Computing and Informatics, Vol. 38, 2019, 728–764, doi: 10.31577/cai_2019_3_728

PETRI NETS AT MODELLING AND CONTROL OF DISCRETE-EVENT SYSTEMS WITH NONDETERMINISM – PART 2

František Čapkovič

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

Abstract. Discrete-Event Systems (DES) are discrete in nature. Petri Nets (PN) are one of the most widespread tools for DES modelling, analyzing and control. Different kinds of PN can be used for such purposes. Some of them were described in [3], being the first part of this paper. Here, the applicability of Labelled PN (LbPN) and Interpreted PN (IPN) for modelling and control of nondeterministic DES, especially with uncontrollable and/or unobservable transitions in the models, will be pointed out. Moreover, another kinds of nondeterminism in DES (errors, failures) will be modelled, and the possibilities of the error recovery of failed system will be presented.

Keywords: Analyzing, control synthesis, discrete-event systems, error recovery, interpreted Petri nets, modelling, labelled Petri nets, place/transition Petri nets, uncertainty, uncontrollable transitions, unmeasurable places, unobservable transitions

Mathematics Subject Classification 2010: 93-C65, 93-C30

1 INTRODUCTION AND PRELIMINARIES

This paper is the Part 2 of the the paper which had started by the Part 1 published as [3]. In the Part 1 the basic background about several crucial kinds of Petri nets

(PN), including Place/Transition PN (P/T PN), Timed PN (TPN), Controlled PN (CtPN), Interpreted PN (IPN) and Labelled PN (LbPN), was presented as well as a simple application of them for modelling and control of flexible manufacturing systems (FMS). In this paper, especially the IPN and LbPN will be used in modelling and control of discrete-event systems (DES), namely some kinds of FMS and a segment of transport systems, to tend towards the applications in practice. Two principled kinds of uncertainties in PN models of DES will be analyzed here:

1. the effect of uncontrollable and unobservable transitions;
2. the occurrence of different kinds of errors/failures.

While resolving the former problem requires the usage of special kinds of PNs (IPN or LbPN), resolving the latter one requires the usage of an error recovery procedure. Both kinds of uncertainties as well as their combination will be analyzed here by PN-based approach.

1.1 Mathematical Model of Petri Nets

At the beginning let us recall the principle definition of the basal PN – P/T PN. As it was introduced in the Part 1 of this paper – see [3] – mathematical expression of P/T PN consists of the (i) expression of the PN structure – PN is a bipartite directed graph $\langle P, T, F, G\rangle$ with $P = \{p_1, \ldots, p_n\}$, $|P| = n$ being the set of places p_i, $i = 1, \ldots, n$; $T = \{t_1, \ldots, t_m,\ |T| = m\}$ being the set of transitions t_j, $j = 1, \ldots, m$; F being the set of directed arcs from places to transitions; G being the set of directed arcs from transitions to places; (ii) description of PN dynamics (the marking development) in the form of the limited discrete linear equation

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

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

with the structural matrix $\mathbf{B} = \mathbf{G}^T - \mathbf{F}$ ($\mathbf{F}$, $\mathbf{G}^T$ correspond to F, G). Here, $\mathbf{x}_k = (\sigma^k_{p_1}, \ldots, \sigma^k_{p_n})^T$ with entries $\sigma^k_{p_i} \in \{0, 1, \ldots, \infty\}$, representing the states of particular places, is the PN state vector in the k^{th} step of the dynamics development; $\mathbf{u}_k = (\gamma^k_{t_1}, \ldots, \gamma^k_{t_m})^T$ with entries $\gamma^k_{p_i} \in \{0, 1\}$, representing the states of particular transitions (either enable – when 1, or disable – when 0) is the control vector; $\mathbf{F}$, $\mathbf{G}^T$ are, respectively, incidence matrices of arcs from places to transitions and contrariwise.

Particulars about P/T PN as well as about other kinds of Petri nets which will be used here (IPN, LbPN) are introduced and explained in detail in the Part 1 of the paper – i.e. in [3].

1.2 Place Invariants in Supervisor Synthesis

In order to control the PN model of a plant, a supervisor has to be proposed. To synthesize the supervisor based on place invariants (P-invariants), where the

invariants are defined as columns of the matrix $\mathbf{W}$ given as follows:

$$\mathbf{W}^T.\mathbf{B} = \mathbf{0}, \tag{3}$$

it is necessary to enforce the suitable restrictive condition on the state vectors of the system (1) as follows:

$$\mathbf{L}.\mathbf{x} \leq \mathbf{b}. \tag{4}$$

Here, $\mathbf{L}$ is a matrix of integers and $\mathbf{b}$ is a vector of integers representing some restrictions on the linear combination of corresponding entries of the state vector. To eliminate the inequality in (4) it is needful to insert there the $(n_s \times 1)$ vector $\mathbf{x}_s$ consisting of *slack* variables. Thus,

$$\mathbf{L}.\mathbf{x} + \mathbf{x}_s = \mathbf{L}.\mathbf{x} + \mathbf{I}_s.\mathbf{x}_s = (\mathbf{L}\mathbf{I}_s).\begin{pmatrix} \mathbf{x} \\ \mathbf{x}_s \end{pmatrix} = \mathbf{b} \tag{5}$$

where $\mathbf{I}_s$ is the $(s \times s)$-dimensional identity matrix. To synthesize the supervisor with a structure $\mathbf{B}_s$ (so far unknown), force the matrix $(\mathbf{L}, \mathbf{I}_s)$ into (3) instead of $\mathbf{W}^T$ and the matrix $(\mathbf{B}^T, \mathbf{B}_s^T)^T$ instead of $\mathbf{B}$. Hence, the structure of the supervisor was found as

$$(\mathbf{L}\mathbf{I}_s).\begin{pmatrix} \mathbf{B} \\ \mathbf{B}_s \end{pmatrix} = \mathbf{0}; \quad \mathbf{B}_s = -\mathbf{L}.\mathbf{B}; \quad \mathbf{B}_s = \mathbf{G}_s^T - \mathbf{F}_s. \tag{6}$$

Here, $\mathbf{B}_s$ represents the interconnections of n_s additional places (called *monitors*) with the original PN. $\mathbf{F}_s, \mathbf{G}_s$ obtained by the decomposition of $\mathbf{B}_s$ are the incidence matrices of the directed arcs. The monitors together with the arcs create the supervisor. The directed arcs realize interconnections of the supervisor with the PN model of a plant in both directions:

1. from the supervisor places to a part of the PN model transitions (a set of the model inputs) and
2. from another part of the PN model transitions (a set of the model outputs) to the places of the supervisor.

The initial state of the supervisor follows from (5) in the form

$$\mathbf{x}_s^0 = \mathbf{b} - \mathbf{L}.\mathbf{x}_0. \tag{7}$$

1.3 Uncontrollable and Unobservable Transitions

When no uncontrollable and unobservable transitions occur in the PN model (it is the ideal case), the supervisor is able to observe all transitions. Consequently, it can either prevent a transition from firing or to fire it – i.e. to enforce a desired behavior on it. However, in practice usually such an assumption is not valid. There are two possibilities for transitions in PN models of realistic systems: (i) a transition $t_j \in T$

may be uncontrollable; (ii) a transition $t_j \in T$ may be unobservable. In the former case the supervisor is not able to prevent the transition from firing – i.e. there is no arc from any supervisor place to t_j. In the latter case the supervisor is not able to detect the firing of the transition – i.e. there is no arc from any transition of the PN model of the plant to the supervisor place. In other words, the feedback from the PN model of the plant to the supervisor is missing. It means that there are neither arcs from an unobservable transition t_j to any controller place p_i, $i = 1, \ldots s$, nor the arc from any controller place p_i, $i = 1, \ldots s$ to t_j.

It appears that a transition being unobservable is also uncontrollable. Therefore, controllability of a transition implies its observability. Consequently, three different kinds of transitions are distinguished:

1. controllable;
2. uncontrollable, but observable; and
3. uncontrollable and unobservable.

Accordingly, the set T of transitions has three following subsets $T = T_{o,c} \cup T_{o,uc} \cup T_{uo,uc}$. Hence, it can be written that $T_{uc} = T_{o,uc} \cup T_{uo,uc}$. Here, $T_{o,c}$ and T_{uc} are, respectively, the subsets of controllable and uncontrollable transitions. $T_{o,uc}$ represents the set of uncontrollable but observable transitions, and $T_{uo,uc}$ consists of transitions that are both uncontrollable and unobservable.

1.4 Errors and Error Recovery

Many times errors of different kinds occur during DES operation. They bring another kind of nondeterminism into DES performance. For instance, in a specific kind of FMS like robotic cells a part may drop out (i.e. fall down) from the robot gripper, in a simple railroad crossing the control system of crossing gate may fail (e.g. the premature gate raising), etc. After the occurrence of the fault the system development is different than the standard one. In such a case the system has to detect what was wrong and recover the normal behaviour, i.e. to eliminate the influence of the error on the system behaviour. Namely, the error recovery is the set of actions that must be performed in order to return the system to its normal state. To do this, it is necessary

1. to synthesize the recovery sequence, and
2. to extend the scope of the controller activity in order to deal with the fault.

The problem is directly connected with reachability.

It is necessary to emphasize that errors of the mentioned kinds cannot be excluded, either in deterministic PN models or in PN models with uncontrollable/unobservable transitions.

1.5 The Paper Organization

The paper is organized as follows:

1. the discussion about the nondeterminism caused by unobservable/uncontrollable transitions and unmeasurable (unobservable) places in PN models of DES and about how to deal with it;
2. the discussion about how to model errors/failures and the dealing with the matter of the error recovery in DES;
3. the introduction of three case studies, the first one based on the IPN model, the second one on the LbPN model and the third one (specific one) based on the PN model of RAS (resource allocation systems) where a special kind of nondeterminism occurs.

2 DEALING WITH UNCONTROLLABLE AND UNOBSERVABLE TRANSITIONS

In case when solely controllable and observable transitions occur, the PN model (1) may be used, and the supervisor may be synthesized by means of (6). In the opposite case, i.e. when uncontrollable and/or unobservable transitions occur, the situation is much more complicated.

2.1 Presence of Uncontrollable and/or Unobservable Transitions

Taking into account the previous relations concerning the PN model, the supervisor synthesis, and the indexing the particular part of transitions described in the Section 1.3 we have to cogitate about how to express the PN model and the supervisor synthesis in this case. We can assume that the incidence matrix $\mathbf{B}$ of the PN model of plant has the form

$$\mathbf{B} = [\mathbf{B}_{o,c}\mathbf{B}_{uc}] = [\mathbf{B}_{o,c}\mathbf{B}_{o,uc}\mathbf{B}_{uo,uc}] \tag{8}$$

where $\mathbf{B}_{uc} = [\mathbf{B}_{o,uc}\mathbf{B}_{uo,uc}]$. Such a configuration of submatrices of $\mathbf{B}$ corresponds to ordering the transitions:

1. $t_1, \ldots, t_{mc}$ (for controllable and observable transitions);
2. $t_{mc+1}, \ldots, t_{mc+mo}$ (for uncontrollable, but observable, transitions);
3. and $t_{mc+mo+1}, \ldots, t_m$ (for uncontrollable and unobservable transitions).

2.1.1 Ideal Enforceability

The ideal enforceability of control interferences occurs if there are no arcs from controller places to transitions $t \in T_{uc}$ and no arcs from transitions $t \in T_{uo,uc}$ to

controller places. It can be easily checked. The controller incidence matrix:

$$\mathbf{B}_s = -\mathbf{LB} = -\mathbf{L}\left[\mathbf{B}_{o,c}\mathbf{B}_{uc}\right] = -\mathbf{L}\left[\mathbf{B}_{o,c}\mathbf{B}_{o,uc}\mathbf{B}_{uo,uc}\right]. \tag{9}$$

In such a case the following inequality has to be valid:

$$-\mathbf{LB}_{o,uc} \geq \mathbf{0} \tag{10}$$

where the signs of the inequality are performed element by element. This relation expresses the fact that the firing of any uncontrollable, but observable, transition does not depend on the number of tokens in a controller place, but may increase this number. Moreover, the following relation has to be valid:

$$\mathbf{LB}_{uo,uc} = \mathbf{0}. \tag{11}$$

This equation expresses the fact that the firing of any uncontrollable and unobservable transition will not affect the number of tokens in a controller place. Finally, for the initial state of the supervisor, the following relation has to hold:

$$\mathbf{x_0^s} = \mathbf{b} - \mathbf{Lx}_0 \geq \mathbf{0}. \tag{12}$$

This inequality expresses that the initial state vector (i.e. initial marking) $\mathbf{x}_0^s$ of the supervisor is a nonzero vector. Note, that it is the same relation as $\mathbf{Lx}_0 \leq \mathbf{b}$.

