, S_\gamma\}$, $\{\alpha, \beta, \gamma\} \subseteq \mathbb{Z}$, is called the set of elementary siphons if $\{\boldsymbol{\gamma}_\alpha, \boldsymbol{\gamma}_\beta, \ldots, \boldsymbol{\gamma}_\gamma\}$ is a linearly independent maximal set of the matrix $^S\boldsymbol{\Gamma}$ consisting of T-vectors $^S\boldsymbol{\gamma}$ – i.e. $^S\boldsymbol{\Gamma} = \mathbf{B}^T.^S\boldsymbol{\Sigma}$, where $^S\boldsymbol{\Sigma}$ is the matrix consisting of P-vectors $^S\boldsymbol{\sigma}$.

The T-vector $^S\boldsymbol{\gamma}$ is associated [91] with each siphon S such that $^S\gamma(i)$ is the number of tokens gained in or lost from S by firing the transition t_i once. A dependent siphon S_0 strongly depends on elementary siphons S_1, S_2, ..., S_k if

$$^S\gamma(0) = a_1.^S\gamma(1) + a_2.^S\gamma(2) + \cdots + a_k.^S\gamma(k) \tag{12}$$

with $a_i \in \mathbb{Z}_{\geq 0}$, $i = 1, 2, \ldots, k$, being positive integers and $^S\boldsymbol{\gamma}(k)$ being nonzero entries of $^S\boldsymbol{\gamma}$. Such dependent siphons are named as the strongly dependent siphons (SDS). S_0 is a weakly dependent siphon (WDS) if some a_i are negative. The T-vectors for elementary siphons are mutually independent. More details can be found e.g. in [47, 48, 49, 50, 51] as well as in [63, 8, 25, 55, 56, 57, 59, 62, 64, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40].

More details can be seen e.g. in [47, 48, 49, 50, 51] as well as in a great deal of applications in [63, 8, 25, 55, 56, 57, 59, 62, 64, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40], where also newest approaches to the siphon-based control of AMS/FMS and RAS are presented as follows.

In [56] the new controllability condition for siphons is presented. In [59] the necessary and sufficient condition of a kind of PN (namely GS³PR) was proved. In [60, 62, 64] important findings in the area of robust deadlock control of AMS with unreliable resources are published. In [53, 50, 49, 48, 47, 91] the application of elementary siphons, being topical at present, is broadly investigated. In [55, 28, 29, 30, 31, 32, 33] the very useful iterative solution how to avoid the need of enumerating all the states or siphons using mathematical programming techniques were published. In [35, 36] also distributed resolution approaches to solving the deadlock avoiding were published. In [37] the very useful approach to simplification of the supervisor structures was published. In [37] the supervisor synthesis and performance improvement in an integrated way are also presented. In the works [38, 39, 40] the direct application of assembly AMS in the practice was shown.

2.4 Resource Allocation Systems vs. Petri Nets

RAS represent [86] a special class of concurrent systems, especially AMS/FMS, where the attention is focused on resources. RAS consist of a finite set of processes that share (in a competitive way) a finite set of resources. Such a competition can bring (i.e., is conducive to) existence of deadlocks. The deadlock causes an unacceptable state when some processes in AMS/FMS are waiting for the evolution of other processes that are also waiting for the evaluation of former ones in order to evolve.

PN models of RAS are especially useful at synthesizing deadlock prevention policies as well as deadlocks avoidance ones. Although many papers about RAS were published in the last three decades, it may be said that principle papers about RAS are [46, 78, 79, 80, 81, 75, 74, 73, 82, 86, 87]. Newer papers with very important contributions in this area are especially [63, 8, 25, 55, 56, 57, 59, 62, 64, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 92, 93, 94, 91, 88, 89, 65, 66, 67, 68, 69, 58, 60, 61, 47, 48, 49, 50, 51, 52, 53].

There exist several standard kinds (paradigms) of RAS [41, 51, 22, 21, 94]. Specific nomenclatures have been established, e.g. Simple Sequential Process (S^2P), Simple Sequential Process with Resources (S^2PR), Systems of Simple Linear Sequential Processes with Resources (S^2LSPR), Systems of Simple Sequential Processes with Multiple Resources (S^3PMR), the subclass of System of Simple Sequential Processes with General Resource Requirements (S^3PGR2) [75], Generalised Systems of Simple Sequential Processes with Resources (GS^3PR), Systems of Simple Sequential Processes with Resources (S^3PR), Linear S^3PR (LS^3PR), Extension of S^3PR (ES^3PR), and already mentioned S^3PGR2 modelling manufacturing systems in general, Weighted System of Simple Sequential Processes with Several Resources (WS^3PSR), System of Sequential Systems with Shared Resources (S^4R), System of Sequential Systems with Shared Process Resources (S^4PR), etc.

The S^3PR are frequently used in AMS and they are modelled by means of PN. They represent a class of AMS with flexible routing and single-unit resource acquisition. In such systems the part being produced using only one copy of one resource at each processing step. Such systems create a subclass of a higher (upper) class S^*PR [94, 21] where more copies of one resource are allowed. The asterix does not represent exactly an integer expressing the number of copies, but a level of complexity. Multiple-unit systems with routing flexibility are much less investigated. S^4PR are explored a lot less than S^3PR. They are adequate [86, 87] for the modeling of a wide variety of RAS. The special syntactic characteristics of this class allow to study the modelled systems from a structural perspective. The newer publications [63, 8, 25, 55, 56, 57, 59, 62, 64, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 92, 93, 94, 91, 88, 89, 65, 66, 67, 68, 69, 58, 60, 61, 47, 48, 49, 50, 51, 52, 53] investigate also such kinds of AMS/FMS.

The relation among some of PN-based models of RAS is illustrated in Figure 4.

From this point of view two kinds of PN places (being added to the PN model because of the siphon control) can be distinguished as to the synthesis of AMS/FMS
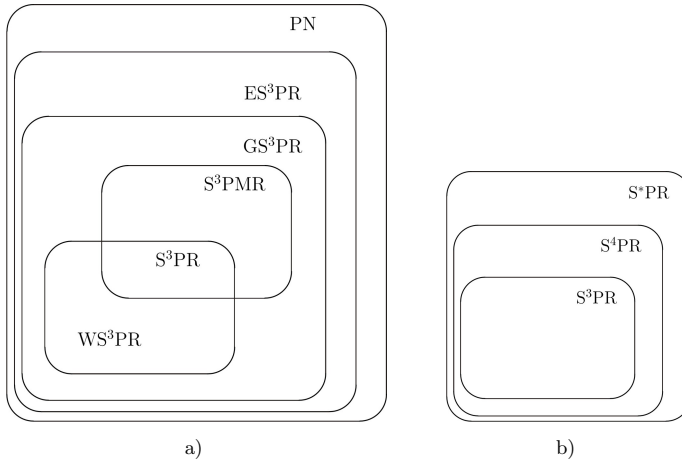
Figure 4. Relations (based on sets) among some of more important PN models of RAS

control, namely:

1. ordinary places;
2. weighted places.

Ordinary places have ordinary arcs and are added to the original PN in order to prevent related siphon from becoming unmarked whenever it is possible. Weighted places adopt a conservative policy controlling the release of component or parts in AMS/FMS, modelled by PN, into the system. It means that they are added to the original/modified PN, namely to the source transitions of the resultant PN, by means of their output arcs.

2.5 Literature Survey

In case of the approach based on P-invariants most important contributions can be seen in [42, 43, 71, 88, 89], but also in many other works.

However, in about the last two decades the siphon-based approach dominates among the methodologies that deal with the deadlock analysis and control of resource allocation systems. This research has uncovered many useful results. Very important share on the development of the siphon-based approach to deadlock avoiding have the works [63, 8, 25, 55, 56, 57, 59, 62, 64, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 92, 93, 94, 91, 88, 89, 65, 66, 67, 68, 69, 58, 60, 61, 47, 48, 49, 50, 51, 52, 53], but also some others. The new controllability condition for siphons is presented in [56]. The necessary and sufficient condition of a kind of PN (namely GS^{3}PR) was proved in [59]. Very important are findings in the area of robust deadlock control of AMS with unreliable resources [60, 62, 64]. The application of elementary siphons, being topical at present, is broadly investigated in [53, 50, 49, 48, 47, 91]. Very useful

is the iterative solution using mathematical programming techniques to avoid the need of enumerating all the states or siphons published in [55, 28, 29, 30, 31, 32, 33] having direct impact on practice. Also distributed resolution approaches to solving the deadlock avoiding were published [35, 36]. Very useful is the approach to simplification of the supervisor structures [37]. The supervisor synthesis and performance improvement in an integrated way are also presented in [37]. For the direct application in the practice of assembly AMS are important the works [38, 39, 40].

A suitable combination of the approaches in the form of invariant-controlled elementary siphons is presented in [50].

Of course, in this paper it is impossible to devote to the complete problem of the deadlock avoidance in AMS/FMS. Here, in the Part 1, only a broader introductory part to the problem will be inducted and illustrated by simple explanatory examples. The permissible scope of this article does not allow more. In the second part – Part 2, being in preparation, newer methods of deadlock avoidance as well as their application to more complicated cases of AMS/FMS will be analyzed. May be that also a third part – Part 3 will be necessary because of the limited space for one paper in this journal.

2.6 The Paper Organization

After the detailed introduction in Section 1 and preliminaries in Section 2, which were necessary for initiation into the problem of the widely developed subject inside PN, the paper includes the next parts.

In Section 3, solving problems in RAS is introduced – namely, the description how to remove deadlocks and how to control RAS. This section is the core of the paper. It consists of three subsections. Subsection 3.1 details the proposal and application of the P-invariant method for synthesizing the supervisor removing deadlocks in RAS while Subsection 3.2 presents the proposal and application of siphon-based method for synthesizing the supervisor removing deadlocks in RAS. Subsection 3.3 presents a short comparison and evaluation of both approaches. Section 4 introduces the research plan for the future. Section 5 concludes the paper.

3 SOLVING PROBLEMS WITH DEADLOCKS IN RAS

Two possible approaches to remove deadlocks by means of the supervisory control are presented here.

The first one starts from the thorough analysis of RT of the PN model of RAS or equivalently from RG. It is necessary to say that RG arises from RT by means of joining all RT leaves with the same name into one node of RG. The adjacency matrix is the same for both RT and RG. From RT/RG (expressed either in graphical form or in the form of the adjacency matrix), information about deadlocks can be obtained. After finding deadlocks, the supervisor based on P-invariants is synthesized in order to remove these deadlocks.

The second approach performs the thorough structural analysis of the PN model of RAS. It finds and uses PN siphons and traps to synthesize the supervisor removing deadlocks. No RT/RG is necessary in this case.

