

# Representation of Management Processes in Socio-Economic Systems with Colored Petri Nets

Vladimir F. Tarasenko<sup>1</sup>, Pavel V. Senchenko<sup>1</sup>, Oleg I. Zhukovsky<sup>1</sup>, Yury B. Gritsenko<sup>1</sup>, Yury P. Ekhlakov<sup>1</sup>,

<sup>1</sup>Tomsk State University of Control Systems and Radioelectronics  
Tomsk, Russian Federation  
pvs@tusur.ru

**Abstract** — This paper analyzes the problems of management processes in socio-economic systems using colored Petri nets. The paper offers variants for formation of colored Petri nets. Relying on analysis of the properties of colored Petri nets, the paper offers net modeling algorithms based on direct and inverse language-preserving transformations of colored and classical Petri nets. The paper discusses the aspects and advantages offered by modeling of individual structures of management processes in socio-economic systems.

**Keywords** — modeling, dynamic models, colored Petri nets, process management, socio-economic systems.

## I. INTRODUCTION

In the majority of situations, representation of management processes in socio-economic systems reduces itself to modeling of business processes. Providing a comprehensive description of such processes requires a great number of models to be built using a variety of professional modeling languages. One of the most frequently used variants of a dynamic model is Petri nets (PN). Models of management processes in socio-economic systems are normally expected to be representative not only of their management structures, but also of their dynamic properties. Experience shows that classical (regular) PNs work well for modeling of simple systems. Enhanced modeling capacity is often achieved through extensions and modifications of common PN variants, for example, colored [1] Petri nets (CPN) [2, 3], E-nets [4], etc.

The authors propose the use of a CPN variant for description of the dynamic properties of structures of socio-economic process management, which offers a graphic representation of important information and have advantage over other CPN variants described in [1], where the model's information content is presented in special tables.

The paper analyzes the methods for CPN transformation that allow to preserve their dynamic properties (the modeling language). It demonstrates that transformation makes it possible to simplify the net structure while preserving the information content in CPN models of business processes.

## II. DEFINITIONS OF COLORED PETRI NETS

Modeling of business processes [5] in socio-economic systems using Petri nets requires the introduction of positions [1, 6] that are not images of process elements, but rather serve to regulate transition firings in the net. Thus, the formally

similar elements in the model have different meanings, requiring additional verbal description, complicating the spatial structure of the model, and making its interpretation difficult [7]. Based on the terms [6], we shall define and introduce the notion of generalized Petri nets which make it possible to simplify the spatial structure of the model.

**Definition 1.** A labeled (colored) Petri net (CPN)  $C$  is a quadruple:

$$C = (P, T, I, O), \quad (1)$$

where  $P = \{p_1, p_2, \dots, p_N\}$  is a finite set of positions,  $N \geq 0$ ;

$T = \{t_1, t_2, \dots, t_M\}$  is a finite set of transitions,  $M \geq 0$ ;  
 $P \cap T = \emptyset$ ;

$I: T \rightarrow P^\infty$  is the input mapping function for a set of transitions to labeled set of positions  $I = (I_1, I_2, \dots, I_L)$ ;

$O: T \rightarrow P^\infty$  is the output mapping function for a set of transitions to labeled set of positions;  
 $O = (O_1, O_2, \dots, O_L)$ ,  $L = |D|$ ;

$D = \{d_1, d_2, \dots, d_L\}$  is a set of labels (colors).

In  $|D|=1$  the definition of Petri nets is consistent with [6]. The cardinal number of set  $P$  is number  $N$ , and the cardinal number of set  $T$  is number  $M$ . An arbitrary element of  $T$  is denoted by symbol  $t_j$ , and an arbitrary element of  $P$ , by  $p_i$ ,  $i = \overline{1, N}$ . Position  $p_i$  is the input position for transition  $t_j$  if  $\exists l: l = \overline{1, L}$  is such that  $p_i \in I_l(t_j)$ . Position  $p_i$  is the output position for transition  $t_j$ , if  $\exists l: l = \overline{1, L}$  is such that  $p_i \in O_l(t_j)$ . Inputs and outputs of transitions together are the body of labeled positions.

Ratio of input position  $p_i \in I_l(t_j)$  from the  $l$ -th set of positions  $(P^\infty)^l$  for transition  $t_j$  is the number of times the position appears in the given transition set, and is denoted as  $\#(p_i, I^l(t_j))$ . Similarly  $\#(p_i, O^l(t_j))$  denotes the ratio of the output position in the  $l$ -th set.

Labeled Petri nets are best visualized through graphic representation.

