

# Interpreted Petri Nets in DES Control Synthesis

František Čapkovič<sup>(✉)</sup>

Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia

Frantisek.Capkovic@savba.sk

**Abstract.** Discrete event systems (DES) control based on interpreted Petri nets (IPN) is presented in this paper. While place/transition Petri nets (P/T PN) are usually used for modelling and control in case of controllable transitions and measurable places, the IPN-based models yield the possibility for the control synthesis also in case when P/T PN models contain some uncontrollable transitions and unmeasurable places. The creation of the IPN model from such a P/T PN model is introduced and the control synthesis is performed. The illustrative examples as well as the case study on a robotized assembly cell are introduced.

**Keywords:** Control · Discrete event systems · Modelling · Place/transition Petri nets · Interpreted Petri nets

## 1 Introduction

Petri nets (PN) [1, 2, 6–8] are a special kind of bipartite directed graphs. Namely, they have two kinds of nodes (places and transitions) and two kinds of edges (from places to transitions and conversely). The PN structure [1] is a triple  $N = \langle P, T, B \rangle$ , where  $P$  is a finite set of  $n$  places and  $T$  is a finite set of  $m$  transitions.  $B$  represents the PN edges. Thus,

$$P \cup T \neq \emptyset; \quad P \cap T = \emptyset \quad (1)$$

$$B \subseteq (P \times T) \cup (T \times P) \quad (2)$$

where  $B = F \cup G$  with  $F \subseteq (P \times T)$  and  $G \subseteq (T \times P)$ .

Except the structure PN have also their *dynamics*. Thus, the complete PN definition is  $PN = (N, M_0)$  with  $M_0$  being the initial marking (represented below in (3) by the initial state vector  $\mathbf{x}_0$ ).

A transition  $t \in T$  may be fired at a marking  $M$  if it is enabled - i.e. if all of its input places have at least one token. The firing of  $t$  removes one token from each of its input places and adds one token to each of its output places. After firing  $t$  a new PN marking  $M'$  is reached. This process is expressed below by the state Eq. (3). The set  $R(M_0)$  expresses all markings reachable from the initial marking  $M_0$ . The PN marking development can be understood to be PN *dynamics*, formally expressed [1] by the quadruplet  $\langle X, U, \delta, \mathbf{x}_0 \rangle$ . Here,

---

F. Čapkovič—Partially supported by the grant VEGA 2/0039/13.

$\delta : X \times U \rightarrow X$ , where  $X$  is a set of state vectors  $\mathbf{x}_k$  and  $U$  is a set of control vectors  $\mathbf{u}_k$ . While the entries of state vectors  $\mathbf{x}_k$  express states of elementary places  $\sigma_{p_i}^k$ ,  $i = 1, \dots, n$  (i.e. the number of tokens), in different steps  $k = 0, 1, \dots, K$ , the entries of control vectors  $\mathbf{u}_k$  represent the states of elementary transitions  $\gamma_{t_j}^k$ ,  $j = 1, \dots, m$  (i.e. their disabling or enabling). The sets  $B, F, G$  can be replaced by the incidence matrices  $\mathbf{B}, \mathbf{F}, \mathbf{G}$  of PN edges. Here,  $\mathbf{B} = \mathbf{G}^T - \mathbf{F}$ . Consequently, PN dynamics can be described by the restricted linear discrete integer system (all of its parameters and variables are non-negative integers) as follows

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{B} \cdot \mathbf{u}_k; \quad k = 0, 1, \dots, K \quad (3)$$

$$\mathbf{F} \cdot \mathbf{u}_k \leq \mathbf{x}_k \quad (4)$$

Here,  $\mathbf{x}_k = (\sigma_{p_1}^k, \dots, \sigma_{p_n}^k)^T$  with  $\sigma_{p_i} \in \{0, 1, \dots, \infty\}$ ,  $i = 1, \dots, n$ , and  $\mathbf{u}_k = (\gamma_{t_1}^k, \dots, \gamma_{t_m}^k)^T$  with  $\gamma_{t_j} \in \{0, 1\}$ ,  $j = 1, \dots, m$ .

The mathematical model (3) - (4) describes the PN marking development. It can be successfully used for modelling DES when all places are measurable and all transitions are controllable. However, in PN models of real systems unmeasurable places and uncontrollable transitions can occur. In such a case Interpreted PN (IPN), being an extension of PN, are applied. IPN were defined e.g. in [3-5, 9, 10] and in other sources.