Each of the mentioned approaches has its advantages and disadvantages. Therefore, we will test both approaches on simple practical examples and compare them. In Part 2 (potentially also in Part 3) more complicated cases of RAS will be tested using the newest findings in that field.

3.1 An Approach Based on P-Invariants with Current Knowledge of RG

After a detailed analysis of RG the deadlocks can be identified. Then, the conditions for supervisor synthesis based on P-invariants may be established.

Let us try to design a controller based on P-invariants – see e.g. [4]. Let the matrix $\mathbf{Y}$ denote the $(s \times n)$ matrix of P-invariants, which are not known till now. Start from the definition of P-invariants

$$\mathbf{Y}^T.\mathbf{B} \overset{!}{=} \mathbf{0}. \tag{13}$$

Consider the restrictive condition (following from the detail analysis of RG) on the state vector in the form as follows:

$$\mathbf{L}.\mathbf{x} \leq \mathbf{b} \tag{14}$$

where $\mathbf{L}$ is the $(s \times n)$ matrix of integers expressing the expected relations among states to eliminate deadlocks, and $\mathbf{b}$ is the $(s \times 1)$ vector of positive integers determining some restrictions on linear combinations of corresponding entries of the state vector $\mathbf{x}$. In order to remove the inequality (14), add a slack vector $\mathbf{x}_s$ and put

$$\mathbf{L}.\mathbf{x} + \mathbf{x}_s = \mathbf{L}.\mathbf{x} + \mathbf{I}_s.\mathbf{x}_s = (\mathbf{L}\mathbf{I}_s).(\mathbf{x}^T\mathbf{x}_s^T)^T = \mathbf{b} \tag{15}$$

where the $(n_s \times 1)$ vector $\mathbf{x}_s$ consists of *slack variables* and $\mathbf{I}_s$ is the $(s \times s)$ identity matrix. Now, when we force $(\mathbf{L}\mathbf{I}_s)$ instead of $\mathbf{Y}^T$, we obtain $(\mathbf{L}\mathbf{I}_s).(\mathbf{B}^T\mathbf{B}_s^T)^T \overset{!}{=} \mathbf{0}$. Hence, the structural matrix and the initial state of the supervisor are

$$\begin{aligned} \mathbf{B}_s &= -\mathbf{L}.\mathbf{B}, \\ \mathbf{x}_s^0 &= \mathbf{b} - \mathbf{L}.\mathbf{x}_0 \end{aligned} \tag{16}$$

where $\mathbf{B}_s = \mathbf{G}_s^T - \mathbf{F}_s$. Then the extended PN model (the original uncontrolled PN model together with the supervisor) has the following structural matrix and the initial state.

$$\begin{aligned} \mathbf{B}_{ex} &= \begin{pmatrix} \mathbf{B} \\ \mathbf{B}_s \end{pmatrix}, \\ \mathbf{x}_{ex}^0 &= \begin{pmatrix} \mathbf{x}_0 \\ \mathbf{x}_s^0 \end{pmatrix}. \end{aligned} \tag{17}$$

The approach can be explained in details by means of the following simple example illustrating the primary problem of RAS – removing deadlocks. There it will be presented what difficulties can deadlocks cause in RAS as well as ways how to deal with them.

3.1.1 Example 1

Consider the very simple PN model in Figure 5 a). The corresponding RT is shown in Figure 5 b). There it can be seen that the state No. 9 (i.e. $\mathbf{x}_9$), being the 10th column of the matrix $\mathbf{X}_r$ in (20) (because the numbering of reachable states starts from 0), represents the deadlock.
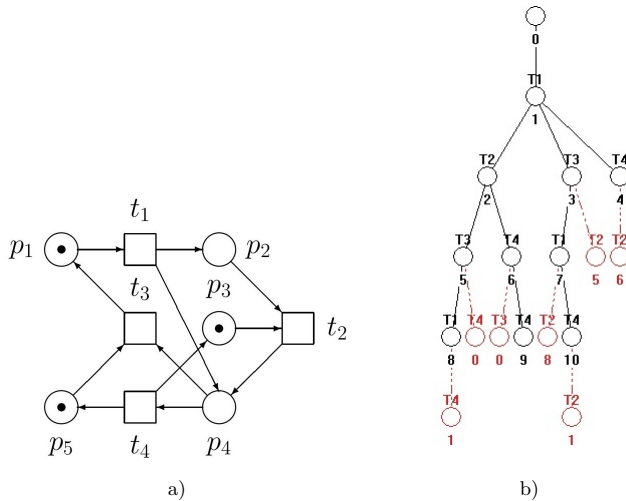

Figure 5. The simple example of a) the deadlocked PN and b) its RT

Therefore, it is necessary to avoid the deadlock. Let us demonstrate removing the deadlock by a supervisor synthesis based on P-invariants.

The P-invariant based approach starts by the thorough analysis of the RT. Doing so we can see that in order to eliminate the deadlock it is necessary to ensure the priority $t_3 \succ t_4$. Namely, it follows from the "fork" emerging from the state No. 2 (i.e. $\mathbf{x}_2$) being the 3rd column in (20), as well as from the "fork" emerging from the state No. 6 (i.e. $\mathbf{x}_6$) being the 7th column in (20).

In this *specific case* it is possible to do this elimination very simply – by adding p_6 to the original PN model and interconnect it with the PN model by the arcs from t_3 to p_6 and from p_6 to t_4. This ensures the priority $t_3 \succ t_4$ in RT given in Figure 5. It is clear from Figure 6 a) and from RT (right). However, the states of the controlled model displayed in Figure 6 as well as the numbers of RT nodes are different from the states and numbers of RT nodes of the uncontrolled model given in Figure 5.
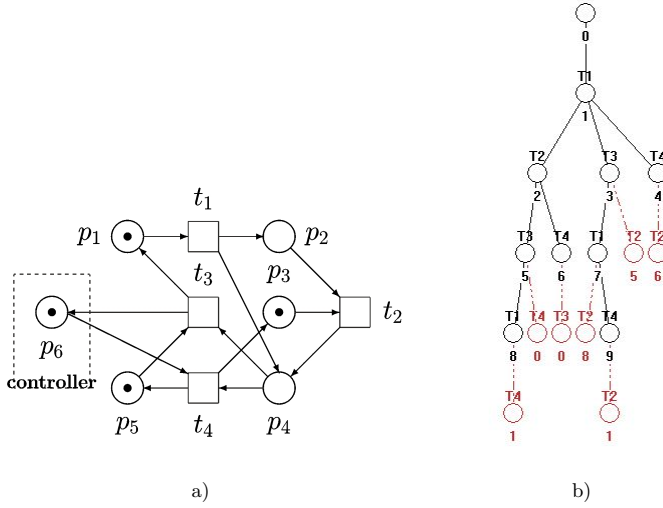
Figure 6. The a) supervised PN model and b) its RT

However, the *P*-invariant based approach analytically described by (13)–(16) is *general one*. The described procedure of the supervisor synthesis presented in [4] can impose the restriction on the state vector $\mathbf{x}$ in the form (14). In our case

$$\mathbf{L} = (0, 0, 1, -1, 0) \qquad \text{and} \qquad \mathbf{b} = (1) \tag{18}$$

where the matrix $\mathbf{L}$ is represented by the row vector and the restriction $\mathbf{b} = (1)$ is the scalar, because only one of two possibilities (firing either the transition t_3 or transition t_4) is possible.

To perform the analytical expression of the approach let us introduce the following. The structural matrix of the original PN model, the initial state $\mathbf{x}_0$ and RT nodes in (20) are as follows:

$$\mathbf{B} = \mathbf{G}^T - \mathbf{F} = \begin{pmatrix} -1 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 1 & 1 & -1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}, \quad \mathbf{x}_0 = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \tag{19}$$

$$\mathbf{X}_r = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 2 & 1 & 0 & 2 \\ 1 & 1 & 0 & 1 & 2 & 0 & 1 & 1 & 0 & 2 & 2 \\ 0 & 1 & 2 & 0 & 0 & 1 & 1 & 1 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 2 & 0 & 2 & 0 & 0 & 3 & 1 \end{pmatrix}. \tag{20}$$
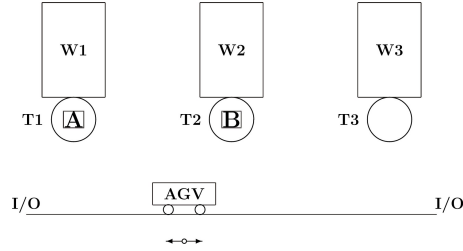
Because of (16), the structural matrix $\mathbf{B}_s$ of the supervisor and the initial state $\mathbf{x}_0^s$ of the supervisor are

$$\mathbf{B}_s = (1, 2, -1, -2) \text{ and } \mathbf{x}_0^s = (0). \tag{21}$$

The structure of the supervisor is

$$\mathbf{B}_s = \mathbf{G}_s^T - \mathbf{F}_s \text{ where } \mathbf{G}_s^T = (1, 2, 0, 0) \text{ and } \mathbf{F}_s = (0, 0, 1, 2). \tag{22}$$

The supervisor has very simple structure – it is represented by the place p_6. Hence, the supervised PN model is given in Figure 7 a). The corresponding RT is shown in Figure 7 b). As we can see in RT, no deadlock is detected. It is necessary to add that the states of the controlled PN in such a way as well as the RT nodes in this case are different from the states and RT nodes in previous two cases.

Figure 7. The a) supervised PN by means of P-invariant and b) its RT

3.1.2 Example 2

Let us introduce now a practical example. A cell of AMS/FMS consists of three workstations, W1 with a robot R1, W2 with a robot R2, and W3 with a robot R3, and a single AGV (being served by the robots) that transports parts among the workstations and the input and output ports I/O of the cell. The simple scheme of the cell is displayed in Figure 8.

There is exercised the concurrent production of the two process types with planes as follows – P1: W1 $\rightarrow$ W2 $\rightarrow$ W3 and P2: W3 $\rightarrow$ W2 $\rightarrow$ W1. P1 produces parts of a kind A while P2 produces parts of a kind B. Each workstation has a working table

Figure 8. The example of the real RAS with the deadlock

on which only one workpiece at a time can be held. It is evident from the routing information about processes P1, P2 that none of the currently loaded parts is able to go forward to the next workstation, because the corresponding working table is occupied by the other part. This situation illustrate the deadlock that can be met in this AMS/FMS. Simultaneously, the Figure 8 can be understood to be a kind of RAS.