**Definition 2.** Graph  $G$  of a labeled Petri net is a directed bipartite multigraph:

$G = (V, A)$ , where  $V = \{v_1, v_2, \dots, v_S\}$  is a set of nodes,  $S = |V|$ ;

$A = \{a_1, a_2, \dots, a_r\}$  is a set of labeled directed arcs;

$r = |A|$ ,  $a_i = (v_j, v_k, d_e)$ ,

where  $v_j, v_k \in V$ ,  $d_e \in D$ ,  $D = (d_1, d_2, \dots, d_L)$  is a set of labels,  $L = |D|$ .

Set  $V$  can be split into two non-overlapping subsets  $P$  and  $T$ , with the property that  $V = P \cup T$ ,  $P \cap T = \emptyset$ , and for each directed labeled arc  $a_i \in A$  it is true that: if  $a_i = (v_j, v_k, d_e)$ , then either  $v_j \in P$  and  $v_k \in T$ , or  $v_j \in T$  and  $v_k \in P$ .

Marking  $\mu$  means the allocation (belonging) of labeled tokens to the Petri net positions. The number, position and labeling (color) of tokens change with execution of the Petri net. Tokens are used to control the execution of Petri nets.

**Definition 3.** Marking  $\mu$  of a labeled Petri net  $C = (P, T, I, O)$  is a function that maps a set of positions into a set of vectors with nonnegative whole components.

**Definition 4.** Marking  $\mu$  is a matrix of the following size:

$$N \times L: \mu = \{\mu_{il}\},$$

where  $N = |P|$  and each  $\mu_{il}$  belongs to a set of nonnegative whole numbers,  $i = \overline{1, N}$ ,  $l = \overline{1, L}$ .

**Definition 5.** A marked labeled Petri net  $M = (C, \mu)$  is a combination of the structure of the labeled Petri net  $C = (P, T, I, O)$  and the marking  $\mu$ , and can be recorded as  $M = (P, T, I, O, \mu)$ .

A labeled Petri net is executed in accordance with the number, labels and distribution of tokens in the net. A Petri net is executed by firing of transitions. A transition is fired by removing tokens from its input positions and forming new tokens which are placed in its output positions. A transition can only be fired if it's enabled.

**Definition 6.** Transition  $t_j \in T$  in a marked labeled (colored) Petri net  $M = (P, T, I, O, \mu)$  with marking

$\mu = \{\mu_{il}\}$  is enabled if  $\forall p_i \in I^l(t_j)$  fulfills the condition:

$$\mu_{il} \geq \#(p_i, I^l(t_j)), l = \overline{1, L}.$$

A transition is fired by removing the enabling tokens from its input positions, and then placing one token for each arc in each output position. Firing a transition changes marking  $\mu$  of the Petri net to a new marking  $\mu'$ . If any transition position does not have sufficient, appropriately labeled tokens, the transition is not enabled and cannot be fired.

Transition  $t_j$  in a marked labeled Petri net with marking  $\mu = \{\mu_{il}\}$  can be fired whenever it is enabled. Firing of an

enabled transition  $t_j$  creates a new marking  $\mu' = \{\mu'_{il}\}$ , definable by the following correlation:

$$\mu'_{il} = \mu_{il} - \#(p_i, I^l(t_j)) + \#(p_i, O^l(t_j)), l = \overline{1, L}.$$

### III. APPLICATION OF COLORED PETRI NETS

Some publications [6, 8], when discussing the notion of Petri net languages, rely on the term *labeled* Petri nets, which also applies to transitions. In order to avoid any ambiguous interpretation of *labeled transitions* and *labeled tokens*, hereinafter we shall be discussing the coloring of arcs and tokens, which we will use in illustration of colored Petri nets.

It should be noted that in construction of nets, the arcs that come into transitions determine the colors of the corresponding enabling tokens in all input positions, and the output arcs of a transition serve as colorants for the tokens that will appear in output positions of the transition when it is fired, if such transition is enabled. A transition is enabled if the lineup and colors of tokens in output positions match the lineup and colors of arcs that run into the transition [7].

Figures 1-4 show an example of colored Petri nets used to optimize a model structure where the problem requires the introduction of redundant elements.

Firing of transition  $t_3$  in Fig. 1 is possible if there are two tokens in position  $P$ , which can appear in position  $P$  as a result of one of the following sequences of transition firings:  $t_1 t_1$ ;  $t_2 t_2$ ;  $t_1 t_2$ ;  $t_2 t_1$ .

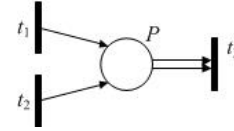


Fig. 1. Simulated simple transition firing, disregarding previous transition firings (single-position PN model).