Supervisory control is a procedure of enforcing the external constraints on a system to be controlled. If a desired control specification is ideally enforceable, the supervisor respects the observability and controllability constraints. When the supervisor respects the uncontrollability and unobservability constraints of the plant, it is marked as admissible. Consequently, the presence of uncontrollable and/or unobservable transitions does not pose any problem. However, in real conditions the enforceability may not exist. Even, the ideal enforceability does not exist.

The specification (4) or (5) is said to be ideally enforceable, if the (ideal) supervisor/controller represented by $\mathbf{B}_s$ and $\mathbf{x}_0^s$ can be realized (i.e., if it is feasible) – i.e., if in case of unobservable and uncontrollable transitions there are no arcs from controller places to transitions in T_{uc} and no arcs from transitions in $T_{uo,uc}$ to controller places. In models of real systems it is not always possible.

2.1.2 Real Enforceability

In real conditions the ideal enforceability does not exist. In general, the solution will not be possible in terms of the original specification. In such a case it is necessary to find modified constrains (control specifications) in the form $\mathbf{L}'.\mathbf{x} \leq \mathbf{b}'$ and compute the controller $\mathbf{B}_s = -\mathbf{L}'\mathbf{B}$ for the new specifications. Here, $\mathbf{x_0^s} = \mathbf{b}' - \mathbf{L}'\mathbf{x}_0$. The problem is solved when we succeed in finding a suitable $\mathbf{L}'$ an $\mathbf{b}'$.

As it was proved in [15], the specifications can have the following form:

$$\mathbf{L}' = (\mathbf{R}_1 + \mathbf{R}_2\mathbf{L}), \tag{13}$$

$$\mathbf{b}' = \mathbf{R}_2(\mathbf{b} + \mathbf{1}^{n_s \times 1}) - \mathbf{1}^{n_s \times 1} \tag{14}$$

where $\mathbf{1}^{n_s \times 1} = (1, \ldots, 1)^T$, $\mathbf{R}_1 \in \mathbb{Z}^{n_s \times n}$ satisfies $\mathbf{R}_1.\mathbf{x} \geq \mathbf{0}$, $\forall \mathbf{x}$, $\mathbf{R}_2 \in \mathbb{Z}^{n_s \times n_s}$ be a diagonal matrix of natural numbers (i.e. positive-definite diagonal matrix of integers).

Choose the entries for $\mathbf{R}_1$ and $\mathbf{R}_2$ to ensure the ideal enforceability. According to conditions (10)–(12) concerning the ideal enforceability introduced and described above, the following relations have to hold:

$$(\mathbf{R}_1 + \mathbf{R}_2\mathbf{L})\mathbf{B}_{o,uc} \leq \mathbf{0}, \tag{15}$$

$$(\mathbf{R}_1 + \mathbf{R}_2\mathbf{L})\mathbf{B}_{uo,uc} = \mathbf{0}, \tag{16}$$

$$(\mathbf{R}_1 + \mathbf{R}_2\mathbf{L})\mathbf{x}_0 \leq \mathbf{R}_2(\mathbf{b} + \mathbf{1}^{n_s \times 1}) - \mathbf{1}^{n_s \times 1}. \tag{17}$$

In [15] it was proved that when

$$[\mathbf{R}_1\mathbf{R}_2] \cdot \begin{bmatrix} \mathbf{B}_{uc} & \mathbf{B}_{uo} & -\mathbf{B}_{uo} & \mathbf{x}_0 \\ \mathbf{L}.\mathbf{B}_{uc} & \mathbf{L}.\mathbf{B}_{uo} & -\mathbf{L}.\mathbf{B}_{uo} & \mathbf{L}.\mathbf{x}_0 - \mathbf{b} - \mathbf{1}^{n_c \times 1} \end{bmatrix} \leq \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{1}^{n_c \times 1} \end{bmatrix}, \tag{18}$$

then the controller

$$\mathbf{B}_s = -(\mathbf{R}_1 + \mathbf{R}_2\mathbf{L})\mathbf{B} = -\mathbf{L}'\mathbf{B}, \tag{19}$$

$$\mathbf{x}_s^0 = \mathbf{R}_2(\mathbf{b} + \mathbf{1}^{n_c \times 1}) - \mathbf{1}^{n_c \times 1} - (\mathbf{R}_1 + \mathbf{R}_2\mathbf{L})\mathbf{x}_0 = \mathbf{b}' - \mathbf{L}'\mathbf{x}_0 \tag{20}$$

exists. Moreover, it causes that all subsequent markings of the closed loop system satisfy the constraint $\mathbf{Lx} \leq \mathbf{b}$. This is achieved without any attempt to inhibit uncontrollable transitions and without detecting unobservable transitions. The advantage of such an approach is that the matrices $\mathbf{R}_1$, $\mathbf{R}_2$ can be generated.

2.1.3 Example – Comparison of Ideal and Real Enforceability

To illustrate the difference between the ideal and real enforceability of control interferences let us introduce the simple PN given in Figure 1 where the transition t_4 is uncontrollable. Namely, the matrix $\mathbf{B}$ consists of two submatrices $(\mathbf{B}_{o,c}, \mathbf{B}_{o,uc})$ as follows:

$$\mathbf{B} = (\mathbf{B}_{o,c}, \mathbf{B}_{o,uc}) = \left(\begin{array}{ccc|c} 0 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 \end{array}\right). \tag{21}$$

Suppose, that in each step k of the system evolution the condition

$$\sigma_{p_2} + 3.\sigma_{p_4} \leq 3, \quad k = 0, 1, \ldots \tag{22}$$

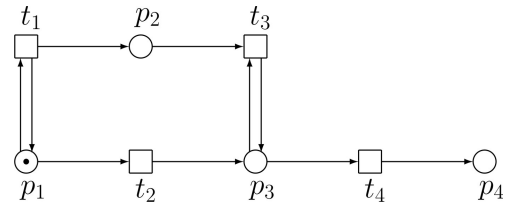

Figure 1. The given uncontrolled PN model

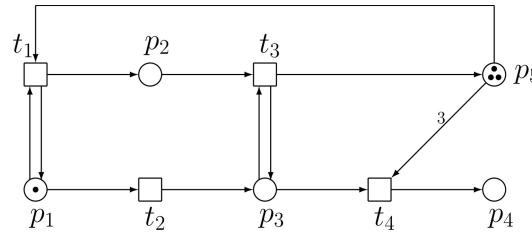

Figure 2. The trial to apply the ideal enforceability

has to be met. Consequently,

$$\mathbf{L} = (0, 1, 0, 3), \quad \mathbf{b} = 3. \tag{23}$$

Because $\mathbf{x}_0 = (1, 0, 0, 0)^T$, with respect to (4)–(7) we can obtain the supervisor (corresponding to the ideal enforcing) given in Figure 2 where the structure of the supervisor and its initial state are the following:

$$\mathbf{B}_s = -\mathbf{L}.\mathbf{B} = (-1, 0, 1, -3); \quad \mathbf{x}_s^0 = \mathbf{b} - \mathbf{L}.\mathbf{x}_0 = 3. \tag{24}$$

However, the condition (10) is not satisfied, because

$$-\mathbf{L}.\mathbf{B}_{o,uc} = -(0, 1, 0, 3).(0, 0, -1, 1)^T = -3 \not\geq 0. \tag{25}$$

It means that the ideal enforceability is impossible. Consequently, the real enforceability approach has to be applied. Therefore, consider

$$\mathbf{R}_1 = (0, 0, 3, 0); \quad R_2 = 1. \tag{26}$$

Then, because of (13), (14),

$$\mathbf{L}' = (0, 1, 3, 3); \quad \mathbf{b}' = 3, \tag{27}$$

$$\mathbf{B}'_s = -\mathbf{L}'.\mathbf{B} = (-1, -3, 1, 0); \quad \mathbf{x}_s^0 = \mathbf{b}' - \mathbf{L}'.\mathbf{x}_0 = 3. \tag{28}$$

The supervised system is given in Figure 3. Here, the inequality (15) is fulfilled

because

$$(\mathbf{R}_1 + \mathbf{R}_2\mathbf{L})\mathbf{B}_{o,uc} = \mathbf{L}'.\mathbf{B}_{o,uc} = (0,1,3,3).(0,0,-1,1)^T = 0 \tag{29}$$

and it should be ≤ 0.

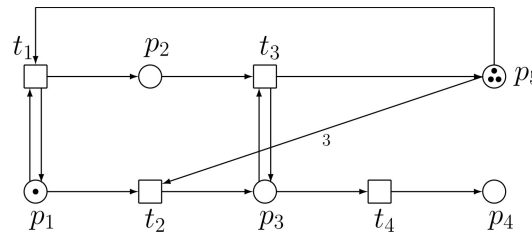

Figure 3. The real enforceability

The reachability trees of the three versions of the PN model structure are given in Figure 4.

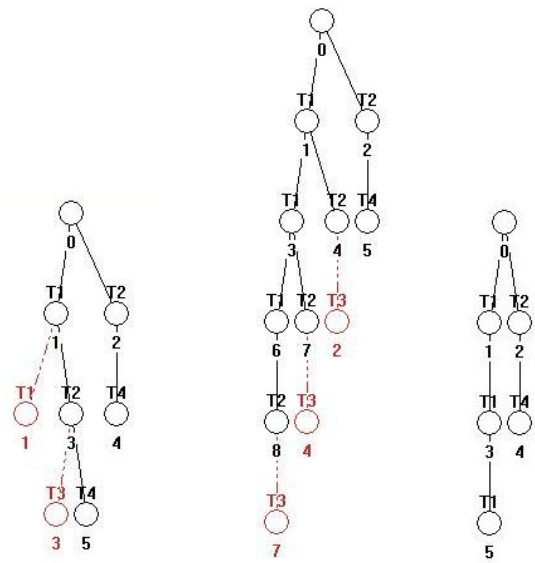

Figure 4. The reachability trees corresponding (from the left to the right) with the PN models in Figures 1, 2 and 3, respectively, with the corresponding reachability matrices $\mathbf{X}_a$, $\mathbf{X}_b$, $\mathbf{X}_c$

Corresponding nodes of these trees are represented by the columns of the reachability matrices as follows:

$$\mathbf{X}_a = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & \omega & 0 & \omega & 0 & \omega \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \tag{30}$$

where ω means unlimited number of tokens in the place p_2 in three reachable states, namely $\mathbf{x}_1$, $\mathbf{x}_3$ and $\mathbf{x}_5$, because of the self-loops in the RT nodes $\mathbf{x}_1$ and $\mathbf{x}_3$, and

$$\mathbf{X}_b = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 2 & 1 & 0 & 3 & 2 & 3 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 3 & 2 & 3 & 1 & 2 & 0 & 0 & 1 & 0 \end{pmatrix}, \tag{31}$$

$$\mathbf{X}_c = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 2 & 0 & 3 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 3 & 2 & 0 & 1 & 0 & 0 \end{pmatrix}. \tag{32}$$

2.1.4 Local Summary

In case where no uncontrollable and unobservable transitions occur, the controller can directly observe and prevent all transitions of PN model of the plant. However, it is much more realistic to abandon such an assumption in PN models of DES working in practice. Namely:

1. Transition t_j may be uncontrollable. It means, that the controller will not be able to directly prevent the transition from firing. In such a case, there will be no arc from any controller place to $t_j \in T$.
2. Transition $t_j \in T$ may be unobservable. It means, that the controller will not be able to directly detect the firing of the transition. Thus, the firing of t_j cannot affect the number of tokens in any controller place (i.e. the *feedback* from the plant to the controller is missing).