The PN model of the cell and its RT are given in Figure 9. Its parameters are

$$
\mathbf{B} = \mathbf{G}^T - \mathbf{F} =
\begin{pmatrix}
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\
-1 & 1 & 0 & 0 & 0 & 0 & -1 & 1 \\
0 & -1 & 1 & 0 & 0 & -1 & 1 & 0 \\
0 & 0 & -1 & 1 & -1 & 1 & 0 & 0
\end{pmatrix}
, \quad \mathbf{x}_0 =
\begin{pmatrix}
1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1
\end{pmatrix}
\quad (23)
$$

The places p_1, p_2, p_3 express the presence of parts A on the working tables, while p_4, p_5, p_6 express the presence of parts B on the working tables. The resource availability is monitored by the marking of the resource places p_7, p_8, p_9.

As we can also see on the PN model of the RAS in Figure 9 a), and especially on RT in Figure 9 b), there is the deadlock in the state $\mathbf{x}_1 = (1\,0\,0\,1\,1\,0\,0\,0\,0)^T$. It occurs immediately after firing t_5 at the initial state $\mathbf{x}_0 = (1\,0\,0\,0\,1\,0\,0\,0\,1)^T$.

To control the system consider the natural initial state $\mathbf{x}_0 = (0\,0\,0\,0\,0\,0\,1\,1\,1)^T$ when all resources are available at the beginning. The corresponding RT is given in Figure 10.

As we can see there, the states $\mathbf{x}_{13} = (1\,1\,0\,1\,0\,0\,0\,0\,0)^T$ and $\mathbf{x}_{14} = (1\,0\,0\,1\,1\,0\,0\,0\,0)^T$, i.e., the 14th and 15th column of the following matrix of reachable states $\mathbf{X}_r$ are deadlocks. All nodes of the RT, where the first column represents the initial
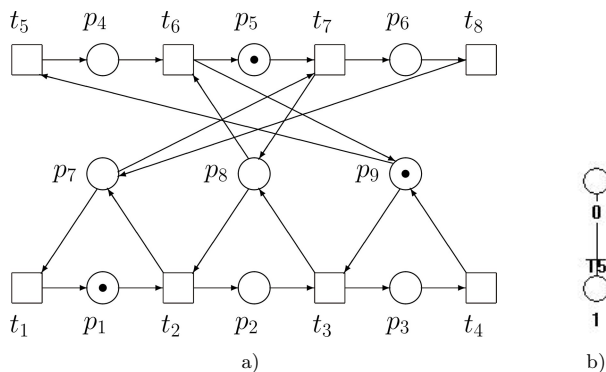
Figure 9. The a) PN model and b) corresponding RT of the RAS

state $\mathbf{x}_0$, are expressed by particular columns as follows:

$$
\mathbf{X}_r = \begin{pmatrix}
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & |1| & |1| & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & |1| & |0| & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & |0| & |0| & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & |1| & |1| & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & |0| & |1| & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & |0| & |0| & 1 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & |0| & |0| & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & |0| & |0| & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & |0| & |0| & 0 & 0 & 1 & 0 & 0
\end{pmatrix}. \qquad (24)
$$

3.1.3 Detail Analysis of RT

The detail analysis of RT yields the following results:

1. the sequences of firing $t_2 \succ t_5$ as well as $t_5 \succ t_2$ have to be forbidden, i.e., p_2 and p_4 cannot be active simultaneously;

2. the sequences of firing $t_1 \succ t_6$ and $t_6 \succ t_1$ have to be forbidden, i.e., p_1 and p_5 cannot be active simultaneously;

3. the sequences of firing $t_1 \succ t_5$ as well as $t_5 \succ t_1$ have to be forbidden, i.e., p_1 and p_4 cannot be active simultaneously.

Thus, putting inequalities $\sigma_{p_2} + \sigma_{p_4} \leq 1$, $\sigma_{p_1} + \sigma_{p_5} \leq 1$, and $\sigma_{p_1} + \sigma_{p_4} \leq 1$ we can synthesize the supervisor. As we can see below, these places create the nonzero entries of the matrix $\mathbf{L}$, and right sides of the inequalities create the vector $\mathbf{b}$. Putting the initial state of the PN model of RAS as $\mathbf{x}_0 = (0\,0\,0\,0\,0\,0\,1\,1\,1)^T$ (i.e., when all three resources are available), the process of the supervisor synthesis is the following.
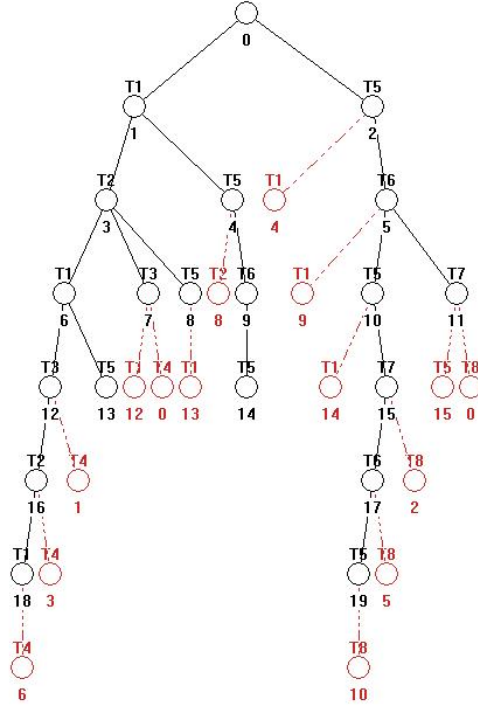
Figure 10. The corresponding RT of the RAS model

3.1.4 Supervisor Synthesis

The anterior inequalities create the restrictive condition with $\mathbf{L}$ and $\mathbf{b}$ in the form

$$\mathbf{L} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \tag{25}$$

$$\mathbf{B}_s = -\mathbf{L}.\mathbf{B} = \begin{pmatrix} 0 & -1 & 1 & 0 & -1 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & -1 & 1 & 0 \\ -1 & 1 & 0 & 0 & -1 & 1 & 0 & 0 \end{pmatrix}, \quad \mathbf{x}_s^0 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}. \tag{26}$$

Here, the positive nonzero entries of $\mathbf{B}_s = \mathbf{G}_s^T - \mathbf{F}_s$ create nonzero entries of $\mathbf{G}_s^T$, while negative nonzero entries of $\mathbf{B}_s$ create nonzero entries of $(-\mathbf{F}_s)$.

The PN model of the supervised system is displayed in Figure 11 where the controller is represented by the triplet $\{p_{10}, p_{11}, p_{12}\}$. RT of such system is given in Figure 12.
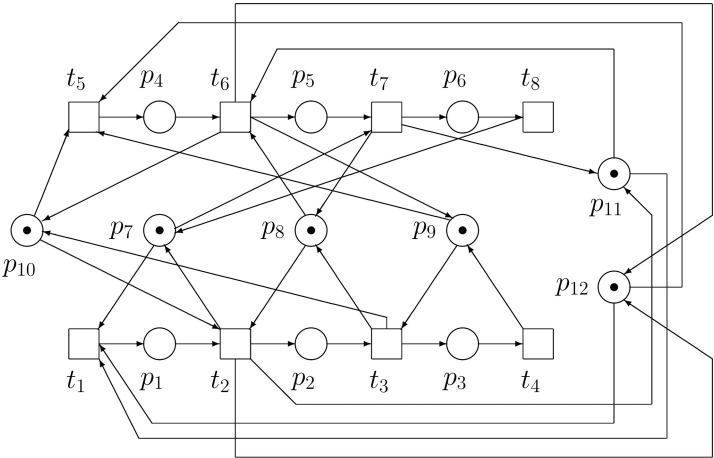
Figure 11. The controlled PN model
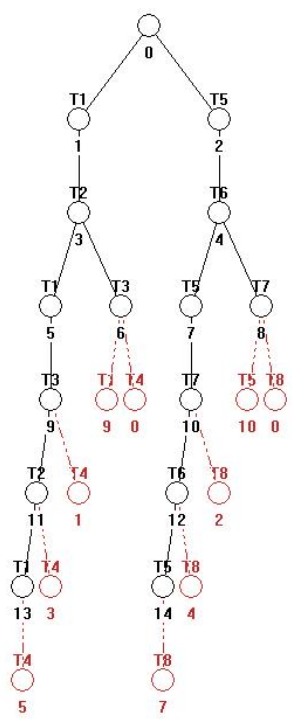
Figure 12. The RT of the controlled PN model

Of course, the incidence matrices and the initial state of the controlled system are

$$\mathbf{F}_{cs} = \left(\begin{array}{c} \mathbf{F} \\ \mathbf{F}_s \end{array} \right), \quad \mathbf{G}_{cs}^T = \left(\begin{array}{c} \mathbf{G}^T \\ \mathbf{G}_s^T \end{array} \right), \quad \mathbf{x}_{cs}^0 = \left(\begin{array}{c} \mathbf{x}_0 \\ \mathbf{x}_s^0 \end{array} \right) \quad (27)$$

where $\mathbf{F}$, $\mathbf{G}^T$ are incidence matrices of the uncontrolled RAS and $\mathbf{x}_0$ is its initial state.

The RT of the controlled RAS is given in Figure 12. No deadlock can be seen in Figure 12. The particular nodes of this RT are expressed by the columns of the matrix

$$\mathbf{X}_r = \left(\begin{array}{ccccccccccccccc} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{array} \right) \quad (28)$$

where the first column represents the initial state $\mathbf{x}_{cs}^0$ of the controlled RAS.

3.1.5 Example 3

Let us add a simple example relating the deadlocked S^3PR PN model. The problem of a deadlock in S^3PR can be resolved also by means of adding a transition to the original deadlocked PN model of RAS. Namely, imbedding such a transition into the deadlocked PN model it is possible to achieve initial state from the deadlocked state. Then, the development of the model can proceed from the initial state in some other way. It was proved in [83] for S^3PR kind of deadlocked PN model. Applying this on (6), the single additional transition brings the structural matrix of the supervisor $\mathbf{B}_s = \mathbf{x}_0 - \mathbf{x}_q$. Consider the deadlocked PN model given in Figure 13 a) having RT displayed in Figure 13 b) where the RT node No. 4 (i.e. $\mathbf{x}_5$) is the deadlock in question.

Because $\mathbf{x}_0 = (2, 0, 1, 0, 0, 1, 0, 2)^T$ and $\mathbf{x}_5 = (1, 1, 0, 0, 0, 0, 1, 1)^T$, after calculation $\mathbf{B}_s = (1, -1, 1, 0, 0, 1, -1, 1)^T$. Thus, $\mathbf{F}_s = (0, 1, 0, 0, 0, 0, 1, 0)^T$ (arcs from places to the transition) and $\mathbf{G}_s^T = (1, 0, 1, 0, 0, 1, 0, 1)^T$ (arcs from the transition to places). Hence, the supervised PN model without the deadlock is given in Figure 14 a) and its RT is displayed in Figure 14 b).