### 1.1 Interpreted Petri Nets

In the sense of the definition introduced in [10] the IPN is the following sextuplet

$$Q = \langle (N, \mathbf{x}_0), \Sigma, \Phi, \lambda, \Psi, \varphi \rangle \quad (5)$$

where,

$PN = (N, \mathbf{x}_0)$  is the original PN;

$\Sigma = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$  is the input alphabet with  $\alpha_i$ ,  $i = 1, \dots, r$ , being the input symbols;

$\Phi = \{\delta_1, \delta_2, \dots, \delta_s\}$  is the output alphabet with  $\delta_i$ ,  $i = 1, \dots, s$ , being the output symbols;

$\lambda : T \rightarrow \Sigma \cup \{\varepsilon\}$  labels the transitions. Either an input symbol  $\alpha_i \in \Sigma$  or the internal event  $\varepsilon$  is assigned to each PN transition by this function. Thus, two sets of transitions arise - the set  $T_c$  of controllable transitions and the set  $T_u$  of uncontrollable transitions. Of course,  $T = T_c \cup T_u$ .

$\Psi : P \rightarrow \Phi \cup \{\varepsilon\}$  labels the places. Either an output symbol  $\delta_i \in \Phi$  or the null event  $\varepsilon$  is assigned to each PN place by this function. Thus, two sets of places arise - the set  $P_m$  of the measurable places and the set  $P_{nm}$  of unmeasurable places.  $P = P_m \cup P_{nm}$ .

$\varphi$  is the output function assigning the output vector  $\mathbf{y}_k = \varphi \cdot \mathbf{x}_k$  to the PN state vector  $\mathbf{x}_k$ . The entries of the output vector  $\mathbf{y}_k$  represent the states of measurable places.

More details about the functions are introduced in [10].

## 2 IPN in Modelling and Control

To indicate the importance of IPN models for DES control, let us introduce two illustrative examples concerning (i) the principle of creating the IPN model of DES at existence of uncontrollable transitions and unmeasurable places; (ii) the principle of its control.

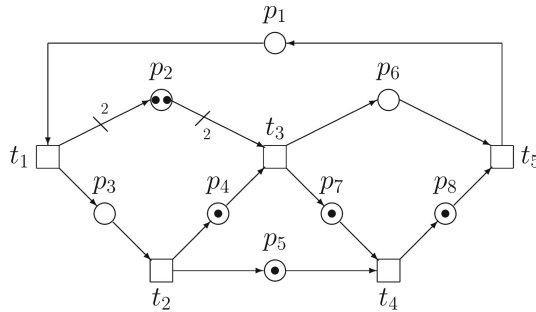


Fig. 1. An example of the PN-based model

To show how the IPN model can be created consider the simple PN-model given in Fig. 1. Suppose that the measurable places are  $P_m = \{p_1, p_5, p_6\}$  and the unmeasurable ones are  $P_{nm} = P \setminus P_m = \{p_2, p_3, p_4, p_7, p_8\}$ . Consider that the controllable transitions are  $T_c = \{t_1, t_5\}$ , while the uncontrollable ones are  $T_u = T \setminus T_c = \{t_2, t_3, t_4\}$ . Thus, for such IPN the input alphabet  $\Sigma = \{\alpha_1, \alpha_5\}$  and the output alphabet  $\Phi = \{\delta_1, \delta_2, \delta_3\}$ . Consequently,  $\lambda(t_k)_{k=1,\dots,5} = \{\alpha_1, \varepsilon, \varepsilon, \varepsilon, \alpha_5\}$ ,  $\Psi(p_i)_{i=1,\dots,8} = \{\delta_1, \varepsilon, \varepsilon, \varepsilon, \delta_2, \delta_3, \varepsilon, \varepsilon\}$ . The output equation is as follows

$$\mathbf{y}_k = \varphi \cdot \mathbf{x}_k \quad \text{where} \quad \varphi = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \quad (6)$$

Thus for the initial state  $\mathbf{x}_0 = (0, 2, 0, 1, 1, 0, 1, 1)^T$  the output vector is  $\mathbf{y}_0 = (0, 1, 0)^T$ .

In order to explain how the IPN model is controlled, consider a segment shown in Fig. 2 left. While the *upper line* containing  $p_4$  and  $t_3$  represents the fragment of the model of the control system  $PN_{cs}$  (containing the control specifications), the *lower line* represents the fragment of the IPN model of the controlled plant  $PN_{pl}$ . The RG of the model is given in Fig. 2 right, where  $\mathbf{x}_0 = (1, 0, 0, 1)^T$ ,  $\mathbf{x}_1 = (0, 1, 0, 1)^T$ ,  $\mathbf{x}_2 = (0, 0, 1, 1)^T$  and  $\mathbf{x}_3 = (0, 0, 1, 0)^T$ .