This implies that there are neither arcs from an unobservable transition t_j to any controller place p_i, $i = 1, \ldots s$, $j = 1, \ldots m$ nor the arc from any controller place p_i to t_j.

Hence, a transition being unobservable implies that it is also uncontrollable. Therefore, controllability of a transition implies its observability. Three different kinds of transitions are distinguished:

1. controllable;
2. uncontrollable but observable; and
3. uncontrollable and unobservable.

In such a case the enforceability of control interferences needs not always be ideal and the supervisor synthesis has to by modified – compare the pair (6), (7) with the pair (19), (20).

3 ERROR RECOVERY

As it was mentioned in the Section 1.4, many times errors (failures) occur during DES operation. For example in FMS a part may drop out from the robot gripper. The system has to detect what was wrong and recover the normal behaviour – i.e., to eliminate the influence of the error on the system behaviour. The error recovery is a set of actions that must be performed in order to return the system to its normal state. It is necessary

1. to synthesize the recovery sequence, and
2. to extend the scope of the controller activity in order to deal with the fault.

The problem is directly connected with reachability.

After the occurrence of the fault the system development is different than the standard one. Consider the situation after the occurrence of the fault $\mathbf{f}_j = (0, \ldots, 0, 1, 0, \ldots, 0)^T$ being the unit vector with dimensionality m (the number of transitions). Here $\mathbf{x}_k = \mathbf{x}_j + \mathbf{B}_f.\mathbf{f}_j$, where $\mathbf{B}_f$ is the structural matrix of the faulty system submodel.

In practice, there are many areas where it comes toward errors of different kinds. Let us illustrate two such areas – FMS and a transport system – namely, the robotized cell and the railroad crossing.

3.1 Error Recovery of a Kind of FMS

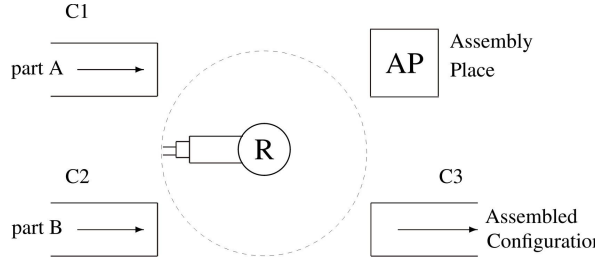

Figure 5. The scheme of the robotized assembly cell

Consider the robotized cell schematically displayed in Figure 5. Here, C1 represents a conveyor feeding parts of a kind A into the cell, C2 expresses a conveyor feeding parts of a kind B into the cell, and C3 pictures a conveyor carrying away the final product – i.e. the assembled part C (i.e. A + B) prepared in the assembly place AP – from the cell. The robot R plays the central role in the cell, because it serves all other devices inside the cell (i.e. C1–C3 and AP).

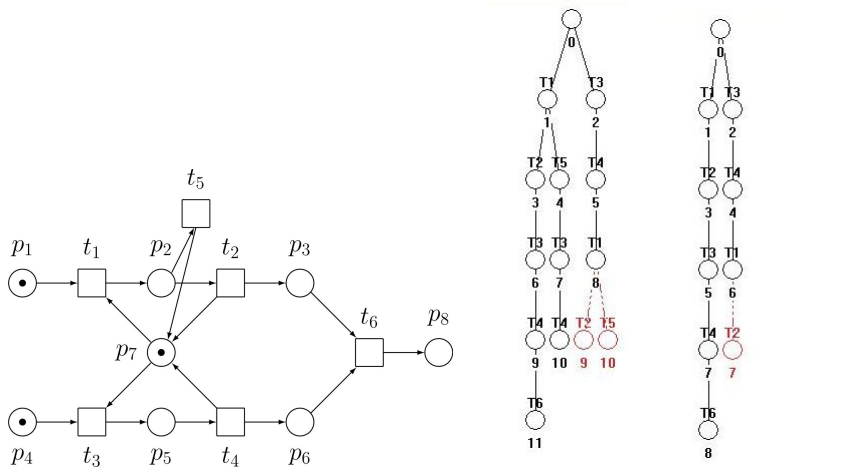

Figure 6. The PN model of the fragment of the plant (left), its full RT including firing of t_5 (middle) and the RT of the model without firing of t_5, i.e. without the occurrence of the error event (right)

The fragment of the global PN model of the robotized cell, displayed in Figure 6 left, models the handling of first two belts by R. The robot consecutively takes away parts A, B from two transport belts C1 and C2, respectively. In the PN model the supplying is realized by means of the places p_1 and p_4, respectively. The operations of taking parts are modelled by p_2, p_5, respectively. The parts A, B prepared to be inserted into the assembly place are modelled by places p_3, p_6, respectively. The robot puts them subsequently into AP modelled by p_8. Although the process is deterministic a fault can occur. The transition t_5 represents the error event – i.e. the fault when a part A drops out from the robot gripper. The fault itself is modelled by firing the transition t_5. The corresponding RT of the model segment, where the fault occurs, is displayed in the middle of the Figure 6. The RT corresponding to the normal situation, when no fault occur, is given in Figure 6 right. The state space of the system (the nodes of the full RT) are represented by the columns of the

matrix (33):

$$\mathbf{X}_r = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{33}$$

Because the fault event is spontaneous, it cannot be prevented. There is only a possibility (if any) for trying to recover the system operation after occurring the fault, i.e. to remove the effects of the fault on the system operation. We must watch the consecutive states of the system after occurring the fault – i.e. the states $\mathbf{x}_4 = (0\,0\,0\,1\,0\,0\,1\,0)^T$ and $\mathbf{x}_{10} = (0\,0\,0\,0\,0\,1\,1\,0)^T$. From these states the recovery procedure has to be evolved. Note, that the first column of the matrix (33) is $\mathbf{x}_0$. Consequenly, $\mathbf{x}_4$ and $\mathbf{x}_{10}$ are displayed, respectively, as 5^{th} and 11^{th} columns of the matrix). However, the state $\mathbf{x}_{10}$ is inadmissible. Moreover, no further development is allowed from it, i.e. $\mathbf{x}_{10}$ has a form of a deadlock. Instead of $\mathbf{x}_{10}$ the recovered state $\mathbf{x}_9 = (0\,0\,1\,0\,0\,1\,1\,0)^T$ is expected. Also the states of PN without the fault – i.e. without firing t_5 – has to be taken into account in order to see the correct development. The form of the PN is the same like that on the left side of Figure 6, only the transition t_5 is not firing (i.e. as if missing). RT of such a PN model is given on the right side of the Figure 6. Its nodes are the columns of the matrix

$$\mathbf{X}_r = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{34}$$

A trial how the fault can be recovered in order to normalize the system behaviour is expressed by amending the PN model. The PN model able to recover the fault is given in Figure 7. The model covers both the normal behaviour and the recovery of the fault behaviour. Its RT is given in Figure 8. The trajectory of the system development when the fault occurred (i.e. when t_5 was fired) and then its entail was recovered is the following: $\mathbf{x}_0 \overset{t_1}{\rightarrow} \mathbf{x}_1 \overset{t_5}{\rightarrow} \mathbf{x}_5 \overset{t_3}{\rightarrow} \mathbf{x}_{11} \overset{t_4}{\rightarrow} \mathbf{x}_{18} \overset{t_7}{\rightarrow} \mathbf{x}_{25} \overset{t_8}{\rightarrow} \mathbf{x}_{31} \overset{t_1}{\rightarrow} \mathbf{x}_{35} \overset{t_2}{\rightarrow} \mathbf{x}_{38} \overset{t_6}{\rightarrow} \mathbf{x}_{39}$. Of course, also the sideways branch starting from $\mathbf{x}_5$: $\mathbf{x}_5 \overset{t_7}{\rightarrow} \mathbf{x}_{12} \ldots \overset{t_4}{\rightarrow} \mathbf{x}_{38}$ may be taken into account. However, the RT expresses also the trajectories of the system development where the firing of t_5 does not occur (i.e. when no fault occurs),

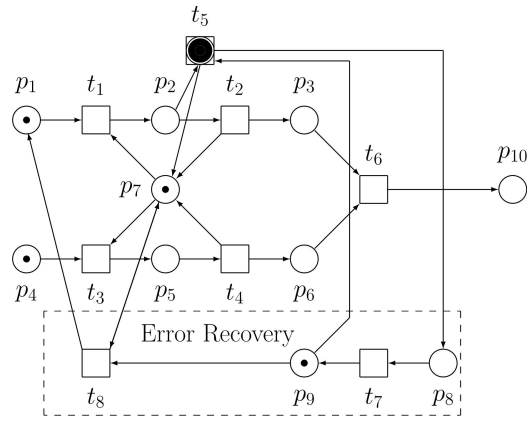

Figure 7. The PN model of the recovered system

e.g.: $\mathbf{x}_0 \overset{t_1}{\rightarrow} \mathbf{x}_1 \overset{t_2}{\rightarrow} \mathbf{x}_4 \overset{t_3}{\rightarrow} \mathbf{x}_9 \overset{t_4}{\rightarrow} \mathbf{x}_{15} \overset{t_6}{\rightarrow} \mathbf{x}_{22} \ldots \overset{t_2}{\rightarrow} \mathbf{x}_{37}$ (together with its sideways branches). The RT nodes are represented by the columns of the matrix

$$\mathbf{X}_r = \begin{pmatrix} 1\,0\,1\,2\,0\,0\,1\,1\,2\,0\,1\,0\,0\,0\,2\,0\,0\,1\,0\,0\,1\,1\,0\,1\,0\,0\,0\,1\,1\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,1\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,1\,0\,0\,0\,1\,0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,1\,0\,0\,0\,0\,1\,1\,1\,0\,0\,0\,0\,0\,1\,2\,0\,0\,0\,0\,1\,2\,0\,1\,0\,2\,0\,1\,1\,1\,0 \\ 1\,1\,0\,1\,1\,1\,0\,1\,0\,0\,1\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,1\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,1\,0\,0\,0\,0\,0\,1\,1\,0\,1\,0\,0\,0\,0\,1\,0\,1\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,1\,0\,0\,0\,0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,1\,1\,1\,0\,0\,1\,0\,0\,1\,0\,1\,0\,1\,0\,0\,0\,1\,0\,1\,0\,0\,1\,1\,0\,0\,1\,0 \\ 1\,0\,0\,1\,1\,1\,1\,0\,0\,0\,1\,0\,1\,0\,1\,1\,0\,0\,1\,0\,1\,0\,1\,1\,1\,1\,0\,0\,1\,0\,0\,1\,1\,0\,1\,0\,0\,1\,1 \\ 0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 1\,1\,1\,0\,1\,0\,1\,0\,0\,1\,0\,0\,1\,1\,0\,1\,0\,0\,0\,1\,0\,0\,1\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,1\,0\,1 \end{pmatrix}. \quad (35)$$

3.2 Error Recovery in a Segment of Transport Systems

Consider an example of the simple railroad crossing (RC). The RC gate prevents a direct contact of trains with vehicles on the road. The global PN model of RC given in Figure 9 consists of three cooperating sub-models expressing the behaviour of the

1. train,
2. crossing gate, and
3. control system.

The firing of the transition t_{f_5} models the occurrence of the failure(s). In general, the failure can occur more times. Thus, the marking of the place p_{14} (i.e. the

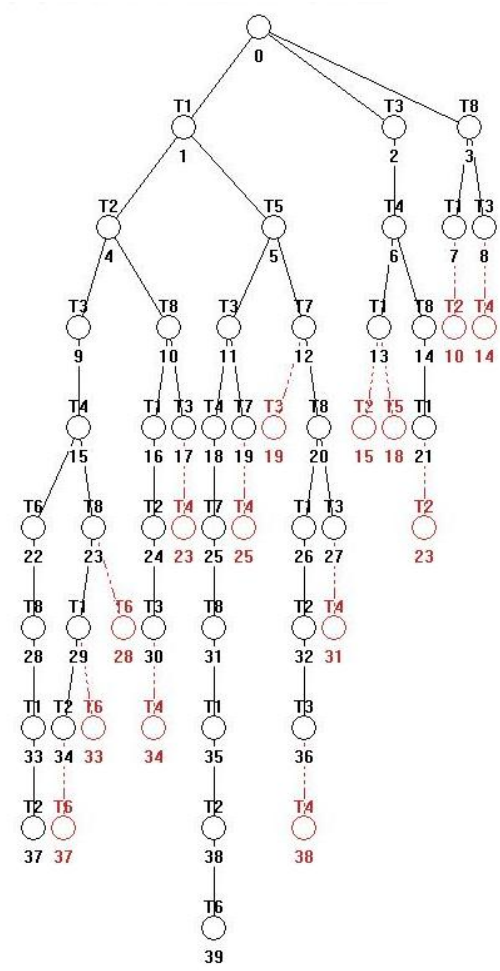

Figure 8. The RT of the PN model of the recovered system

number of its tokens) represents the number of the failure occurrences. What is very dangerous in doing so is that the firing of t_{f_5} involves an erroneous generation of a token in p_{l0} which directly influences the position of the barrier. Such an error may cause accidents. Many accidents allover the world has been caused due to such errors.