Sometimes such a procedure can be used in more complicated RT having so called *diamond(s)* between two different nodes. Even, the paths (left and right)
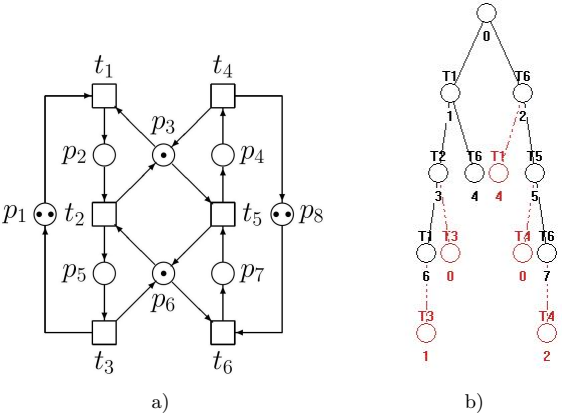
Figure 13. The a) PN model of RAS with the deadloch $\mathbf{x}_5$ and b) its RT
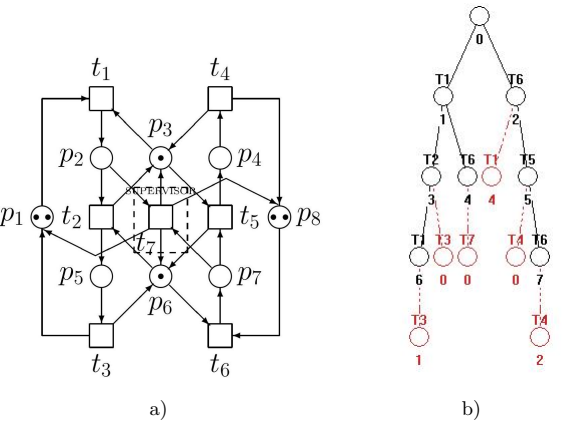

Figure 14. The a) supervised PN model by means of the added transition t_7 and b) its RT

creating the diamond(s) may be longer than that in RT displayed in Figure 14 a) and Figure 14 b). More about this will be said in Part 2 of this paper.

3.1.6 Local Conclusion

The approach based on P-invariants has an exact analytically expressed procedure. After thorough analysis of RG and finding conditions how to mutually eliminate states of relevant places (in order to eliminate deadlocks), the procedure of the supervisor synthesis is very clear and simple. Introduced examples illustrate that the deadlock problem can be resolved relatively simply. However, in a more complicated

structure of the PN model of RAS with RT having too patulous branching, just the choice of the restrictions in the form of inequalities may be complicated or even impossible. Therefore, it is also necessary to look for another approaches or combinations of them.

3.2 Approach Based on Siphons Without the Need to Know RG

Based on the previous findings, especially those being introduced in Section 2 (Preliminaries), we have to unconditionally control siphons. Of course, first of all we have to know them, i.e., we have to compute them. There are several algorithms how to do this. Although the calculation of siphons is not a subject of this paper, let us mention at least several approaches to it by means of:

1. solving logical equations, namely a siphon S has to satisfy a set of conditions: $\forall t_i \in T : t_i^\bullet \cap S \neq \varnothing \Rightarrow {}^\bullet t_i \cap S \neq \varnothing$ – see e.g. [44];

2. linear algebraic calculation – see e.g. [45];

3. Thelen's prime implicant method – see [90];

and some others. However, for working in Matlab the most useful seems to be the GPenSIM tool developed by Davidrajuh [12, 13] for PN, which is able to calculate (among other things) also siphons and traps as well as minimal siphons and minimal traps.

The problem of deadlock avoidance in DES is equivalent to the problem of the avoidance of empty siphons in the original ordinary PN model. The siphon based control of a deadlocked PN has to guarantee that none of its siphons ever becomes empty. Unfortunately, this approach does not have as much analytical support as in the first approach based on P-invariants. Therefore, it is necessary to work with graphical tools for the PN modelling and analyzing, more than in the first approach. The siphon behaviour is such that if it has no token in a state (marking) of PN, then it remains without any token in each successor state. The trap behaviour is such that if it has at least one token in a state (marking) of PN, then it remains marked under each successor state.

It can be said that a siphon can only *lose* tokens whereas a trap can only *gain* tokens. Therefore, arising out of these properties, we want to utilize siphons and traps for analyzing of PN liveness as well as for synthesizing supervisors in order to avoid deadlocks in PN models.

Siphons are tied with deadlocks especially in PN models of RAS. As it was already mentioned, once a siphon loses all its tokens, it remains unmarked at any subsequent markings that are reachable from the current marking – see e.g. [22]. If a siphon is emptied at a certain marking, some of its output transitions would never be enabled. This leads to a deadlock. There exist many papers about deadlock prevention which have been based on siphons. They especially add monitors (additional places) to the PN model for strict minimal siphons in order to achieve

deadlock prevention. In this paper using the minimal siphons will be illustrated on simple examples.

On the other hand, there exist newer papers – e.g. [63, 8, 25, 55, 56, 57, 59, 62, 64, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 92, 93, 94, 91, 88, 89, 65, 66, 67, 68, 69, 58, 60, 61, 47, 48, 49, 50, 51, 52, 53]. They bring very interesting contributions in the form of new approaches how to deal with deadlocks by means of siphons, as it already was mentioned in Section 2.5. Some of them utilize elementary siphons. These newer approaches will be analyzed in Part 2 of this paper, which is being prepared.

A siphon S is said to be controlled in a net (N, M_0) *iff* $\forall M \in R(N, M_0)$, $M(S) > 0$. Hence, any siphon that contains a marked trap is controlled, since the marked trap can never be emptied. In an ordinary Petri net, a siphon that is controlled does not cause any deadlock.

Siphon S in an ordinary net system (N, M_0) is [51] invariant-controlled by P-invariant I under M_0 *iff* $L^T.M_0 > 0$ and $\forall p \in P \backslash S, I(p) \leq 0$, or equivalently, $l^T.M_0 > 0$ and $||I||^+ \subseteq S$. Here $||I||^+ = p \in P \mid I(p) > 0$ is the positive support of P-vector I.

Briefly, a siphon S is said to be controlled if it can never be emptied, and it is said to be invariant-controlled by P-invariant I if $l^T.M_0 > 0$ and $||I||^+ \subseteq S$.

The problem of deadlock avoidance in DES is equivalent [88, 89] to the problem of avoidance of empty siphons in ordinary PN model of DES. This is very important especially in real AMS/FMS. Siphons that do not contain other siphons are named as minimal siphons. It is sufficient to consider only minimal siphons at the supervisor synthesis. Hence, it is necessary to ensure that the sum of the number of tokens in each minimal siphon S is never less than one in any reachable marking. Thus, the general condition for i^{th} siphon $\mathbf{S}_i.\mathbf{x} \geq b$ proceeds into the form $\mathbf{S}_i.\mathbf{x} \geq 1$.

Generally, the main purpose of control of DES by means of PN is to avoid *undesirable* or *illegal* markings. In [42, 43, 88] an appropriate formal specification

$$\mathbf{l}^T.\mathbf{x} \geq b \quad \text{i.e.} \quad b - \mathbf{l}.\mathbf{x} \leq 0 \tag{29}$$

is proposed where $\mathbf{l}$ is a $(s \times 1)$-dimensional weight row vector; $\mathbf{x}$ is a state vector (i.e. marking); and b is a scalar. Verbally it means that the weighted sum of the number of tokens in each place should be greater than or equal to a constant. The theorem was proved there that if a PN with incidence matrix $\mathbf{B}$ satisfies $b - \mathbf{l}^T.\mathbf{x}_0 \leq 0$, then a control place p_c can be added which enforces the previous inequality (29). When $\mathbf{b}_c : T \to \mathbb{Z}$ denotes the weight vector of arcs connecting p_c with the transitions in the PN, the $\mathbf{b}_c$ can be obtained by

$$\mathbf{b}_c = \mathbf{l}^T.\mathbf{B} \tag{30}$$

and the initial number of tokens in p_c is

$$\mathbf{x}_0(p_c) = \mathbf{l}^T.\mathbf{x}_0 - b \geq 0. \tag{31}$$

The control place p_c enforces maximally permissive control strategy (or logic). It means that the only reachable markings of the original net N, that p_c avoids, are those violating [42, 43]. Here, the $\mathbf{b}_c$ is the row extending the matrix $\mathbf{B}$ with respect to p_c.

In general (for more additive places $p_1, \ldots, p_s$) it can be written the following

$$\mathbf{L}.\mathbf{x} \geq \mathbf{b} \quad \text{or} \quad \mathbf{L}.\mathbf{x} - \mathbf{x}_c = \mathbf{b} \tag{32}$$

where $\mathbf{L} \in \mathbb{Z}_{\geq 0}^{s \times n}$ having the weighted vectors $\mathbf{l}_i^T$, $i = 1, \ldots, s$ as its rows; $\mathbf{b} \in \mathbb{Z}_{\geq 0}^{s \times 1}$ is the vector of restrictions and $\mathbf{x}_c \in \mathbb{Z}_{\geq 0}^{s \times 1}$ is the vectors of slacks. We can found the extended incidence matrix $\mathbf{B}_{ex}$ and the initial state of extended state vector $\mathbf{x}_{ex}^0$ as follows:

$$\mathbf{B}_c = \mathbf{L}.\mathbf{B}, \tag{33}$$

$$\mathbf{x}_c^0 = \mathbf{L}.\mathbf{x}_0 - \mathbf{b} \overset{!}{\geq} \mathbf{0}, \tag{34}$$

$$\mathbf{B}_{ex} = \left(\begin{array}{c} \mathbf{B} \\ \mathbf{B}_c \end{array} \right), \quad \mathbf{x}_{ex}^0 = \left(\begin{array}{c} \mathbf{x}_0 \\ \mathbf{x}_c^0 \end{array} \right) \tag{35}$$

where $\mathbf{B}_c = \mathbf{G}_c^T - \mathbf{F}_c$ is the matrix corresponding to s additive places (monitors) $p_1, \ldots, p_s$.

Putting $\mathbf{l}_i^T \overset{!}{=} S_i$, where S_i is the ith PN siphon, or in general $\mathbf{L} \overset{!}{=} \mathbf{S}_m$ where $\mathbf{S}_m$ is the matrix of all PN siphons (being its rows), we have the structure of the supervisor as follows $\mathbf{B}_c = \mathbf{S}_m.\mathbf{B}$.

3.2.1 Example 4

Consider the same PN model given in Figure 5. In the PN siphon-based approach we have to find siphons and traps.