Here, the controllable transition  $t_1$  is enabled because it is required to reach the stated output while  $p_4$  represents the state of a sensor. The self-loop between them represents the relation between the place of the control specification and the plant controllable discrete event. The transition  $t_3$  represents enabling the event expressing the situation when the plant and control specification have the same output and  $p_3$  represents the state of the sensor. The self-loop between them expresses the relation between the plant measured place and the control specification. Note that the fragment of RG accordant with the segment of the controlled plant is straight-lined. Such fragments occur also in more complicated structures. Of course, RG of more complicated structures of the plant models will not be entirely straight-lined as in Fig. 2. Some branchings occur in such cases as well - it is apparent in Fig. 4 as well as in Fig. 6 introduced in the Sect. 3. However, they are not so extensive.

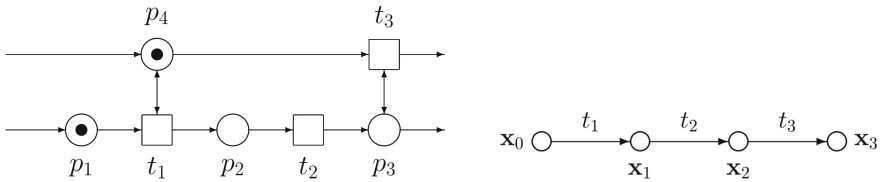


Fig. 2. The controlled segment of the IPN-based model (left) and its RG (right)

### 2.1 Illustrative Example

To illustrate the control process deeper, consider the more complicated configuration of the P/T PN model of a plant (without uncontrollable transitions and unmeasurable places) given in Fig. 3 left. Its reachability graph (RG) is in Fig. 3 right. The model parameters and the states reachable from the initial state (columns of  $\mathbf{X}_{reach}^{PN_{pl}}$ ) are Control of the system can be modelled as it is shown in Fig. 4 left where the model of the controlled plant together with the controller are introduced. Corresponding RG is given in Fig. 4 right. The model parameters and the reachable states are as follows

$$\mathbf{F}^{PN_{pl}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}; \mathbf{G}^{PN_{pl}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}; \mathbf{x}_0^{PN_{pl}} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$