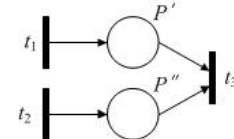


Fig. 2. Simulated simple transition firing, with consideration of previous transition firings (two-position PN model).

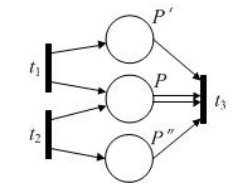


Fig. 3. Simulated simple transition firing, with consideration of previous transition firings (three-position PN model)

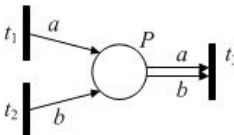


Fig. 4. Simulated simple transition firing, with consideration of previous transition firings (CPN model).

Position  $P$  indicates some modeled object. The request for transition firing  $t_3$  in variant  $t_1t_2$  or  $t_2t_1$  can be represented by models in Fig. 2 and 3. In the situation shown in Fig. 2, it is impossible to represent the modeled object with a single position, and we need to interpret various positions  $P_1$  and  $P_2$  as attributable to one and the same object. The situation shown in Fig. 3 requires the introduction of buffer positions  $P'_1$  and  $P'_2$  in the model. Furthermore, the formally similar elements of the model (positions) develop different interpretations.

In order to avoid the above problems and inconsistencies, CPN models should be a preferred option. In such case, the problem statement is fully resolved by the model itself (by adding colors  $a$  and  $b$ ) without the introduction of any additional elements (Fig. 4).

It is evident that using CPNs in system modeling makes it possible in some situations to simplify the spatial structure of CPN models and reduce the resources required to interpret them outside of the model. Accordingly, following the terms of [9], CPNs have a higher expressive power and a number of properties to be discussed further.

#### IV. KEY PROPERTIES OF COLORED PETRI NETS

Above we have discussed the definition of CPNs based on the introduction of multidimensional mapping functions  $I$  and  $O$  of the set of transitions  $T$  to  $P^\infty$  sets of positions. In real process modeling, PN models are most often expected to demonstrate one of the three properties – safety, restriction, persistence – or a combination thereof. Let us examine the same properties of CPNs [8].

**Definition 7.** For a colored Petri net  $C=(P,T,I,O)$  with the initial marking  $\mu_0$  position  $p_i \in P$  will be safe for color  $l$  ( $l=\overline{1,L}$ ), if for any marking  $\mu'$ , achievable from  $\mu_0$ , it is true that  $\mu^l(p_i)=1$ .

**Definition 8.** A CPN  $M=(C,\mu_0)$  is safe for color  $l$  ( $l=\overline{1,L}$ ), if each of its positions  $p_i \in P$  is safe for color  $l$ . Therefore, a CPN can be safe with respect to one of several colors, but not necessarily all colors, which opens new opportunities for interpretation and implementation of real system models.

**Definition 9.** For a CPN  $M=(C,\mu_0)$  position  $p_i \in P$  will be  $K$ -safe for color  $l$ , if for any marking  $\mu'$  achievable from  $\mu_0$ , it is true that  $\mu^l(p_i)=K$ . If in this case  $K' \geq K$ , then position  $p_i \in P$  is also  $K'$ -safe.

**Definition 10.** A CPN  $M=(C,\mu_0)$  is  $K$ -safe for color  $l$ , if each of its positions  $p_i \in P$  is  $K$ -safe for color  $l$ .

We shall say that a CPN that is  $K$ -safe for color  $l$  is also restricted with respect of the same color.

**Definition 11.** A Petri net  $M=(C,\mu_0)$  is persistent with respect to color  $l$ , if for any marking  $\mu'$  achievable from  $\mu_0$ , it

is true that  $\sum_{p_i \in P} \mu^l(p_i) = \sum_{p_i \in P} \mu'(p_i)$ .

To fulfill this condition it is necessary and sufficient that for  $\forall t_j \in T$ , which can be fired at least once, it is true that

$$\sum_{p_i \in P} \#(p_i, I^l(t_j)) = \sum_{p_i \in P} \#(p_i, O^l(t_j)), \quad (2)$$

i.e. the number of input and output positions from sets corresponding to color  $l$  for transition  $t_j$  should be equal. Otherwise, the number of tokens of that color would change after transition  $t_j$  is fired.

**Definition 12.** CPN  $M=(C,\mu_0)$  is safe if it is safe with respect to all  $l$  ( $l=\overline{1,L}$ ,  $L=|D|$ ,  $D$  is a set of colors or a dimension of functions  $I$  and  $O$ ). CPN  $M=(C,\mu_0)$  is restricted if it is restricted with respect to all  $l$ . CPN  $M=(C,\mu_0)$  is persistent if it is persistent with respect to all  $l$ .