Minimal siphons in this net are $\{p_1, p_3, p_4\}$, $\{p_1, p_2, p_4\}$ and in the matrix form:

$$\mathbf{S}_m = \left(\begin{array}{ccccc} 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{array} \right). \tag{36}$$

Minimal traps in this net are $\{p_1, p_3, p_4\}$, $\{p_1, p_4, p_5\}$ and in the matrix form:

$$\mathbf{T}_m = \left(\begin{array}{ccccc} 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \end{array} \right). \tag{37}$$

We can see that the first siphon is equal to the first trap. This is also clear from comparing first rows of the matrices $\mathbf{S}_m$ and $\mathbf{T}_m$. Such a siphon is out of our interest because it cannot be emptied once it is initially marked. Namely, this siphon contains the marked trap, i.e., there is no deadlock threat.

Now, we have to consider the second siphon $\{p_1, p_2, p_4\}$. We can see from $\mathbf{X}_r$ in Example 1 (see (20)) that for $S = (1\,1\,0\,1\,0)$ no marking contains $M(S) > 0$.
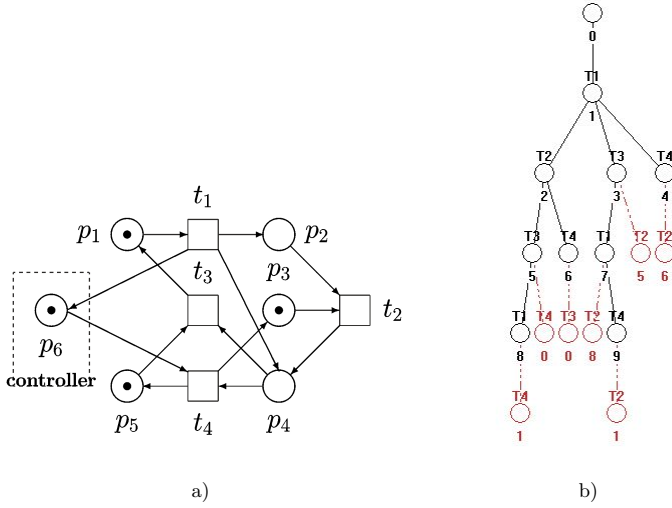
Figure 15. The a) supervised PN by means of siphons and b) its RT

The siphon contains neither the marked trap, because the trap is $T = (1\,0\,0\,1\,1)$. Consequently, the siphon is not controlled. Hence, it is necessary to control it in order to avoid the deadlock. Putting $l \overset{!}{=} S$, or in general $\mathbf{L}^T \overset{!}{=} \mathbf{S}_m$, we have the structure of the supervisor as follows

$$
\mathbf{B}_c = \mathbf{S}_m \cdot \mathbf{B} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} -1 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & -1 & 0 & 1 \\ 1 & 1 & -1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 \end{pmatrix}.
$$

(38)

Because the first row of $\mathbf{B}_c$ is the zero vector, the equation implies that there is only one place p_c, namely p_6 in Figure 15, that will represent the supervisor/controller. Thus, incidence matrices of the supervisor are as follows:

$$
\mathbf{F}_c = \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{G}_c^T = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}.
$$

(39)

The reachable states of the controlled (supervised) PN model are the following

$$
\mathbf{X}_c = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 2 & 1 & 0 & 2 \\ 1 & 1 & 0 & 1 & 2 & 0 & 1 & 1 & 0 & 2 & 2 \\ 0 & 1 & 2 & 0 & 0 & 1 & 1 & 1 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 2 & 0 & 2 & 0 & 0 & 3 & 1 \\ 1 & 2 & 2 & 2 & 1 & 2 & 1 & 3 & 3 & 0 & 2 \end{pmatrix}
$$

(40)

where the columns represent state vectors $\mathbf{x}_0 \ldots \mathbf{x}_{10}$ being nodes of the RT.

The controllability of siphon S is ensured by adding a monitor. Namely, the number of tokens leaving S is limited by a marking invariant law being implemented by a P-invariant whose *support* contains the monitor.

The supervised PN model is given in Figure 15 a). The corresponding RT is shown in Figure 15 b) in order to see that no deadlock occurs in the supervised system.

By the way, the controlled system in Figure 15 a) is the same as that in Figure 6 a) obtained by the prejudged relation $t_3 \succ t_4$.

3.2.2 Example 6

In Subsection 3.1.2 (Example 2) the supervisor eliminating deadlocks was proposed. Consider here the same deadlocked PN model and let us use the siphon-based approach to solve the problem. The PN model contains the following minimal siphons and traps:

$$S_1 = \{p_3, p_4, p_9\},$$
$$Tr_1 = \{p_3, p_4, p_9\},$$
$$S_2 = \{p_2, p_5, p_8\},$$
$$Tr_2 = \{p_2, p_5, p_8\},$$
$$S_3 = \{p_1, p_6, p_7\},$$
$$Tr_3 = \{p_1, p_6, p_7\},$$
$$S_4 = \{p_3, p_5, p_8, p_9\},$$
$$Tr_4 = \{p_2, p_4, p_8, p_9\},$$
$$S_5 = \{p_2, p_6, p_7, p_8\},$$
$$Tr_5 = \{p_1, p_5, p_7, p_8\},$$
$$S_6 = \{p_3, p_6, p_7, p_8, p_9\},$$
$$Tr_6 = \{p_1, p_4, p_7, p_8, p_9\}.$$

or in the matrix form

$$\mathbf{S}_m = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}, \quad \mathbf{T}_m = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}. \quad (41)$$

The first three siphons are equal to the first three traps.

As it can be seen from (41), the places p_2, p_6, p_7, p_8 of the siphon S_5 (the 5th row of $\mathbf{S}_m$) are active in several state vectors of the PN model – see the corresponding rows 2, 6, 7, 8 of this matrix (24).

As we can see in Figure 9, places p_2, p_6, p_7, p_8 included in the siphon S_5, create even the empty siphon (as to marking). All the output transitions of S_5 are $S_5^{\bullet} = \{t_2, t_3, t_6, t_7, t_8\}$. The transition t_1 does not belong in $S_5^{\bullet}$ because it is the *source* transition (generating siphons) and it does not fall with the siphon definition. By the way, t_5 is also the *source* transition. All actual output transitions of $S_5^{\bullet}$ are disabled, since they require at least one token from some place in S_5.

All of input transitions of S are $^{\bullet}S_5 = \{t_2, t_3, t_6, t_7, t_8\}$.

Even, any transition which could bring tokens in S_5 is a part of $^{\bullet}S_5$, and consequently is disabled. Consequently, S_5 will remain empty during entire system dynamics evolution as well as the transitions in $S_5^{\bullet}$ will be dead during this evolution. Alike, we could analyze also other nonzero siphons S_4, S_6. Such a work is too toilsome.

It means that it is necessary to find a way how to deliver tokens into the siphon places. Therefore, let us compute the monitors that will make this for us.

Put $\mathbf{L} \overset{!}{=} \mathbf{S}_m$. Then,

$$\mathbf{B}_c = \mathbf{S}_m.\mathbf{B} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & -1 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & -1 & 1 & 0 \\ -1 & 0 & 1 & 0 & -1 & 0 & 1 & 0 \end{pmatrix}. \tag{42}$$

Excluding the upper zero sub-matrix we obtain the controller with the incidence matrices

$$\mathbf{F}_c = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{G}_c^T = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}. \tag{43}$$

The supervised PN model of RAS and its RT are given in Figure 16 a) and 16 b), respectively. The supervisor consists of the imbedded places (monitors) A, B, C. RT is not necessary in this approach but it is included in order to see that no deadlock occurs in the supervised RAS.

The nodes of RT are the rows of the following matrix where the first row is the initial state vector $\mathbf{x}_0$. No state vector represents a deadlock. The nodes of the RT are the state vectors being rows of the following matrix
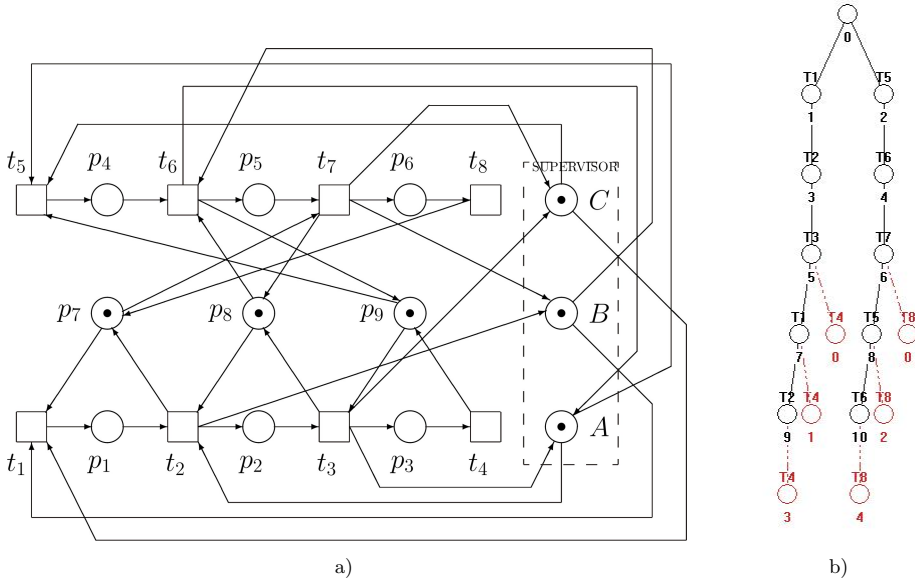
Figure 16. The PN model of the supervised RAS

$$\mathbf{X}_r^T = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}. \tag{44}$$

3.2.3 Local Conclusion

The approach based on siphons has not so exact analytically expressed procedure (especially in case of siphons and traps computation) like the previous approach based on P-invariants and analysis of RT. Firstly, it is necessary to analyze the PN model of RAS and find its siphons and traps. There are Matlab tools for calculation of siphons and traps – see e.g. [12, 13]. After thorough analysis of the set of siphons and the set of traps, the procedure of the supervisor synthesis can start on. The analytical approach to computation of monitors is utilized. Then, the procedure is

also clear and simple as well as in case of the previous approach. The analysis of RT is not necessary in this case. However, it is possible to generate RT in order to be sure that the controlled PN model of RAS is deadlock-free.

3.3 A Short Comparison of Both Approaches

When we compare both approaches applied on the relatively simple cases of dead-locked RAS, we can say the following:

1. the weak point (shortcomming, weakness) of the first approach consists in the computational demands at finding RT/RG and the labor consumption at finding the conditions for elimination deadlocks. In case of largely branched RT computing takes a very long time. The dependance on the initial state is also a weakness;

2. the weak point of the second approach consists in computational demands at finding siphons. This may also takes a very long time.

The first approach yields the supervisor which restricts the development of the system a little less, than the supervisor synthesized by the second approach. On the other hand, there is a perspective of finding new and new methods in the second approach.