The following depicts the purport of places in the failure-free cases: As to the train, its states regarding the crossing are: p_1 = approaching; p_2 = being before; p_3 = being within; p_4 = being after. The states of the barrier are: p_{11} = up; p_{12} = down. The transitions t_6 and t_7 model, respectively, the events of raising and

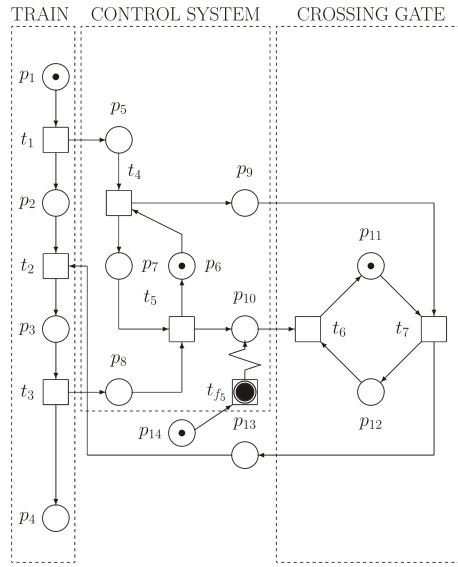

Figure 9. The PN model of the RC with the failure expressed by firing t_{f_5}

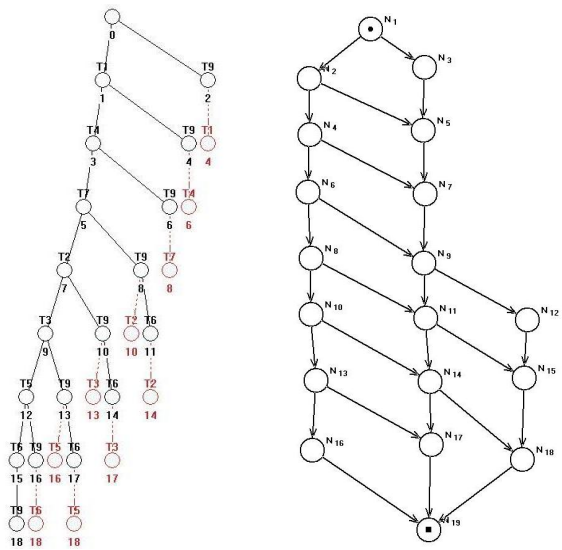

Figure 10. The RT of the PN model of RC with only one possible occurrence of the failure represented by t_{f_5} (left), and the corresponding RG (right)

lowering the barrier. The states of the control system are: $p_5 - p_{10}$. The place p_{13} models the interlock. Being active (having token) it gives the warning signal for the train – the alert that the barrier is still up.

Very dangerous (critical as to safety reasons) is the situation when the barrier is going up, and simultaneously, the failure t_{f_5} occurs. For detection of such situation the redundant information is needed. The control system must issue such information. It can be seen that the places p_6 and p_7 in the control system correspond to p_{11} and p_{12} in the real crossing gate. If p_7 and p_{11} are active simultaneously, a contradiction between the fault situation (real) and the standard situation (normal, error free) is detected. For the error recovery it is necessary to set what state is accepted to be the true one. Supposing that the barrier is up and drops down, the recovery is realized by means of the transition t_{r_1}.

Considerably simpler situation occurs when the barrier is up and none train is approaching. Namely, by means of the transition t_{r_2} the fail signal p_{10} from the control system in a close touch with the activity of the place p_{11} ensures that the fail signal can be practically ignored.

The RT and RG of the failed system are given in Figure 10. The particular nodes of RT/RG are the columns of the reachability matrix

$$\mathbf{X}_{reach} = \begin{pmatrix} 1\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,1\,0\,1\,1\,1\,1\,0\,1\,0\,0\,1\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,1\,1\,1\,0\,1\,1\,1\,1 \\ 0\,1\,0\,0\,1\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 1\,1\,1\,0\,1\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,1\,1\,0\,1 \\ 0\,0\,0\,1\,0\,1\,1\,1\,1\,1\,1\,1\,0\,1\,1\,0\,0\,1\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0 \\ 0\,0\,0\,1\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,1\,0\,0\,2\,0\,1 \\ 1\,1\,1\,1\,1\,0\,1\,0\,0\,0\,0\,1\,0\,0\,1\,1\,0\,1\,1 \\ 0\,0\,0\,0\,0\,1\,0\,1\,1\,1\,1\,0\,1\,1\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0\,1\,0\,0\,1\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0 \\ 1\,1\,0\,1\,0\,1\,0\,1\,0\,0\,1\,0\,0\,1\,0\,0\,0 \end{pmatrix}. \tag{36}$$

The PN model of the system with the recovered error is given in Figure 11, while the RT and RG are displayed in Figure 12.

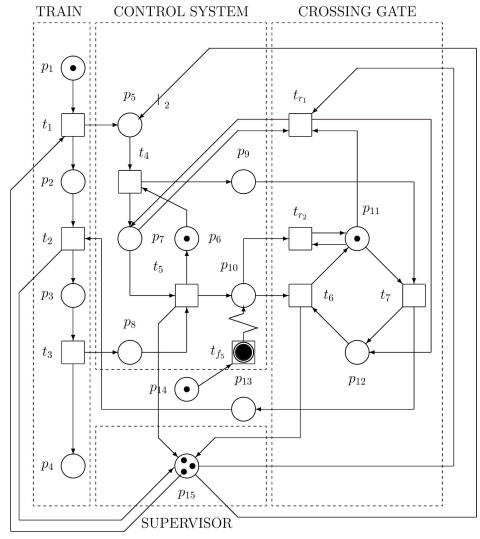

Figure 11. The PN model of the system with the recovered failure and with the supervisor removing the deadlock occurring during the process of the failure recovery

The RT/RG nodes corresponding to such model are represented by the columns of the reachability matrix

$$\mathbf{X}^{s}_{reach} = \left(\begin{array}{cccccccccccccccccccccccccc}
1&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&1&1&0&1&1&1&0&1&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&1&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&0&0&1&1&1&1&1\\
0&1&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&1&1&0&1&1&0&0&1&0&0&0&0&0&0&0&1&0&0&0&1&1&0&0&1&1\\
0&0&0&1&0&0&1&1&0&1&1&1&1&1&1&1&0&1&1&1&0&0&1&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&1&0\\
0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&1&0&0&1&0&0&1&0&0&1&0&0&1&1&0&0&0&2&0&0&1&1\\
1&1&1&1&1&1&0&1&1&0&0&1&0&0&1&0&0&0&1&0&1&0&1&0&1&0\\
0&0&0&0&0&0&1&0&0&1&1&0&1&1&0&1&1&1&0&1&0&1&0&1&0&1\\
0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0\\
1&1&0&1&0&0&1&0&0&1&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0\\
-&-\\
3&2&3&0&2&3&0&0&2&1&0&0&1&1&1&0&2&1&2&1&3&2&2&1&3&2&3
\end{array}\right) \tag{37}$$

where the last row represents the states (marking) of the supervisor in particular RT/RG nodes. More details concerning the multiplicity of the error occurrence are discussed in [4].

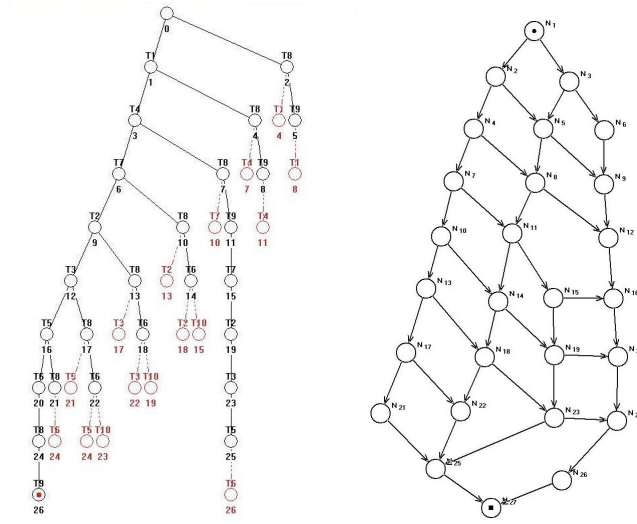

Figure 12. The RT of the recovered system with removed deadlock by means of the supervisor (left) and the corresponding RG (right)

4 CASE STUDY ON CONTROL OF COMPLETE FMS BY MEANS OF IPN

Having resolved the error recovery of the FMS fragment in Section 3.1, let us apply the IPN-based approach to control the complete FMS including the error. The uncontrolled PN model of the plant is the expanded form of the fragment given in Figure 6 with the same fault, of course. The IPN-based control of it (before the error recovering) is presented in Figure 13.

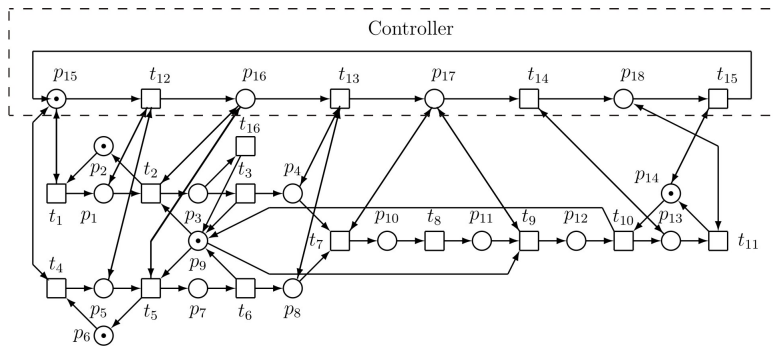

Figure 13. The PN model of the controlled system

The RT of the PN model is given in Figure 14. To save a space, it is turned by 90 degrees to the left.

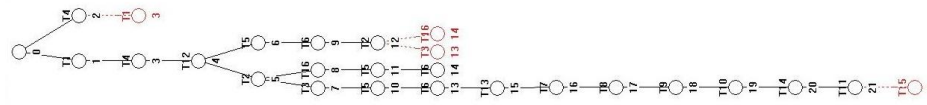

Figure 14. RT of the PN model of the controlled system

The reachability set, i.e. the particular nodes of the RT in the form of the state vectors are represented by means of the columns of the following matrix

$$\mathbf{X}_r = \begin{pmatrix}
0&1&0&1&1&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&1&0&0&1&0&1&1&0&1&1&1&1&1&1&1&1&1&1&1&1\\
0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&1&0&0&1&0&1&0&0&0&0&0&0\\
0&0&1&1&1&1&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&1&0&0&0&0&1&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1\\
0&0&0&0&0&0&1&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&1&1&1&1&0&0&0&0&0&0\\
1&1&1&1&1&0&0&1&1&1&0&0&0&1&1&1&1&1&0&1&1&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&0&1\\
1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1
\end{pmatrix}. \tag{38}$$

4.1 Error Recovery

Let us recover the fault analogically to the approach presented in Figure 7. As we can see in Figure 14, the occurrence of the fault represented by the transition t_{16} (when it is firing) tends to the deadlock state $\mathbf{x}_{14}$ (see the column 15 of the matrix (38) because RT node N1 corresponds to $\mathbf{x}_0$) presented by active places p_2, p_6, p_8, p_9, p_{14}, p_{16}, all of them with marking equal to 1. To prevent such situation the recovery must tend to the elimination of the deadlock state $\mathbf{x}_{14}$ (caused by firing t_{16}) and simultaneously recover the system operation.