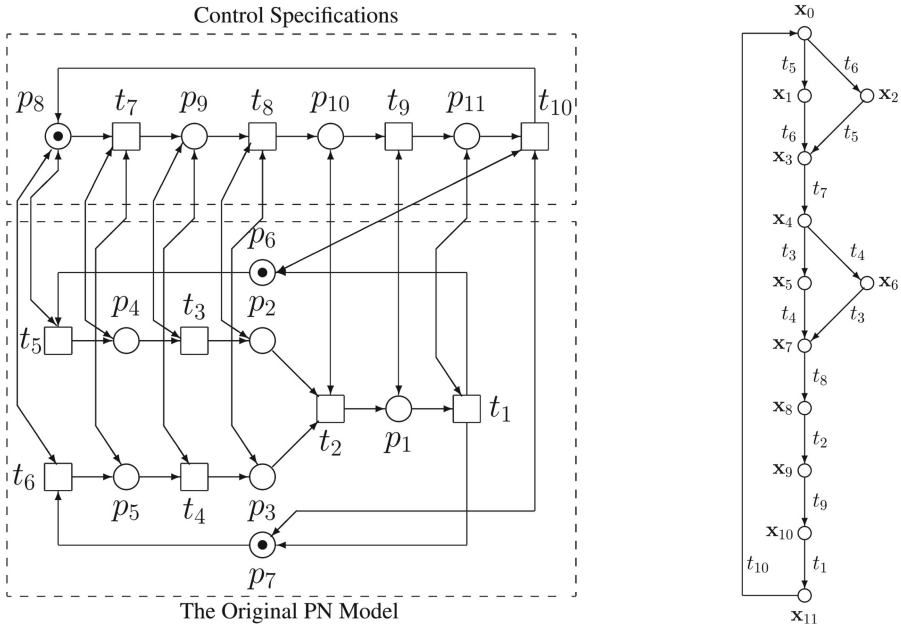


Fig. 4. The controlled PN-based model of the plant and its RG

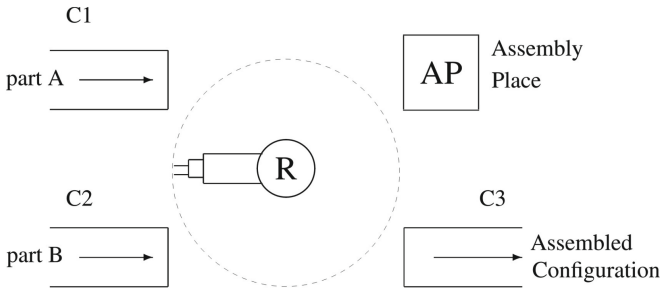
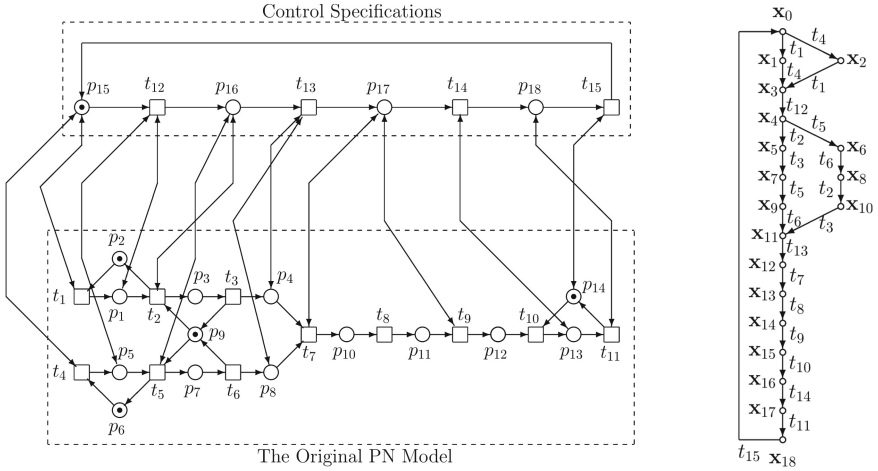


Fig. 5. The scheme of the robotized assembly cell

Comparing RGs introduced in Figs. 3 and 4 it is evident that the latter RG does not include complicated branching. Thus, the IPN-based approach can be successfully applied also on P/T PN without uncontrollable transitions and unmeasurable places.

### 3 Case Study on Robotized Assembly Cell

Consider the robotized assembly cell (RAC) displayed schematically in Fig. 5. The input conveyors  $C_1$  and  $C_2$  deliver to the RAC, respectively, parts A and parts B. The robot  $R$  takes the part A from  $C_1$  and inserts it into the assembly place



**Fig. 6.** The controlled IPN-based model of the plant (left) and its RG (right)

(AP). Then R takes the part B from C2 and inserts it into AP. In AP both parts are assembled into the final configuration. After finishing the assembly process, R takes the assembled configuration and put it on the output conveyor C3. Such the progress of work is repeated. The PN-based model of RAC is given in Fig. 6. The lower dashed box represents the PN model  $PN_{pl}$  of the plant, while the upper one expresses control specifications. The PN places represent the following activities:  $p_1$  - C1 conveys the part A;  $p_2$  - C1 is available;  $p_3$  - R takes A from C1 and transfers it to AP;  $p_4$  - R inserts A into AP;  $p_5$  - C2 conveys the part B;  $p_6$  - C2 is available;  $p_7$  - R takes B from C2 and transfers it to AP;  $p_8$  - R inserts B into AP;  $p_9$  - it ensures the mutual exclusion, because R cannot take A from C1 and B from C2 simultaneously;  $p_{10}$  - the parts A, B are assembled in AP;  $p_{11}$  - R unloads the finished configuration from AP;  $p_{12}$  - R transfers the finished configuration from AP to C3;  $p_{13}$  - R put the finished configuration on C3;  $p_{14}$  - the free place on C3 is available.

In this model the transition  $t_8$  is uncontrollable (i.e. PN turns to IPN). The RG of the uncontrolled PN model has 813 nodes and it is extensively branched. Consequently, it cannot be introduced here. When only one input part A and only one input part B are allowed the RG has 28 nodes. But it is also too big and branched.

### 3.1 Control of the Robotized Assembly Cell

Let us apply know the IPN-based approach to control the RAC with the uncontrollable transition  $t_8$ . The assembly process is spontaneous and it cannot be





$$\mathbf{x}_0 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ - \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} ; \mathbf{X}_{reach} = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ - & - \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

The first column of the reachability matrix  $\mathbf{X}_{reach}$  represents the initial state  $\mathbf{x}_0$  and other columns represent the states reachable from  $\mathbf{x}_0$  - i.e. the feasible states. As it can be seen from Fig. 6, the PN model of the plant itself has no feedback - it is straightforward. The working cycle is closed by means of the loop of the control specifications - see the upper dashed box in Fig. 6.