However, because from several simple DES and RAS it is impossible to draw serious conclusions, we will do this in Part 2 (may be also in Part 3) of this paper, which is being prepared. There, not only the more complicated examples will be tested by both approaches, and then compared each other, but also newer approaches will be analyzed, especially some of those published in [63, 8, 25, 55, 56, 57, 59, 62, 64, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 92, 93, 94, 91, 88, 89, 65, 66, 67, 68, 69, 58, 60, 61, 47, 48, 49, 50, 51, 52, 53].

4 FURTHER RESEARCH IN THE FUTURE

In this paper, the problem of deadlock avoiding in RAS was presented by means of two different approaches. Relatively simple examples, but one of them being practical, were introduced to illustrate the applicability of these approaches. In the further research, we will continue in solving the problem with deadlocks in RAS, especially of S^3PR and S^4PR kinds, investigated in [63, 8, 25, 55, 56, 57, 59, 62, 64, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 92, 93, 94, 91, 88, 89, 65, 66, 67, 68, 69, 58, 60, 61, 47, 48, 49, 50, 51, 52, 53]. Both approaches (first based on P-invariants and second based on siphons) will be tested on the same more complicated practical examples. The effectiveness of finding their results will be mutually compared and evaluated.

Moreover, a deeper view on admissibility of deadlock-free control of RAS will be performed. Namely, there exist following kinds of PN states (markings) [65]:

1. legal;

2. illegal, or else forbidden;

3. admissible,

i.e., such from which the system cannot uncontrollably reach illegal state. In PN which are controllable only partially, an illegal state may be reachable from a legal one by firing of uncontrollable transitions (see e.g. [3, 4, 5, 6, 65, 66, 67, 68, 69, 53, 71, 85]). Consequently, it is necessary to find more restrictive control policy which will enforce a subset of legal states, i.e. admissible states. From such states the system cannot uncontrollably reach an illegal state. However, this topic will be analyzed later, probably only in Part 3 of this paper (because of the limited space for one paper in this journal).

5 CONCLUSION

PN are a formal modelling tool. They are a popular mathematical formalism to investigate and analyze modelling and control of DES. PN theory has been one of the most interesting topics in computer science. PN find wide application in contemporary technical systems, especially in AMS/FMS where there is also a mathematical framework to investigate the deadlock control problems in a variety of RAS. In such a way PN become the effective tool for the design and management of modern AMS/FMS. In the last years the siphon-based approach has dominated among the methodologies dealing with the deadlock analysis and control of RAS – see the new research especially in [63, 8, 25, 55, 56, 57, 59, 62, 64, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 92, 93, 94, 91, 88, 89, 65, 66, 67, 68, 69, 58, 60, 61, 47, 48, 49, 50, 51, 52, 53]. Although the approach to the same problem based on P-invariants is a little older, it does not lag behind as to the quality of its results (sometimes more to the contrary).

In this paper, both approaches were presented and illustrated by examples. First on simple examples and then on more complicated ones related to real RAS. The total number of introduced examples was 6.

Each approach was evaluated in the particular local conclusion (see Subsection 3.1.6 and Subsection 3.2.3). It was demonstrated that both approaches are suitable for the deadlock elimination very well. The detailed comparison of both approaches on the same more complicated examples will be performed in the planned Part 2 (may be only Part 3) of this paper to be published later.

Acknowledgement

.

REFERENCES

[1] Barkaoui, K.—Couvreur, J. M.—Dutheillet, C.: On the Liveness in Extended Non Self-Controlling Nets. In: De Michelis, G., Diaz, M. (Eds.): Application and Theory of Petri Nets 1995 (ICATPN 1995). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 935, 1995, pp. 25–44, doi: 10.1007/3-540-60029-9_32.

[2] Barkaoui, K.—Pradat-Peyre, J. F.: On Liveness and Controlled Siphons in Petri Nets. In: Billington, J., Reisig, W. (Eds.): Application and Theory of Petri Nets 1996 (ICATPN 1996). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 1091, 1996, pp. 57–72, doi: 10.1007/3-540-61363-3_4.

[3] Čapkovič, F.: Petri Nets at Modelling and Control of Discrete-Event Systems Containing Nondeterminism – Part 1. Computing and Informatics, Vol. 37, 2018, No. 5, pp. 1258–1292, doi: 10.4149/cai_2018_5_1258.

[4] Čapkovič, F.: Petri Nets at Modelling and Control of Discrete-Event Systems Containing Nondeterminism – Part 2. Computing and Informatics, Vol. 38, 2019, No. 3, pp. 728–764, doi: 10.31577/cai_2019_3_728.

[5] Čapkovič, F.: Modeling and Control of Discrete-Event Systems with Partial Non-Determinism Using Petri Nets. Acta Polytechnica Hungarica, Vol. 17, 2020, No. 4, pp. 47–66, doi: 10.12700/APH.17.4.2020.4.3.

[6] Čapkovič, F.: Timed and Hybrid Petri Nets at Solving Problems of Computational Intelligence. Computing and Informatics, Vol. 34, 2015, No. 4, pp. 746–778.

[7] Chao, D. Y.—Pan, Y. L.: Uniform Formulas for Compound Siphons, Complementary Siphons and Characteristic Vectors in Deadlock Prevention of Flexible Manufacturing Systems. Journal of Intelligent Manufacturing, Vol. 26, 2015, No. 1, pp. 13–23, doi: 10.1007/s10845-013-0757-7.

[8] Chao, D. Y.: Max'-Controlled Siphons for Liveness of S^3PGR^2. IET Control Theory and Applications, Vol. 1, 2007, No. 4, pp. 933–936, doi: 10.1049/iet-cta:20060275.

[9] Chen, Y. F.—Li, Z. W.: Design of a Maximally Permissive Liveness-Enforcing Supervisor with a Compressed Supervisory Structure for Flexible Manufacturing Systems. Automatica, Vol. 47, 2011, No. 5, pp. 1028–1034, doi: 10.1016/j.automatica.2011.01.070.

[10] Chen, Y. F.—Li, Z. W.: Optimal Supervisory Control of Automated Manufacturing Systems. 1st Edition. CRC Press, 2012, doi: 10.1201/b14588.

[11] Chu, F.—Xie, X. L.: Deadlock Analysis of Petri Nets Using Siphons and Mathematical Programming. IEEE Transactions on Robotics and Automation, Vol. 13, 1997, No. 6, pp. 793–804, doi: 10.1109/70.650158.

[12] Davidrajuh, R.: GPenSIM, General Purpose Petri Net Simulator for MATLAB Platform. Available at: http://www.davidrajuh.net/gpensim/.

[13] Davidrajuh, R.: General Purpose Petri Net Simulator GPenSIM. Version 9.0. University of Stavanger, Norway, 2014. Available at: http://www.davidrajuh.net/gpensim/v9/GPenSIM_v9_User_Manual.pdf.

[14] Diaz, M. (Ed.): Petri Nets: Fundamental Models, Verification and Applications. John Wiley and Sons, 2009, doi: 10.1002/9780470611647.

[15] DIJKSTRA, E. W.: Cooperating Sequential Processes. Technical Report, Technological University, Eindhoven, Netherlands, 1965.

[16] DIJKSTRA, E. W.: Cooperating Sequential Processes. In: Genuys, F. (Ed.): Programming Languages: NATO Advanced Study Institute. Lectures at the Summer School, Villard-le-Lans, 1966, pp. 43–112, Academic Press Inc., London, 1968.

[17] DESEL, J.—REISIG, W.: Place/Transition Petri Nets. In: Reisig, W., Rozenberg, G. (Eds.): Lectures on Petri Nets I: Basic Models (ACPN 1996). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 1491, 1998, pp. 122–173, doi: 10.1007/3-540-65306-6_15.

[18] EZPELETA, J.—COLOM, J. M.—MARTINEZ, J.: A Petri Net Based Deadlock Prevention Policy for Flexible Manufacturing Systems. IEEE Transaction on Robotics and Automation, Vol. 11, 1995, No. 2, pp. 173–184, doi: 10.1109/70.370500.

[19] FANTI, M. P.—MAIONE, B.—MASCOLO, S.—TURCHIANO, B.: Event-Based Feedback Control for Deadlock Avoidance in Flexible Production Systems. IEEE Transaction on Robotics and Automation, Vol. 13, 1997, No. 3, pp. 347–363, doi: 10.1109/70.585898.

[20] FANTI, M. P.—MAIONE, B.—TURCHIANO, B.: Comparing Digraph and Petri Net Approaches to Deadlock Avoidance in FMS. IEEE Transaction on Systems, Man, and Cybernetics, Part B (Cybernetics), Vol. 30, 2000, No. 5, pp. 783–798, doi: 10.1109/3477.875452.

[21] FAROOQ, A.—HUANG, H.—WANG, X. L.: Petri Net Modeling and Deadlock Analysis of Parallel Manufacturing Processes with Shared-Resources. Journal of Systems and Software, Vol. 83, 2010, No. 4, pp. 675–688, doi: 10.1016/j.jss.2009.11.705.

[22] GUAN, X.—LI, Y.—XU, J.—WANG, C.—WANG, S.: A Literature Review of Deadlock Prevention Policy Based on Petri Nets for Automated Manufacturing Systems. International Journal of Digital Content Technology and Its Applications (JDCTA), Vol. 6, 2012, No. 21, pp. 426–433, doi: 10.4156/jdcta.vol6.issue21.48.

[23] HERNÁNDEZ-FLORES, E.—LÓPEZ-MELLADO, E.—RAMÍREZ-TREVIÑO, A.: Diagnosability Analysis of Partially Observable Deadlock-Free Petri Nets. Proceedings of the 3rd International Workshop on Dependable Control of Discrete Systems (DCDS '11), Saarbrücken, Germany, 2011, pp. 174–179, doi: 10.1109/dcds.2011.5970337.

[24] HOU, Y. F.—BARKAOUI, K.: Deadlock Analysis and Control Based on Petri Nets: A Siphon Approach Review. Advances in Mechanical Engineering, Vol. 9, 2017, No. 5, pp. 1–30, doi: 10.1177/1687814017693542.

[25] HOU, Y. F.—ZHAO, M.—LIU, D.—HONG, L.: An Efficient Siphon-Based Deadlock Prevention Policy for a Class of Generalized Petri Nets. Discrete Dynamics in Nature and Society, Vol. 2016, Art. No. 8219424, 12 pp., doi: 10.1155/2016/8219424.

[26] HU, W. S.—ZHU Y. Y.—LEI, J.: The Detection and Prevention of Deadlock in Petri Nets. Physics Procedia, Vol. 22, 2011, pp. 656–659, doi: 10.1016/j.phpro.2011.11.102.