A trial directed towards the error recovery proposal is given in Figure 15. The RT of the system is given in Figure 16 while the corresponding complete RG of the recovered system representing all system trajectories is displayed in Figure 17.

There exist four production cycles – i.e. the trajectories/paths starting from $\mathbf{x}_0$ and ending also in $\mathbf{x}_0$ – in the RG of the recovered system displayed in Figure 17. Their lengths are 15, 22, 29, and 30 steps (a step represents the firing of a transition). They can be analyzed separately. These cycles include the internal sub-cycles (sub-trajectories).

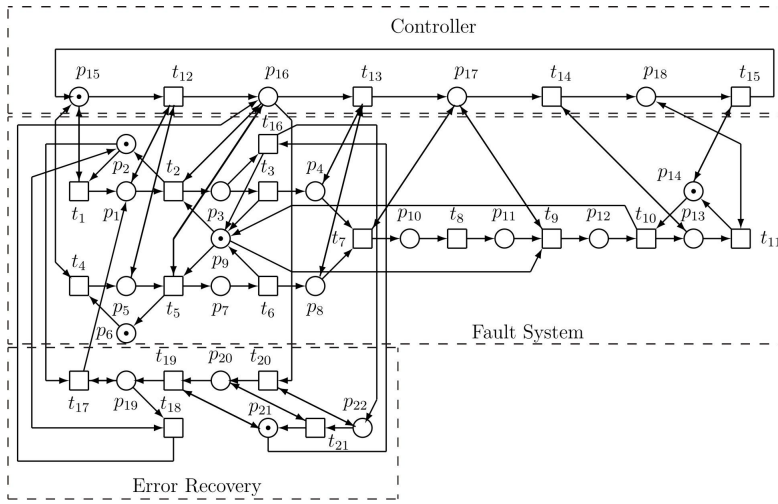

Figure 15. The PN model of the controlled system able to recover the failure

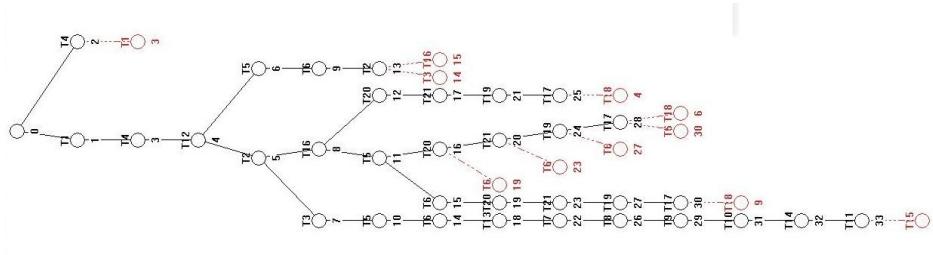

Figure 16. The RT of the PN model of the controlled system with the recovered failure

Let us analyze in detail the shortest cycle with 15 steps – it is given in Figure 18. It contains four sub-cycles (sub-trajectories). None of them contains a deadlock. Particular steps in the trajectories are realized by successive firing of transitions in corresponding sequences. They realize the transition from a state to another (next)

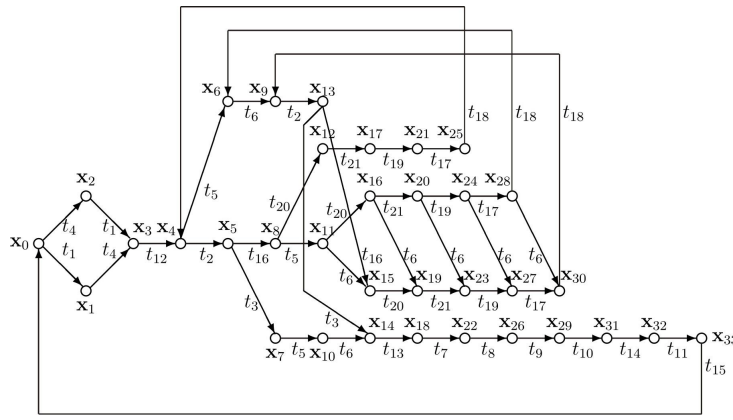

Figure 17. The corresponding RG of the recovered system

state in the trajectories.

$$\mathbf{X}_r = \begin{pmatrix}
0\,1\,0\,1\,1\,0\,1\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,1\,0\,1\,0\,0\,0 \\
1\,0\,1\,0\,0\,1\,0\,1\,1\,0\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,0\,1\,1\,0\,1\,0\,1\,1\,1 \\
0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,1\,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\,0\,0\,0\,0\,1\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\
0\,0\,1\,1\,1\,1\,0\,1\,1\,0\,0\,0\,1\,0\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0 \\
1\,1\,0\,0\,0\,0\,1\,0\,0\,1\,1\,1\,0\,1\,1\,1\,1\,0\,1\,1\,1\,0\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1 \\
0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,1\,1\,0\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0 \\
0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,1\,1\,1\,0\,0\,1\,1\,0\,0\,0\,1\,0\,0\,0\,1\,1\,0\,0\,1\,0\,0\,0 \\
1\,1\,1\,1\,1\,0\,0\,1\,1\,1\,0\,0\,1\,0\,1\,1\,0\,1\,1\,1\,0\,1\,1\,1\,0\,1\,1\,1\,0\,0\,1\,1\,1\,1\,1 \\
0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,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\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,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\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,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\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,0 \\
1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,0\,0\,1 \\
1\,1\,1\,1\,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\,0\,0\,0\,0\,0\,0 \\
0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1\,0\,1\,1\,1\,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\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,1\,0\,1\,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\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1 \\
0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,1\,1\,0\,1\,1\,0\,1\,0\,0\,0 \\
0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,1\,1\,0\,1\,1\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\
1\,1\,1\,1\,1\,1\,1\,1\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,0\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1 \\
0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0
\end{pmatrix} \tag{39}$$

Let as analyze the shortest path in details. As we can see from the RT/RG, as a matter of fact the trajectory (path) aggregates four sub-trajectories:

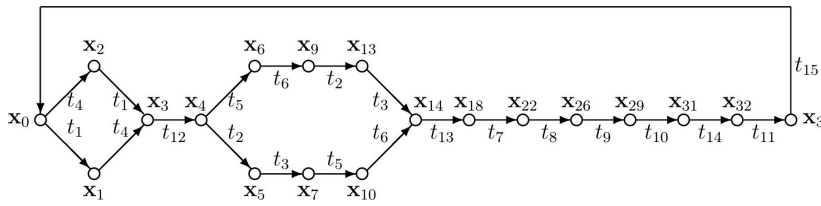

Figure 18. The 15 steps cycle in the RG of the recovered system

1. $\mathbf{x}_0 \xrightarrow{t_4} \mathbf{x}_2 \xrightarrow{t_1} \mathbf{x}_3 \xrightarrow{t_{12}} \mathbf{x}_4 \xrightarrow{t_5} \mathbf{x}_6 \xrightarrow{t_6} \mathbf{x}_9 \xrightarrow{t_2} \mathbf{x}_{13} \xrightarrow{t_3} \mathbf{x}_{14} \xrightarrow{t_{13}} \mathbf{x}_{18} \xrightarrow{t_7} \mathbf{x}_{22} \xrightarrow{t_8}$
   $\mathbf{x}_{26} \xrightarrow{t_9} \mathbf{x}_{29} \xrightarrow{t_{10}} \mathbf{x}_{31} \xrightarrow{t_{14}} \mathbf{x}_{32} \xrightarrow{t_{11}} \mathbf{x}_{33} \xrightarrow{t_{15}} \mathbf{x}_0$;
2. $\mathbf{x}_0 \xrightarrow{t_4} \mathbf{x}_2 \xrightarrow{t_1} \mathbf{x}_3 \xrightarrow{t_{12}} \mathbf{x}_4 \xrightarrow{t_2} \mathbf{x}_5 \xrightarrow{t_3} \mathbf{x}_7 \xrightarrow{t_5} \mathbf{x}_{10} \xrightarrow{t_4} \mathbf{x}_{14} \xrightarrow{t_{13}} \mathbf{x}_{18} \xrightarrow{t_7} \mathbf{x}_{22} \xrightarrow{t_8}$
   $\mathbf{x}_{26} \xrightarrow{t_9} \mathbf{x}_{29} \xrightarrow{t_{10}} \mathbf{x}_{31} \xrightarrow{t_{14}} \mathbf{x}_{32} \xrightarrow{t_{11}} \mathbf{x}_{33} \xrightarrow{t_{15}} \mathbf{x}_0$;
3. $\mathbf{x}_0 \xrightarrow{t_1} \mathbf{x}_1 \xrightarrow{t_4} \mathbf{x}_3 \xrightarrow{t_{12}} \mathbf{x}_4 \xrightarrow{t_5} \mathbf{x}_6 \xrightarrow{t_6} \mathbf{x}_9 \xrightarrow{t_2} \mathbf{x}_{13} \xrightarrow{t_3} \mathbf{x}_{14} \xrightarrow{t_{13}} \mathbf{x}_{18} \xrightarrow{t_7} \mathbf{x}_{22} \xrightarrow{t_8}$
   $\mathbf{x}_{26} \xrightarrow{t_9} \mathbf{x}_{29} \xrightarrow{t_{10}} \mathbf{x}_{31} \xrightarrow{t_{14}} \mathbf{x}_{32} \xrightarrow{t_{11}} \mathbf{x}_{33} \xrightarrow{t_{15}} \mathbf{x}_0$;
4. $\mathbf{x}_0 \xrightarrow{t_1} \mathbf{x}_1 \xrightarrow{t_4} \mathbf{x}_3 \xrightarrow{t_{12}} \mathbf{x}_4 \xrightarrow{t_2} \mathbf{x}_5 \xrightarrow{t_3} \mathbf{x}_7 \xrightarrow{t_5} \mathbf{x}_{10} \xrightarrow{t_4} \mathbf{x}_{14} \xrightarrow{t_{13}} \mathbf{x}_{18} \xrightarrow{t_7} \mathbf{x}_{22} \xrightarrow{t_8}$
   $\mathbf{x}_{26} \xrightarrow{t_9} \mathbf{x}_{29} \xrightarrow{t_{10}} \mathbf{x}_{31} \xrightarrow{t_{14}} \mathbf{x}_{32} \xrightarrow{t_{11}} \mathbf{x}_{33} \xrightarrow{t_{15}} \mathbf{x}_0$.

Other longer cycles have 22 (with 18 sub-trajectories), 29 (with 32 sub-trajectories), and 30 (with 16 sub-trajectories) steps. The last of them represents practically two consecutive shortest trajectories (i.e. two successive cycles without a failure). While any of the sub-trajectories of the shortest cycle introduced above does not contain the transition t_{16} representing the failure, the sub-trajectories of the longer cycles (namely with 22 steps and 29 steps) do. In spite of this they operate correctly, because of the successful error recovery. It means that the cycle consisting of 15 steps represents the normal operation of the controlled plant, i.e., the situations when no failure occurs. However, the longer cycles are able to deal with the failure (occurring event represented by firing t_{16}) successfully.

Unfortunately, there is not a sufficient space here for a graphical presentation of the trajectories, or all their sub-trajectories. Nevertheless, they can be traced from Figure 17. For illustration, let us introduce them at least in the aggregated form in Figure 19 (up) and Figure 19 (down). They show that the system is able to deal with the failure – i.e., the error recovery was successful.

4.2 Local Summary

The application of IPN for modelling and control of DES with nondeterminism represented by uncontrollable and/or unobservable transitions was presented in this section. IPN seems to be a suitable, sound and relatively simple approach for resolving such tasks, especially technical ones. In Section 5 the application of LbPN will be presented pointing out the difference from IPN.