### 4 Conclusion

Two main problems were investigated in this paper. Namely, (i) how to replace the P/T PN model of DES (in our case the assembly manufacturing system) by the IPN model in case when some uncontrollable transitions and unmeasurable places occur in the plant model; (ii) the control synthesis of DES at using the IPN model.

After introducing PN and P/T PN, IPN were introduced. Then, the creation of the IPN models was explained. To illustrate these techniques in detail, the simple demonstrative examples were introduced. Simultaneously, the attention was paid also to RGs. While in case of the P/T PN models RGs are usually extensively branched and contain a big number of nodes, the proposed approach yields RGs with fewer nodes and simpler structure. It was shown that the approach proposed for the IPN control can be applied also on the P/T PN (without the uncontrollable transitions and unmeasurable places). Thus, the simpler RGs of the P/T PN models can be found. As it is evident from the comparison of RGs

displayed in Figs. 3 and 4, the latter RG does not include complicated branching - i.e. it is more straightforward. Consequently, the proposed IPN-based approach can be successfully applied also on P/T PN without uncontrollable transitions and unmeasurable places.

The main part of the paper concerns the case study on modelling and control of the robotized assembly cell. Here, the transition  $t_8$  is uncontrollable, because its firing cannot be realized till the moment when the parts A, B are assembled ( $p_{10}$ ). The assembly process cannot be influenced from outside, it depends only on the machine realizing the assembly. Namely, on its capability and efficiency. Moreover, some unexpected (accidental) influences may extend processing time of the assembly. Only, after finishing the assembly operation,  $t_8$  is fired and the robot can unload the finished configuration from the machine  $p_{11}$  and the working progress of the RAC can continue.

The presented approach to controlling DES modelled by means of IPN seems to be useful for practice. Moreover, to a certain extent it is also comparatively general. It is able to deal with straightforward PN models without any problems. However, it is necessary to say that at systems, where PN models contain internal cycles, some problems can occur. Therefore, in future it is necessary to pay attention to the investigation of such problems in detail.

**Acknowledgments.** The author thanks the Slovak Grant Agency VEGA for the partial support under grant # 2/0039/13.

## References

1. Čapkovič, F.: Agent based approach to modelling ATM network. In: Proceedings of the 2012 IEEE 6th International Conference on Intelligent Systems - IEEE IS 2012, Sofia, Bulgaria, 6–8 September 2012, vol. I, pp. 102–107. IEEE Press, Piscataway (2012)
2. Desel, J., Reisig, W.: Place/transition Petri nets. In: Reisig, W., Rozenberg, G. (eds.) APN 1998. LNCS, vol. 1491, pp. 122–173. Springer, Heidelberg (1998)
3. Dotoli, M., Fanti, M., Mangini, A.: On-line identification of discrete event systems by interpreted Petri nets. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics - SMC 2006, Taipei, Taiwan, 8–11 October 2006, vol. 4, pp. 3040–3045. IEEE Press, Piscataway (2006)
4. Karatkevich, A.: Dynamic Analysis of Petri Net-Based Discrete Systems. Lecture Notes in Control and Information Sciences, vol. 356. Springer, Heidelberg (2007)
5. Lutz-Ley, A., Mellado, E.L.: Synthesis of fault recovery sequences in a class of controlled discrete event systems modelled with Petri nets. In: 2013 Iberoamerican Conference on Electronics Engineering and Computer Science, Procedia Technology, vol. 7, pp. 257–264 (2013)
6. Murata, T.: Petri nets: properties, analysis and applications. Proc. IEEE **77**, 541–580 (1989)
7. Peterson, J.L.: Petri nets. Comput. Surv. **9**, 223–252 (1977)
8. Peterson, J.L.: Petri Net Theory and the Modeling of Systems. Prentice-Hall Inc., Englewood Cliffs (1981)

9. Rivera-Rangel, I., Ramírez-Treviño, A., Aguirre-Salas, L.I., Ruiz-León, L.I.: Geometrical characterization of observability in interpreted Petri nets. *Kybernetika* **41**, 553–574 (2005)
10. Santoyo-Sanchez, A., Pérez-Martínez, M.A., De Jesús-Velásquez, V., Aguirre-Salas, L.I., Alvarez-Ureña, M.A.: Modeling methodology for NPC's using interpreted Petri nets and feedback control. In: Proceedings of the 2010 7th International Conference on Electrical Engineering, Computing Science and Automatic Control - CCE 2010, Tuxtla Gutiérrez, Chiapas, Mexico, 8–10 September 2010. CD-ROM, pp. 369–374. IEEE Press (2010)