[27] HU, H.—LIU, Y.—YUAN, L.: Supervisor Simplification in FMSs: Comparative Studies and New Results Using Petri Nets. IEEE Transactions on Control Systems Technology, Vol. 24, 2016, No. 1, pp. 81–95, doi: 10.1109/TCST.2015.2420619.

[28] HU, H. S.—ZHOU, M. C.—LI, Z. W.: Liveness Enforcing Supervision of Video Streaming Systems Using Non-Sequential Petri Nets. IEEE Transactions on Multimedia, Vol. 11, 2009, No. 8, pp. 1457–1465, doi: 10.1109/TMM.2009.2032678.

[29] HU, H. S.—ZHOU, M. C.—LI, Z. W.: Algebraic Synthesis of Timed Supervisor for Automated Manufacturing Systems Using Petri Nets. IEEE Transactions on Automation Science and Engineering, Vol. 7, 2010, No. 3, pp. 549–557, doi: 10.1109/TASE.2009.2037825.

[30] HU, H. S.—ZHOU, M. C.—LI, Z. W.: Low-Cost and High-Performance Supervision in Ratio-Enforced Automated Manufacturing Systems Using Timed Petri Nets. IEEE Transactions on Automation Science and Engineering, Vol. 7, 2010, No. 4, pp. 933–944, doi: 10.1109/TASE.2010.2046412.

[31] HU, H. S.—ZHOU, M. C.—LI, Z. W.: Supervisor Design to Enforce Production Ratio and Absence of Deadlock in Automated Manufacturing Systems. IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, Vol. 41, 2011, No. 2, pp. 201–212, doi: 10.1109/TSMCA.2010.2058101.

[32] HU, H. S.—ZHOU, M. C.—LI, Z. W.: Supervisor Optimization for Deadlock Resolution in Automated Manufacturing Systems with Petri Nets. IEEE Transactions on Automation Science and Engineering, Vol. 8, 2011, No. 4, pp. 794–804, doi: 10.1109/TASE.2011.2156783.

[33] HU, H. S.—ZHOU, M. C.—LI, Z. W.: Liveness and Ratio-Enforcing Supervision of Automated Manufacturing Systems Using Petri Nets. IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, Vol. 42, 2012, No. 2, pp. 392–403, doi: 10.1109/TSMCA.2011.2162502.

[34] HU, H. S.—ZHOU, M. C.—LI, Z. W.—TANG, Y.: An Optimization Approach to Improved Petri Net Controller Design for Automated Manufacturing Systems. IEEE Transactions on Automation Science and Engineering, Vol. 10, 2013, No. 3, pp. 772–782, doi: 10.1109/TASE.2012.2201714.

[35] HU, H. S.—LIU, Y.—ZHOU, M. C.: Maximally Permissive Distributed Control of Large Scale Automated Manufacturing Systems Modeled with Petri Nets. IEEE Transactions on Control Systems Technology, Vol. 23, 2015, No. 5, pp. 2026–2034, doi: 10.1109/TCST.2015.2391014.

[36] HU, H. S.—SU, R.—ZHOU, M. C.—LIU, Y.: Polynomially Complex Synthesis of Distributed Supervisors for Large-scale AMSs Using Petri Nets. IEEE Transactions on Control Systems Technology, Vol. 24, 2016, No. 5, pp. 1610–1622, doi: 10.1109/TCST.2015.2504046.

[37] HU, H. S.—LIU, Y.: Supervisor Simplification for AMS Based on Petri Nets and Inequality Analysis. IEEE Transactions on Automation Science and Engineering, Vol. 11, 2014, No. 1, pp. 66–77, doi: 10.1109/TASE.2013.2288645.

[38] HU, H. S.—LIU, Y.: Supervisor Synthesis and Performance Improvement for Automated Manufacturing Systems by Using Petri Nets. IEEE Transactions on Industrial Informatics, Vol. 11, 2015, No. 2, pp. 450–458, doi: 10.1109/TII.2015.2402619.

[39] Hu, H. S.—Zhou, M. C.: A Petri Net-Based Discrete-Event Control of Automated Manufacturing Systems with Assembly Operations. IEEE Transactions on Control Systems Technology, Vol. 23, 2015, No. 2, pp. 513–524, doi: 10.1109/TCST.2014.2342664.

[40] Hu, H. S.—Zhou, M. C.—Li, Z. W.—Tang, Y.: Deadlock-Free Control of Automated Manufacturing Systems with Flexible Routes and Assembly Operations Using Petri Nets. IEEE Transactions on Industrial Informatics, Vol. 9, 2013, No. 1, pp. 109–121, doi: 10.1109/TII.2012.2198661.

[41] Huang, Y. S.—Jeng, M. D.—Xie, X. L.—Chung, D. H.: Siphon-Based Deadlock Prevention Policy for Flexible Manufacturing Systems. IEEE Transactions on Systems, Man, and Cybernetics – Part A: System and Humans, Vol. 36, 2006, No. 6, pp. 1248–1256, doi: 10.1109/TSMCA.2006.878953.

[42] Iordache, M. V.—Antsaklis, P. J.: Supervision Based on Place Invariants: A Survey. Discrete Event Dynamic Systems, Vol. 16, 2006, No. 4, pp. 451–492, doi: 10.1007/s10626-006-0021-9.

[43] Iordache, M. V.—Antsaklis, P. J.: Supervisory Control of Concurrent Systems: A Petri Net Structural Approach. Birkhäuser, 2006, doi: 10.1007/0-8176-4488-1.

[44] Karatkevich, A.: Dynamic Analysis of Petri Net-Based Discrete Systems. Springer, Heidelberg, Lecture Notes in Control and Information Sciences, Vol. 356, 2007, doi: 10.1007/978-3-540-71560-3.

[45] Lautenbach, K.: Linear Algebraic Calculation of Deadlocks and Traps. In: Voss, K., Genrich, H. J., Rozenberg, G. (Eds.): Concurrency and Nets. Springer, Berlin, Heidelberg, 1987, pp. 315–336, doi: 10.1007/978-3-642-72822-8_21.

[46] Lawley, M. A.—Reveliotis, S. A.: Deadlock Avoidance for Sequential Resource Allocation Systems: Hard and Easy Cases. International Journal of Flexible Manufacturing Systems, Vol. 13, 2001, No. 4, pp. 385–404, doi: 10.1023/A:1012203214611.

[47] Li, Z. W.—Zhou, M. C.: Elementary Siphons of Petri Nets and Their Application to Deadlock Prevention in Flexible Manufacturing Systems. IEEE Transactions on Systems, Man, and Cybernetics, Part A: System and Humans, Vol. 34, 2004, No. 1, pp. 38–51, doi: 10.1109/TSMCA.2003.820576.

[48] Li, Z. W.—Zhou, M. C.: Control of Elementary and Dependent Siphons in Petri Nets and Their Application. IEEE Transactions on Systems, Man, Cybernetics, Part A, System, Humans, Vol. 38, 2008, No. 1, pp. 133–148, doi: 10.1109/TSMCA.2007.909548.

[49] Li, Z. W.—Zhou, M. C.: Elementary Siphons of Petri Nets for Efficient Deadlock Control. In: Zhou, M. C., Fanti, M. P. (Eds.): Deadlock Resolution in Computer-Integrated Systems. CRC Press, 2005, pp. 309–348, doi: 10.1201/9781315214665.

[50] Li, Z. W.—Wei, N.: Deadlock Control of Flexible Manufacturing Systems via Invariant-Controlled Elementary Siphons of Petri Nets. The International Journal of Advanced Manufacturing Technology, Vol. 33, 2007, pp. 24–35, doi: 10.1007/s00170-006-0452-3.

[51] Li, Z. W.—Zhou, M. C.: Deadlock Resolution in Automated Manufacturing Systems: A Novel Petri Net Approach. Springer, London, Advances in Industrial Control Series, 2009, doi: 10.1007/978-1-84882-244-3.

[52] Li, Z. W.—Wu, N. Q.—Zhou, M. C.: Deadlock Control of Automated Manufacturing Systems Based on Petri Nets – A Literature Review. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), Vol. 42, 2012, No. 4, pp. 437–462, doi: 10.1109/TSMCC.2011.2160626.

[53] Li, X. Y.—Liu, G. Y.—Li, Z. W.—Wu, N. Q.—Barkaoui, K.: Elementary Siphon-Based Robust Control for Automated Manufacturing Systems with Multiple Unreliable Resources. IEEE Access, Vol. 7, 2019, pp. 21006–21019, doi: 10.1109/access.2019.2897753.

[54] Liu, G. J.—Jiang, C. J.: Incidence Matrix Based Methods for Computing Repetitive Vectors and Siphons of Petri Net. Journal of Information Science and Engineering, Vol. 25, 2009, No. 1, pp. 121–136.

[55] Liu, G. Y.—Li, Z.: General Mixed Integer Programming-Based Liveness Test for System of Sequential Systems with Shared Resources Nets. IET Control Theory and Applications, Vol. 4, 2010, No. 12, pp. 2867–2878, doi: 10.1049/iet-cta.2009.0557.

[56] Liu, G.—Li, Z.—Zhong, C.: New Controllability Condition for Siphons in a Class of Generalised Petri Nets. IET Control Theory and Applications, Vol. 4, 2010, No. 5, pp. 854–864, doi: 10.1049/iet-cta.2009.0264.

[57] Liu, G.—Li, Z.—Zhong, C.: Correction to 'New Controllability Condition for Siphons in a Class of Generalised Petri Nets'. IET Control Theory and Applications, Vol. 7, 2013, No. 4, pp. 632–633, doi: 10.1049/iet-cta.2012.0357.

[58] Liu, G. Y.—Barkaoui, K.: A Survey of Siphons in Petri Nets. Information Sciences, Vol. 363, 2016, pp. 198–220, doi: 10.1016/j.ins.2015.08.037.

[59] Liu, G. Y.—Barkaoui, K.: Necessary and Sufficient Liveness Condition of GS^3PR Petri Nets. International Journal of Systems Science, Vol. 46, 2015, No. 7, pp. 1147–1160, doi: 10.1080/00207721.2013.827257.

[60] Liu, G. Y—Li, Z. W.—Al-Ahmari, A. M.: Liveness Analysis of Petri Nets Using Siphons and Mathematical Programming. IFAC Proceedings Volumes, Vol. 47, 2014, No. 2, pp. 383–387, doi: 10.3182/20140514-3-FR-4046.00078.

[61] Liu, D.—Barkaoui, K.—Zhou, M. C.: On Intrinsically Live Structure of a Class of Generalized Petri Nets Modeling FMS. IFAC Proceedings Volumes, Vol. 45, 2012, No. 29, pp. 187–192, doi: 10.3182/20121003-3-MX-4033.00032.