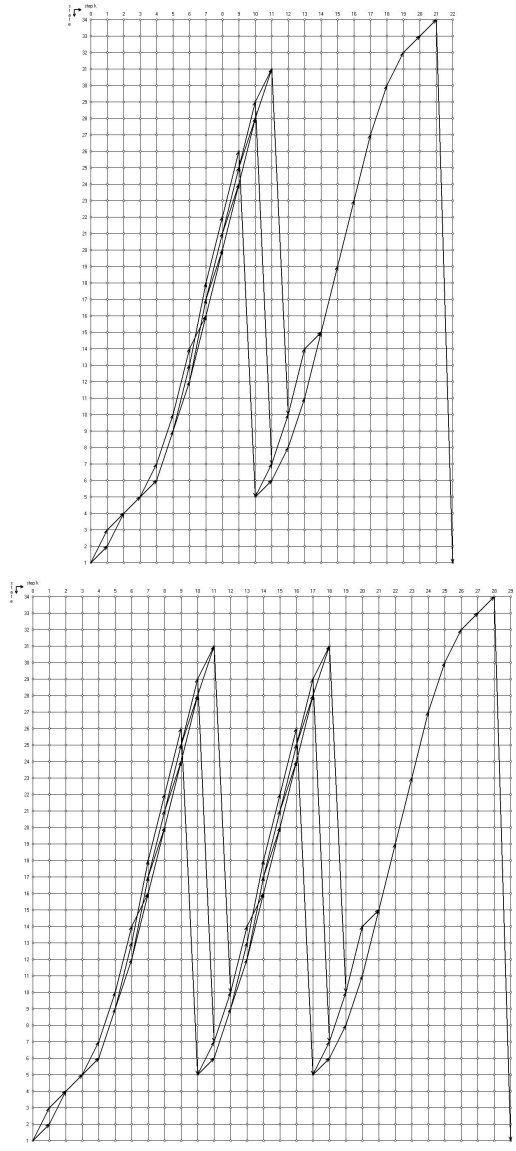

Figure 19. The aggregated form of the path with 22 steps (up) and 29 steps (down). In both cases on the horizontal axes the numbers of steps are displayed, while on the vertical axes the numbers of state vectors (shifted by +1) are displayed – namely, the RG nodes N_{i+1} corresponds to $\mathbf{x}_i$, $i = 0, 1 \ldots$

5 CASE STUDY ON CONTROL OF FMS BY MEANS OF LBPN

Consider the plant (a robotic cell) schematically displayed in Figure 20. There are two production lines producing two different kinds of final products. The plant consists of four machines (M1–M4), four robots (R1–R4), one automatically guided vehicle (AGV) system, one buffer (B) with the finite capacity, two input transport belts I1 and I2 feeding the cell, respectively, by parts of a kind Pa1 and of a kind Pa2. Finally, the output transport belts O1 and O2 export, respectively, the processed final parts Fp1, Fp2 from the cell. A similar production plant was used for another reason in [5], based on [26].

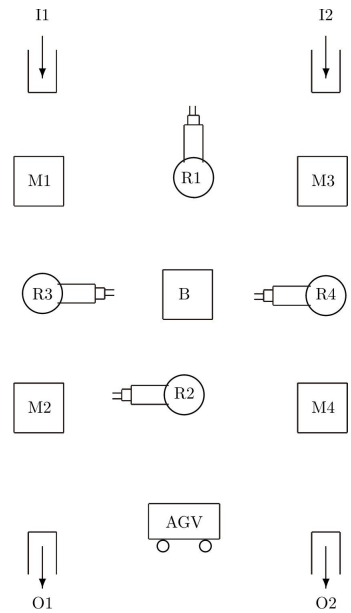

Figure 20. The scheme of the plant

While in an ideal P/T PN model all transitions are considered to be controllable and no faults occur in the plant, in the model of a real plant none of those presumptions holds true. However, when there occur uncontrollable transitions and even some errors, the situation changes. Therefore, let us use the LbPN model given in Figure 21.

The particular sections of the model are denoted by framed badges corresponding to individual devices.

The marking of the place p_1 represents the number of parts (in general α) of the type Pa1 entering the production line PL1 (the left column of devices in Figure 20), while the marking of the place p_{16} represents the number of parts (in general β) of the type Pa2 entering the production line PL2 (the right column of devices in

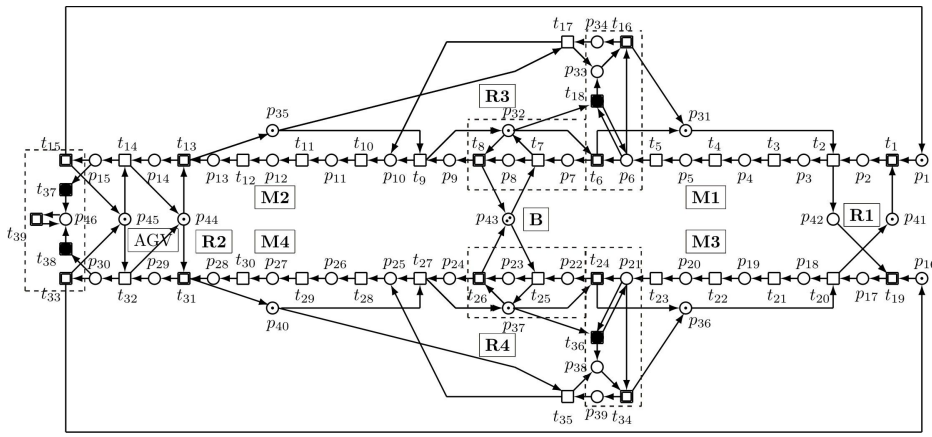

Figure 21. The LbPN model of the real plant with unobservable/uncontrollable transitions

Figure 20). Devices in the central column in Figure 20 – R1, R2 as well as B and AGV – are common for both production lines. The robot R3 belongs solely to PL1 and R4 solely to PL2.

Formally said, p_1 and p_{16} are fed, respectively, by means of I1 and I2. On the other hand t_{15} and t_{33} feed, respectively, O1 with final products Fp1 and O2 with final products Fp2. The marking of the place p_{43} expresses the number of free slots in B (they should be two minimally).

Places from the set $\{p_{33}, p_{34}, p_{38}, p_{39}, p_{46}\}$ represent the *faulty behavior*, i.e. errors. They are marked only if a fault has occurred. Transitions $t_{18}, t_{36}, t_{37}, t_{38}$, filled in the black color, model (by means of their firing) occurrence of the failures. Segments of the PN model containing those faults are demarcated by means of the dashed rectangles with the vertical dimension being greater than the horizontal dimension. The transition t_{18} models a fault of the robot R3 that moves a part from the output buffer of the machine M1 to the input buffer of the machine M2 instead of putting it into the buffer B. In the like manner t_{36} models a fault in the robot R4 that moves a part from the output buffer of the machine M3 to the input buffer of the machine M4 instead of to insert it into the buffer B. Finally, t_{37}, t_{38} model a fault in the AGV. When AGV is working correctly, a completely finished part exits from the robotic cell and a new part enters AGV. If a fault occurs in AGV, parts are not exiting the production lines. Hence, they cannot be replaced by new input parts.

In the model we can see three kinds of transitions:

1. *observable transitions* t_1, t_6, t_8, t_{13}, t_{15}, t_{19}, t_{24}, t_{26}, t_{31}, t_{33}, t_{16}, t_{34}, t_{39}. They are drawn by means of thick lines. The dashed rectangles with the greater width than their height are added in order to cover also t_8 along with t_7 and t_{26} along with t_{25};

2. *unobservable* but *regular transitions* t_2, t_3, t_4, t_5, t_7, t_9, t_{10}, t_{11}, t_{12}, t_{14}, t_{20}, t_{21}, t_{22}, t_{23}, t_{25}, t_{27}, t_{28}, t_{29}, t_{30}, t_{32}. Using the mark ε established in [3] (the Part 1 of this paper), the transitions may be renamed as ε_2, ε_3, ε_4, ε_5, ε_7, ε_9, ε_{10}, ε_{11}, ε_{12}, ε_{14}, ε_{20}, ε_{21}, ε_{22}, ε_{23}, ε_{25}, ε_{27}, ε_{28}, ε_{29}, ε_{30}, ε_{32}, respectively. They are drawn normally, i.e. by means of thin lines;
3. *fault transitions* t_{18}, t_{36}, t_{37}, t_{38}. They are filled in by black colour.

The LbPN model has 46 places and 39 transitions. Because of the great number of places and transitions, and rather complicated structure, the RT/RG may have the extensive number of nodes (i.e. reachable states of the model from a given initial state) depending on the initial state $\mathbf{x}_0$. At the initial state displayed in Figure 21 – i.e., when each of the places p_1, p_{41}, p_{16}, p_{31}, p_{36}, p_{32}, p_{37}, p_{35}, p_{40}, p_{44}, p_{45} possesses one token and p_{43} possesses two tokens – the number of nodes is 1640. The RT was computed in Matlab using the toolbox SPNBOX elaborated in [21]. Consequently, neither RT nor RG can be displayed here because of insufficient space.

It is not easy to compute the RT/RG of such large dimensionality because of long computational time, sometimes also for the limited capacity of the computer memory, especially in case of the initial state containing more entering parts – represented by number of tokens α, β stationed, respectively, in p_1, p_{16} – and/or the higher capacity of the buffer B represented by the number of tokens stationed in p_{43}.

Suppose that each robot is equipped with a sensor. Thus, it always can be observed whether the robot grasps a part (e.g. from a belt) and/or inserts it (e.g. into a machine or B) or not. Taking [3] into account, in LbPN the term $\mathcal{L} = L \cup \varepsilon$ is an alphabet representing a finite set of events, where L represents observable events and ε represents unobservable events. In our case in particular, we can set for observable transitions the following:

1. for R1 $\mathcal{L}(t_1) = a$ and $\mathcal{L}(t_{19}) = e$;
2. for R2 $\mathcal{L}(t_{13}) = c$ and $\mathcal{L}(t_{31}) = l$;
3. for R3 $\mathcal{L}(t_6) = \mathcal{L}(t_8) = \mathcal{L}(t_{16}) = b$;
4. for R4 $\mathcal{L}(t_{24}) = \mathcal{L}(t_{26}) = \mathcal{L}(t_{34}) = g$;
5. providing that it is possible to observe each time when a part is moved by the AGV, also $\mathcal{L}(t_{15}) = \mathcal{L}(t_{33}) = \mathcal{L}(t_{39}) = d$.

The robot R1 always starts taking one part from the first production line. Having denoted (see above) the number of tokens in p_1 as α, while the number of tokens in p_{16} as β, we can analyze the LbPN model behaviour in detail.

All cases in which $\alpha = 0$ present only one node corresponding to M_0. Moreover, all cases in which $\beta = 0$ present 19 nodes in RG.

For $\alpha \neq 0$ and $\beta \neq 0$ many of states occur in RG. For example for initial state where $p_1 = 1$, $p_{16} = 1$, $p_{31} = 1$, $p_{32} = 1$, $p_{35} = 1$, $p_{36} = 1$, $p_{39} = 1$, $p_{40} = 1$, $p_{41} = 1$, $p_{43} = 2$, $p_{44} = 1$, $p_{45} = 1$, 592 nodes occur in RG. However, it is valid for cases without failures.

When the failures occur in the model, the situation is changed. For example when $p_1 = 1$, $p_{16} = 0$, $p_{31} = 1$, $p_{32} = 1$, $p_{33} = 1$, $p_{35} = 1$, $p_{36} = 1$, $p_{38} = 1$, $p_{39} = 1$, $p_{40} = 1$, $p_{41} = 1$, $p_{43} = 2$, $p_{44} = 1$, $p_{45} = 1$, $p_{46} = 1$ the number of RG nodes is 1266. When $p_1 = 0$, $p_{16} = 1$, $p_{31} = 1$, $p_{32} = 1$, $p_{33} = 1$, $p_{35} = 1$, $p_{36} = 1$, $p_{38} = 1$, $p_{39} = 1$, $p_{40} = 1$, $p_{41} = 1$, $p_{43} = 2$, $p_{44} = 1$, $p_{45} = 1$, $p_{46} = 1$ the number of RG nodes is 19.

However, in case when $p_1 = 0$, $p_{16} = 1$, $p_{31} = 1$, $p_{32} = 1$, $p_{33} = 1$, $p_{35} = 1$, $p_{36} = 1$, $p_{37} = 1$, $p_{38} = 1$, $p_{40} = 1$, $p_{41} = 1$, $p_{43} = 2$, $p_{44} = 1$, $p_{45} = 1$ the number of RG nodes is 10, for $p_1 = 1$, $p_{16} = 0$, $p_{31} = 1$, $p_{32} = 1$, $p_{33} = 1$, $p_{35} = 1$, $p_{36} = 1$, $p_{37} = 1$, $p_{38} = 1$, $p_{40} = 1$, $p_{41} = 1$, $p_{43} = 2$, $p_{44} = 1$, $p_{45} = 1$ the number of RG nodes is 516.

But, for $p_1 = 1$, $p_{16} = 1$, $p_{31} = 1$, $p_{32} = 1$, $p_{33} = 1$, $p_{35} = 1$, $p_{36} = 1$, $p_{37} = 1$, $p_{38} = 1$, $p_{40} = 1$, $p_{41} = 1$, $p_{43} = 2$, $p_{44} = 1$, $p_{45} = 1$ the number of RG nodes is 2 904, for $p_1 = 2$, $p_{16} = 1$, $p_{31} = 1$, $p_{32} = 1$, $p_{33} = 1$, $p_{35} = 1$, $p_{36} = 1$, $p_{37} = 1$, $p_{38} = 1$, $p_{40} = 1$, $p_{41} = 1$, $p_{43} = 2$, $p_{44} = 1$, $p_{45} = 1$ the number of RG nodes is 20 356.

For greater α, β the numbers of RG nodes reach tens thousands, even hundred thousands.

Because PL1 is perfectly symmetric with PL2, the number of RG nodes does not change by altering α with β.

Local Summary. In [22] the observation of events is considered as outputs in most problem settings, such as state estimation and fault diagnosis. Some authors, e.g. [18, 20], consider an extended LbPN model enriched with state observations. It is assumed there that the token content in some places of the net is measurable. Such understanding of the extension of LbPN is named as Interpreted PN (IPN). It is suitable especially for technical applications, as shown in the Section 4.

However, in [20] we can see that such a type of LbPN can always be converted into an equivalent standard LbPN by a suitable re-definition of the transition labels, and hence the LbPN-based models and the IPN-based models have the same modeling power. In spite of this it is interesting and useful to utilize the IPN models along with the LbPN models. In general, the scope of LbPN abilities seems to be a little wider than that of IPN because LbPN are suitable also for the fault diagnosis and state estimation (as it was pointed out in this Section). Although both of the approaches are very suitable for analysing and modelling DES containing nondeterminism (as it was demonstrated above), the successful applicability of both kinds of PN (i.e. IPN and LbPN) in the supervisor synthesis may be limited by the complexity of the modelled plant (and especially by the consecutive complexity of its PN model). Namely, the so called curse of dimensionality well known in respect of the Bellman's dynamic programming [2] is, unfortunately, in force also in many other cases including the computation and analysis of RT/RG in PN at the supervisor synthesis. This is generally true for the PN-based control synthesis utilizing RT/RG and it has nothing to do with the convenience of the IPN and LbPN applicability to DES with nondeterminism.

In case of the estimation of the LbPN marking, the *representative markings* and corresponding *representative marking graphs* are defined [12] in order to find the *consistent markings*. The consistent marking set consists of all markings that are reachable from initial marking by firing some sequences whose observation is a word (a sequence of transitions being consecutively fired). In [13] the PN reachability problem is analyzed by means of defining the basis marking. There, the set T of PN transitions is partitioned into two subsets – the explicit and implicit transitions. The subset of implicit transitions does not contain directed cycles. The reachability set obtained by firing of implicit transitions is created by a subset of reachable markings called basis makings. Consequently, the basis reachability graph (BRG) can be obtained by means of the efficient algorithm presented just in [13]. It is suitable for bounded PN and after an extension also for unbounded PN. For unbounded PN the newest approach to computing basis coverability graph (BCG) is presented in [10]. Such approaches help to reduce an enormity (hugeness) of RT/RG sometimes also exponentially – see [10].

To avoid the problem with the huge RT/RG at the DES control synthesis by means of PN, also the *theory of regions* – see e.g. [16, 8, 9, 1, 23, 17, 14] – can be used. Consider the labelled transition system given as a tuple $(S, \mathcal{L}, T_{\rightarrow})$, where S is a set of states, $\mathcal{L}$ is a set of labels and $T_{\rightarrow}$ is a set of labelled transitions. A region r of such transition system is a mapping assigning to each state $s \in S$ a number $\sigma(s)$ (natural number for P/T PN, binary number for some event structures) and to each transition label ℓ a number $\tau(\ell)$ such that consistency condition $\sigma(s') = \sigma(s) + \tau(\ell)$ holds whenever $(s, \ell, s') \in T_{\rightarrow}$. So, a region r corresponds to a place of the Petri net which we would like to associate with a given step transition system. Consequently, the PN structure is, of course, enriched in such a way. This makes possible to find a more favourable structure in comparison with the original one. However, this topic is out of the scope of this paper.

6 ILLUSTRATIVE EXAMPLE OF ANOTHER KIND OF NONDETERMINISM IN FMS

Let us introduce, exclusively for illustration, another situation in complicated FMS leading to the nondeterminism. It concerns the problems occurring at sharing resources in a plant during cooperation of its subsystems. In FMS such systems are named as Resource Allocation Systems (RAS).

There are frequently used the systems of simple sequential processes with resources (S^3PRs) in FMS. In such systems the part being produced uses only one copy of one resource at each processing step. Such systems create a subclass of a higher (upper) class S^*PRs [24, 7] where more copies of one resource are allowed. Consider the production system with the principal scheme given in Figure 22 performing the production routing with the scheme given in Figure 23.

Here, M1–M4 are four machine tools each using some of the four sets of cutting/surfacing tools h1–h4. Three robots R1–R3, being the crucial devices, serve

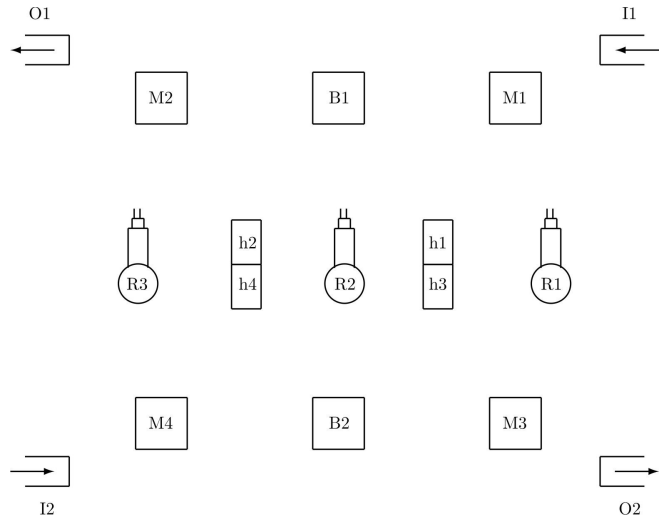

Figure 22. The scheme of the RAS plant

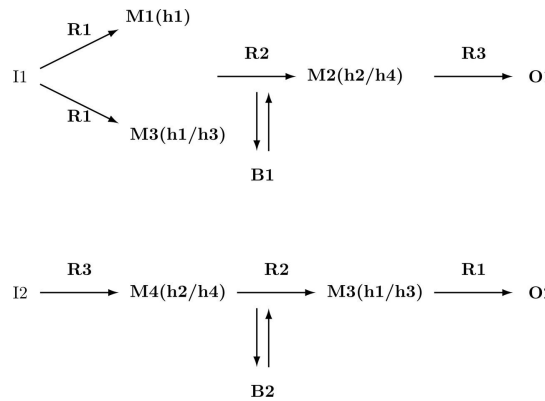

Figure 23. The production routing scheme of the RAS plant

these machines as well as buffers B1, B2, the input and output belts I1, I2 and O1, O2. The symbol Mi(hj/hk) in the routing scheme in Figure 23 means that the machine Mi uses the tools from the sets hj and hk, namely at first the tool from hj and then the tool from hk.

In the PN terminology, the resource(s) can be understood in a wider sense – e.g., in our case the instantaneous availability of a machine (because it can be either idle, i.e. available for an interested device, or not), cutting/surfacing tool, robot, buffer, etc.

The principle of PN model of a resource can be illustrated by means of Figure 24. In general, two or more production lines working as parallel processes sharing com-

mon resources, bring problems of different kinds to be solved, especially deadlocks because of a lack of resources in one or more of the processes. The simplest case of a resource is illustrated in Figure 24 left, while two simple parallel processes with common resource(s) are given in Figure 24 right. The parallel running of these processes is impossible when $n = 1$ while it is possible when $n > 1$.

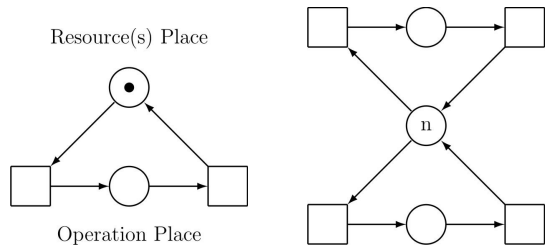

Figure 24. Resorce(s). In case of S^{3}PRs $n = 1$ (left). In case of S*PRs $n > 1$ (right). Here, two parallel processes share the common resource(s). When $n = 1$ the simultaneous using of the resource by the upper process and lower one is mutually excluded. However, for $n > 1$ both processes can run simultaneously.

The PN model of the RAS plant is displayed in Figure 25.

In general, the symbol $p_x \models$ Dy means that a place p_x models the activity of a device Dy. Then, in the PN model the following relations between PN places and the FMS devices are actual: $p_6 \models$ M1, $p_9 \models$ B1, $p_{10} \models$ M2, $p_{17} \models$ B2, $p_{19} \models$ M4, $p_{21} \models$ R1, $p_{22} \models$ h1, $p_{23} \models$ h3, $p_{24} \models$ M3, $p_{25} \models$ R2, $p_{26} \models$ h4, $p_{27} \models$ h2, $p_{28} \models$ R3.

This model corresponds to the situation when all resources are reliable. At the given initial state $\mathbf{x}_0 = (0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 4, 0, 2, 10, 4, 0, 2, 10, 1, 2, 2, 2, 0, 1, 2, 2, 1, 0)^T$ there exist 63 469 reachable states – e.g. the last of them is $\mathbf{x}_{63439} = (0, 0, 2, 1, 1, 1, 0, 0, 2, 0, 1, 3, 0, 2, 1, 3, 2, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2)^T$. For that reason, neither RT/RG nor the reachability matrix (containing state vectors being RT/RG nodes as its columns) cannot be introduced here.

In case of the unreliable resources the situation is more complicated. The PN model of such a case is given in Figure 27. Here, in comparison with Figure 25 five sub-nets represented by the dashed boxes are added. Namely, any unreliable resource has to be equipped by an additional place which represents waiting for the recovery of the resource failure. The scheme of such a sub-net is given in Figure 26. The dashed boxes in Figure 27 represent just such sub-nets. Of course, the model itself is not able to deal with such a nondeterminism. The recovery consists in the renewal of the state of resources.