[62] Liu, G. Y.—Li, Z. W.—Barkaoui, K.—Al-Ahmari, A. M.: Robustness of Deadlock Control for a Class of Petri Nets with Unreliable Resources. Information Sciences, Vol. 235, 2013, pp. 259–279, doi: 10.1016/j.ins.2013.01.003.

[63] Liu, G. Y.—Barkaoui, K.: A Survey of Siphons in Petri Nets. Information Sciences, Vol. 363, 2016, pp. 198–220, doi: 10.1016/j.ins.2015.08.037.

[64] Liu, G.—Li, P.—Li, Z.—Wu, N.: Robust Deadlock Control for Automated Manufacturing Systems with Unreliable Resources Based on Petri Net Reachability Graphs. IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 49, 2019, No. 7, pp. 1371–1385, doi: 10.1109/TSMC.2018.2815618.

[65] Ma, Z. Y.—Li, Z. W.—Giua, A.: Computation of Admissible Marking Sets in Weighted State Machines by Dynamic Programming. Proceedings of the 2017 IEEE 56th Annual Conference on Decision and Control (CDC), Melbourne, VIC, Australia, 2017, pp. 4847–4852, doi: 10.1109/CDC.2017.8264375.

[66] MA, Z. Y.—LI, Z. W.—GIUA, A.: Design of Optimal Petri Net Controllers for Disjunctive Generalized Mutual Exclusion Constraints. IEEE Transactions on Automatic Control, Vol. 60, 2015, No. 7, pp. 1774–1785, doi: 10.1109/TAC.2015.2389313.

[67] MA, Z. Y.—LI, Z. W.—GIUA, A.: A Constraint Transformation Technique for Petri Nets with Certain Uncontrollable Structures. IFAC Proceedings Volumes, Vol. 47, 2014, No. 2, pp. 66–72, doi: 10.3182/20140514-3-fr-4046.00085.

[68] MA, Z. Y.—LI, Z. W.—GIUA, A.: Petri Net Controllers for Disjunctive Generalized Mutual Exclusion Constraints. Proceedings of the 2013 IEEE 18th Conference on Emerging Technologies and Factory Automation (ETFA), Cagliari, Italy, 2013, pp. 1–8, doi: 10.1109/ETFA.2013.6648003.

[69] MA, Z. Y.—LI, Z. W.—GIUA, A.: Petri Net Controllers for Generalized Mutual Exclusion Constraints with Floor Operators. Automatica, Vol. 74, 2016, pp. 238–246, doi: 10.1016/j.automatica.2016.07.042.

[70] MINOUX, M.—BARKAOUI, K.: Deadlocks and Traps in Petri Nets as Horn-Satisfiability Solutions and Some Related Polynomially Solvable Problems. Discrete Applied Mathematics, Vol. 29, 1990, No. 2-3, pp. 195–210, doi: 10.1016/0166-218X(90)90144-2.

[71] MOODY, J. O.—ANTSAKLIS, P. J.: Petri Net Supervisors for DES with Uncontrollable and Unobservable Transitions. IEEE Transactions on Automatic Control, Vol. 45, 2000, No. 3, pp. 462–476, doi: 10.1109/9.847725.

[72] MURATA, T.: Petri Nets: Properties, Analysis and Applications. Proceedings of the IEEE, Vol. 77, 1989, No. 4, pp. 541–580, doi: 10.1109/5.24143.

[73] NAKAMOTO, F. Y.—MIYAGI, P. E.—dos SANTOS FILHO, D. J.: Automatic Generation of Control Solution for Resource Allocation Using Petri Net Model. Produçaõ (International Journal of Advanced Manufacturing Technology), Vol. 19, 2009, No. 1, pp. 8–26, doi: 10.1590/S0103-65132009000100002.

[74] NAKAMOTO, F. Y.—MIYAGI, P. E.—dos SANTOS FILHO, D. J.: Resources Allocation Control in Flexible Manufacturing Systems Using the Deadlock Avoidance Method. ABCM (Brazilian Society of Mechanical Science and Engineering), ABCM Symposium Series in Mechatronics, Vol. 3, 2008, pp. 454–460.

[75] PARK, J.—REVELIOTIS, S. A.: Deadlock Avoidance in Sequential Resource Allocation Systems with Multiple Resource Acquisitions and Flexible Routings. IEEE Transactions on Automatic Control, Vol. 46, 2001, No. 10, 2001, pp. 1572–1583, doi: 10.1109/9.956052.

[76] REISIG, W.: Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies. Springer, 2013, doi: 10.1007/978-3-642-33278-4.

[77] REISIG, W.: Petri Nets. Springer, Berlin, Heidelberg, EATCS Monographs on Theoretical Computer Science, Vol. 4, 1985, doi: 10.1007/978-3-642-69968-9.

[78] REVELIOTIS, S. A.—FERREIRA, P. M.: Deadlock Avoidance Policies for Automated Manufacturing Cells. IEEE Transactions on Robotics and Automation, Vol. 12, 1996, No. 6, pp. 845–857, doi: 10.1109/70.544768.

[79] REVELIOTIS, S. A.—LAWLEY, M. A.—FERREIRA, P. M.: Polynomial-Complexity Deadlock Avoidance Policies for Sequential Resource Allocation Systems. IEEE

Transaction on Automatic Control, Vol. 42, 1997, No. 10, pp. 1344–1357, doi: 10.1109/9.633824.

[80] REVELIOTIS, S. A.: Logical Control of Complex Resource Allocation Systems. Foundations and Trends in Systems and Control, Vol. 4, 2017, No. 1-2, pp. 1–223, doi: 10.1561/2600000010.

[81] REVELIOTIS, S. A.: Coordinating Autonomy: Sequential Resource Allocation Systems for Automation. IEEE Robotics and Automation Magazine, Vol. 22, 2015, No. 2, pp. 77–94, doi: 10.1109/MRA.2015.2401295.

[82] REVELIOTIS, S. A.: On the Siphon-Based Characterization of Liveness in Sequential Resource Allocation Systems. In: van der Aalst, W. M. P., Best, E. (Eds.): Applications and Theory of Petri Nets 2003 (ICATPN 2003). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 2679, 2003, pp. 241–255, doi: 10.1007/3-540-44919-1_17.

[83] ROW, T. C.—SYU, W. M.—PAN, Y. L.—WANG, C. C.: One Novel and Optimal Deadlock Recovery Policy for Flexible Manufacturing Systems Using Iterative Control Transitions Strategy. Mathematical Problems in Engineering, Vol. 2019, Art. No. 4847072, 12 pp., doi: 10.1155/2019/4847072.

[84] SCHMIDT, K.: Verification of Siphons and Traps for Algebraic Petri Nets. In: Azéma, P., Balbo, G. (Eds.): Application and Theory of Petri Nets 1997 (ICATPN 1997). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 1248, 1997, pp. 427–446, doi: 10.1007/3-540-63139-9_49.

[85] TAOKA, S.—FURUSATO, S.—WATANABE, T.: A Heuristic Algorithm FSDC Based on Avoidance of Deadlock Components in Finding Legal Firing Sequences of Petri Nets. In: van der Aalst, W. M. P., Best, E. (Eds.): Applications and Theory of Petri Nets 2003 (ICATPN 2003). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 2679, 2003, pp. 417–439, doi: 10.1007/3-540-44919-1_26.

[86] TRICAS, F.—EZPELETA, J.: Computing Minimal Siphons in Petri Net Models of Resource Allocation Systems: A Parallel Solution. IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, Vol. 36, 2006, No. 3, pp. 532–539, doi: 10.1109/TSMCA.2005.855751.

[87] TRICAS, F.: Deadlock Analysis, Prevention and Avoidance in Sequential Resource Allocation Systems. Ph.D. Thesis, University of Zaragoza, Zaragoza, Spain, 2003. Available at: https://www.researchgate.net/publication/33419595.

[88] WANG, Y.—LAFORTUNE, S.—KELLY, T.—KUDLUR, M.—MAHLKE, S.: The Theory of Deadlock Avoidance via Discrete Control. ACM SIGPLAN Notices, Vol. 44, 2009, No. 1, pp. 252–263, doi: 10.1145/1594834.1480913.

[89] WANG, Y.—KELLY, T.—KUDLUR, M.—MAHLKE, S.—LAFORTUNE, S.: The Application of Supervisory Control to Deadlock Avoidance in Concurrent Software. Proceedings of 9th IEEE/IFAC International Workshop on Discrete Event Systems (WODES '08), Göteborg, Sweden, 2008, pp. 287–292, doi: 10.1109/WODES.2008.4605961.

[90] WEGRZYN, A.—KARATKEVICH, A.—BIEGANOWSKI, J.: Detection of Deadlocks and Traps in Petri Nets by Means of Thelen's Prime Implicant Method. International Journal of Applied Mathematics and Computer Science, Vol. 14, 2004, No. 1, pp. 113–121.

[91] WU, W. H.—CHAO, D. Y.: Controllability of Weakly Dependent Siphons under Elementary-Siphon Control. Transactions of the Institute of Measurement and Control, Vol. 38, 2016, No. 8, pp. 941–955, doi: 10.1177/0142331214568606.

[92] YAN, M. M.— ZHU, R. M.—LI, Z. W.—WANG, A.—ZHOU, M. C.: A Siphon-Based Deadlock Prevention Policy for a Class of Petri Nets – S^{3}PMR. Proceedings of the 17$^{\text{th}}$ World Congress of the International Federation of Automatic Control (IFAC), Seoul, Korea, 2008, Vol. 6, 2008, pp. 3352–3357. Available at: `http://toc.proceedings.com/04672webtoc.pdf`. ISBN: 978-1605607580.

[93] YAN, M. M.—LI, Z. W.—WEI, N.—ZHAO, M.: A Deadlock Prevention Policy for a Class of Petri Nets S^{3}PMR. Journal of Information Science and Engineering, Vol. 25, 2009, No. 1, pp. 167–183.

[94] YUE, H.—XING, K. Y.—HU, H. S.—WU, W. M.—SU, H. Y.: Petri-Net-Based Robust Supervisory Control of Automated Manufacturing Systems. Control Engineering Practice, Vol. 54, 2016, pp. 176–189, doi: 10.1016/j.conengprac.2016.05.009.

František ČAPKOVIČ received his Master's degree in 1972 from the Faculty of Electrical Engineering of the Slovak Technical University, Bratislava, Slovakia. Since 1972 he has been working with the Slovak Academy of Sciences (SAS), Bratislava, in 1972–1991 at the Institute of Technical Cybernetics, in 1991–2001 at the Institute of Control Theory and Robotics and in 2001 till now at the Institute of Informatics. In 1980 he received his Ph.D. from SAS, since 1998 serving as Associate Professor. He works in the area of modelling, analysing and intelligent control of Discrete-Event Systems (DES) and Hybrid Systems. He is the author of more than 240 publications.