As it follows from Figure 26 and especially from the dashed sub-nets in Figure 27, no transition having p_u as its input place (i.e. $t_9, t_{12}, t_{13}, t_{25}, t_{27}$) cannot be fired during the occurrence of the rest of the corresponding breakdown. Any supervisor being synthesized for such a PN model has to respect this rule. Namely, the token in p_w (i.e. p_2, p_6, p_{12}, p_{20}, p_{29}) cannot be returned to p_u (i.e. p_3, p_5, p_{13}, p_{16}, p_{28})

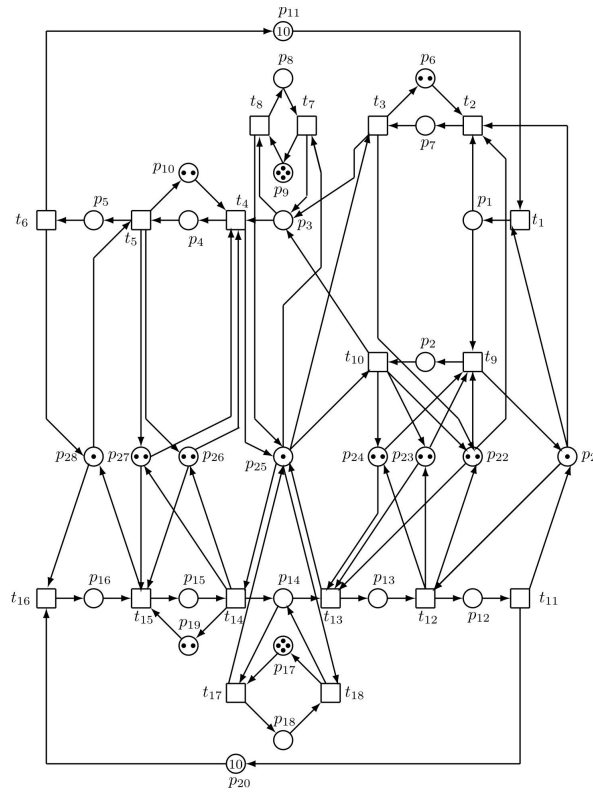

Figure 25. The PN model of the RAS plant with multiple resources equipped with the waiting places P_w, $w = 2, 6, 12, 20, 29$ waiting for the error recovery of unreliable resources

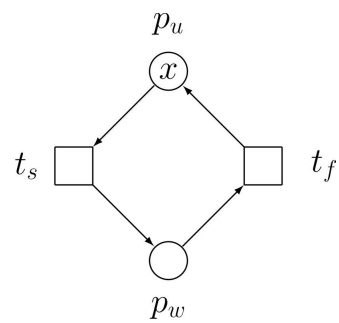

Figure 26. Unreliable resource x is represented by the place p_u, while the place p_w represents the process of waiting for the recovery. Transitions t_s and t_f represent, respectively, the start and finish of the waiting.

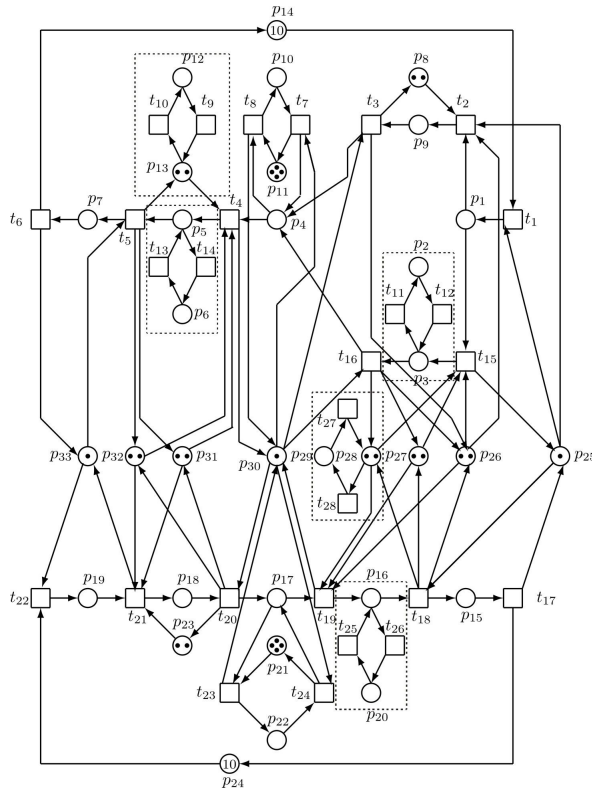

Figure 27. The PN model of the plant with multiple resources

until all faults related to p_u are corrected. The corrections have to be performed by error recovery subsystems.

Local Summary. The plant (FMS of a kind RAS) being the DES consists of a set of versatile resources i.e. machine tools, robots, buffers, cutting/surfacing tools, etc. Usually, there is a limited number of such resources shared by the plant sub-processes. Consequently, a lot of deadlocks can originate for this reason. Of course, deadlocks in FMS are undesirable, and highly unfavorable. Namely, the entire plant or at least a part of it remains stagnate and the primal intention of the production cannot be achieved. Such a situation can also be understood to be a form of nondeterminism, which is very unsafe. The recovery of resource failures is very complicated in this case. A fault-tolerant supervisor able to handle resource failures has to be used in order to resolve deadlocks, e.g. the Banker's algorithm [6] or its newer modifications. Another approaches to removal of deadlocks can be used as well – e.g. the supervisor based on PN siphons [25, 11, 19].

The error recovery of this kind of FMS (i.e. ARS) is out of the focus of this paper. However, dealing with such a problem may be an idea for further research in the DES containing nondeterminism.

7 CONCLUSION

This paper is a continuation of the paper [3]. It represents the Part 2 of the paper [3] being the Part 1. Both parts are inseparable components of one topic. While in [3] the different kinds of PN (especially P/T PN, timed PN, controlled PN, labelled PN, interpreted PN) were described and their applicability for DES modelling, analysis and control were indicated, the Part 2 brings more examples and case studies from PN-based modelling FMS and transport systems with nondeterminism.

The main aim of the article was to apply the results of article [3] particularly on more kinds of DES with the nondeterminism of different types. Some essential preliminaries were introduced at the beginning. Then, the particular topics were discussed. At first, the nondeterminism resulting from unobservable/uncontrollable transitions and/or unobservable (unmeasurable) places of PN-based models was dealt with. The ideal and real enforceability of control interferences were confronted in the example. After that, the problem of the error recovery was analyzed and two case studies on error recovery in real systems were introduced – namely, the case of the segment of the robotic cell of FMS and the segment of the transport system (the railroad crossing). Next, the case study on the error recovery of the complete robotic cell by means of IPN-based model was introduced. Afterward, the case study on the error recovery of another complex FMS by means of the LbPN-based model was presented. After all, the example of the specific kind of nondeterminism in the special kind of FMS – RAS (Resource Allocation Systems) was introduced for illustration.

The case studies and examples bear witness to applicability of PN in general, and especially IPN and LbPN, on dealing with nondeterminism in PN models of DES.

Because the local summaries were introduced at the end of the particular sections, they need not to be repeated here.

Acknowledgement

The author thanks for the partial support of the VEGA Agency (under Grant No. 2/0029/17).

REFERENCES

[1] Badouel, E.—Darondeau, P.: Theory of Regions. In: Reisig, W., Rozenberg, G. (Eds.): Lectures on Petri Nets I: Basic Models. Advanced Course on Petri Nets

(ACPN 1996). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 1491, 1998, pp. 529–586, doi: 10.1007/3-540-65306-6_22.

[2] Bellman, R. E.: Dynamic Programming. Princeton University Press, 1957. Republished 2003.

[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.: Failures in Discrete-Event Systems and Dealing with Them by Means of Petri Nets. Vietnam Journal of Computer Science. Vol. 5, 2018, No. 2, pp. 143–155, doi: 10.1007/s40595-018-0110-3.

[5] Cabasino, M. P.—Giua, A.—Seatzu, C.: Fault Detection for Discrete Event Systems Using Petri Nets with Unobservable Transitions. Automatica, Vol. 46, 2010, No. 9, pp. 1531–1539, doi: 10.1016/j.automatica.2010.06.013.

[6] Dijkstra, E. W.: Selected Writings on Computing: A Personal Perspective. Springer Verlag, 1982, doi: 10.1007/978-1-4612-5695-3.

[7] 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.

[8] Ghaffari, A.—Rezg, N.—Xie, X.: Net Transformation and Theory of Regions for Optimal Supervisory Control of Petri Nets. IFAC Proceedings Volumes, Vol. 35, 2002, No. 1, pp. 443–448, doi: 10.3182/20020721-6-es-1901.00076.

[9] Ghaffari, A.—Rezg, N.—Xie, X.: Design of a Live and Maximally Permissive Petri Net Controller Using the Theory of Regions. IEEE Transactions on Robotics and Automation, Vol. 19, 2003, No. 1, pp. 137–141, doi: 10.1109/tra.2002.807555.

[10] Lefaucheux, E.—Giua, A.—Seatzu, C.: Basis Coverability Graph for Partially Observable Petri Nets with Application to Diagnosability Analysis. In: Khomenko, V., Roux, O. (Eds.): Applications and Theory of Petri Nets and Concurrency (PETRI NETS 2018). Springer, Cham, Lecture Notes in Computer Science, Vol. 10877, 2018, pp. 164–183, doi: 10.1007/978-3-319-91268-4_9.

[11] 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.

[12] Ma, Z.—Tong, Y.—Li, Z.—Giua, A.: Marking Estimation in Labelled Petri Nets by the Representative Marking Graph. 20th IFAC World Congress, Toulouse, France, July 2017. IFAC PapersOnLine, Vol. 50, 2017, No. 1, pp. 11175–11181, doi: 10.1016/j.ifacol.2017.08.1240.

[13] Ma, Z. Y.—Tong, Y.—Li, Z.—Giua, A.: Basis Marking Representation of Petri Net Reachability Spaces and Its Application to the Reachability Problem. IEEE Transactions on Automatic Control, Vol. 62, 2017, No. 3, pp. 1078–1093, doi: 10.1109/TAC.2016.2574120.

[14] Meis, B.—Bergenthum, R.—Desel, J.: Synthesis of Elementary Net Systems with Final Configurations. Proceedings of the 16th International Conference on Application of Concurrency to System Design (ACSD 2016), Torun, Poland, 2016. CEUR Workshop Proceedings, Vol. 1592, 2016, pp. 47–57, available online at: http://ceur-ws.org/Vol-1592/paper04.pdf.

[15] 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.

[16] Mukund, M.: Petri Nets and Step Transition Systems. International Journal of Foundations of Computer Science, Vol. 3, 1992, No. 4, pp. 443–478, doi: 10.1142/s0129054192000231.

[17] Pan, Y.-L.—Huang, Y.-S.—Jeng, M.—Chung, S.-L.: Enhancement of an Efficient Control Policy for FMSs Using the Theory of Regions and Selective Siphon Method. The International Journal of Advanced Manufacturing Technology, Vol. 66, 2013, No. 9–12, pp. 1805–1815, doi: 10.1007/s00170-012-4460-1.

[18] Ramirez-Trevino, A.—Rivera-Rangel, I.—Lopez-Mellado, E.: Observability of Discrete Event Systems Modeled by Interpreted Petri Nets. IEEE Transactions on Robotics and Automation, Vol. 19, 2003, No. 4, pp. 557–565, doi: 10.1109/tra.2003.814503.

[19] Row, T.-C.—Pan, Y.-L.: Maximally Permissive Deadlock Prevention Policies for Flexible Manufacturing Systems Using Control Transition. Advances in Mechanical Engineering, Vol. 10, 2018, No. 7, pp. 1–10, doi: 10.1177/1687814018787406.

[20] Ru, Y.—Hadjicostis, C. N.: Fault Diagnosis in Discrete Event Systems Modeled by Partially Observed Petri Nets. Discrete Event Dynamic Systems, Vol. 19, 2009, No. 4, pp. 551–575, doi: 10.1007/s10626-009-0074-7.

[21] SPNBOX: A Toolbox for the Supervisory Control of Petri Nets. Available at: `http://mviordache.name/abs/spnbox/`.

[22] Tong, Y.—Li, Z.—Giua, A.: Observation Equivalence of Petri Net Generators. IFAC Proceedings Volumes, Vol. 47, 2014, No. 2, pp. 338–343, doi: 10.3182/20140514-3-fr-4046.00060.

[23] Uzam, M.: An Optimal Deadlock Prevention Policy for Flexible Manufacturing Systems Using Petri Net Models with Resources and the Theory of Regions. The International Journal of Advanced Manufacturing Technology, Vol. 19, 2002, No. 3, pp. 192–208, doi: 10.1007/s001700200014.

[24] Yue, H.—Xing, K.—Hu, H.—Wu, W.—Su, H.: 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.

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

[26] Zhou, M. C.—DiCesare, F.: Petri Net Synthesis for Discrete Event Control of Manufacturing Systems. Kluwer Academic Publishers, 1993, doi: 10.1007/978-1-4615-3126-5.

František Čapkovič received his Master degree in 1972 from the Faculty of Electrical Engineering of the Slovak Technical University, Bratislava, Slovakia. Since 1972 he has been working at 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 since 2001 till now he has worked at the Institute of Informatics. In 1980 he received the Ph.D. degree from SAS. In 1998 he was appointed Associate Professor. His interests are 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.