A CPN can be restricted and persistent not necessarily for all colors. Therefore, it makes it possible to create process models with the property of persistence, while separating process model components from the components that affect the progress of the process, but not necessarily possess the property of persistence. That improves the models' visualization and accuracy to the real processes, which serves to support the importance of study of this class of CPNs.

Structural simplification (reduction of the number of positions) of a PN graph using colors, as demonstrated in this paper, is not the only effect of the CPN apparatus. The existence of language-equivalent regular PNs suggests that the approach is consistent and applicable when using CPNs. Moreover, it becomes possible, for the purpose of analysis of a specific class of systems, to minimize the effort associated with actual execution of the net. In such case, the dynamic properties of the system are studied through analysis of its structure. It offers the opportunity to give identical interpretation to identical elements of the model without any additional verbal description, which makes it possible to build models that provide a sufficient and accurate description of real processes in socio-economic systems.

A CPN can be restricted and persistent not necessarily for all colors. Therefore, it makes it possible to create process models with the property of persistence, while separating process model components from the components that affect the progress of the process, but not necessarily possess the property of persistence. That improves the models' visualization and accuracy to the real processes, which serves to support the importance of study of this class of CPNs.

It must be emphasized that the properties of safety, restriction and persistence of a CPN are significantly determined by the initial marking  $\mu_0$ , i.e. with a certain initial marking  $\mu_1 = \mu_0$  the properties of one and the same CPN model can show significant variation. For this reason, above, in discussion of the property of persistence with respect to any color, it is stipulated that condition (2) must be fulfilled for

such transitions  $t$  that can be fired at least once with the initial marking  $\mu_0$ . At the same time, there can exist transitions  $t$  that do not satisfy condition (2) and cannot be fired even once with such initial marking  $\mu_0$ . Nevertheless, such marked CPN will possess the property of persistence with respect to the color in question. Therefore, unless the activity of transition  $t$  is restricted, (2) shall be a sufficient yet not necessary condition of persistence.

The property of transition activity can be another valuable aspect in the analysis of CPN models. Five levels of transition activity (0–4) are identified for PN models, based on the possible number of transition firings: whether the transition can be fired at all (level 0); whether the transition can be fired at least once (level 1), whether the transition can be fired a specified number of times (level 2), whether the transition can be fired an infinite number of times (level 3), whether the transition can be fired an infinite number of times at any moment (level 4). The authors do not offer specific definitions of the levels of CPN transition activity, since, as is shown below in paragraph 5, any CPN model can be put in correspondence with only one PN model, possessing a more complex spatial structure of elements than the CPN model, but an equivalent set of transitions (it should be noted, that an reverse assumption of a sole corresponding CPN model is false (see paragraph 6)). Therefore, the definitions will differ only in the dimensions of the  $\mu$  function. Due to the same reason this paper does not address the matters relating to the CPN languages (a language for PNs and CPNs is defined as a set of enabled sequences of transition firings), since mutually corresponding CPN model and PN model will be equivalent in terms of their language, resulting, specifically, from the equivalence of their sets of transitions. All PN language matters are equally applicable to CPN languages. The properties of CPN transition activity and language are too significantly determined by the initial marking  $\mu_0$ .

Proceeding from the above, it follows that the problems of CPN analysis can be solved in the following order:

1) development of a CPN model of a real process (the assumption is that in a number of situations such model offers better visualization and accuracy than a PN model);

2) unambiguous transformation of the CPN model into a PN model following the algorithm described in paragraph 5;

3) analysis of the resulting PN model utilizing the conventional methods applicable to Petri nets.

It is possible, however, to restrict the class of CPN models used in order to attempt to solve the analysis problems directly using the CPN terms.

## V. METHOD OF COLORED PETRI NET TO REGULAR NET TRANSITION

Let a colored Petri net be represented by a quadruple  $C = (P, T, I, O)$  (see formula (1)).

The method for CPN to regular PN transition, as described below, is an iteration process, applied consecutively to each of positions  $p \in P$ , and consists in step-by-step formation of set

$P^\nabla$  and description of input  $I^\nabla$  and output  $O^\nabla$  functions of the corresponding PN, which we shall define as

$$C^\nabla = (P^\nabla, T^\nabla, I^\nabla, O^\nabla),$$

where  $T^\nabla$  is a set of transitions for  $C^\nabla$ , which coincide with the set of transitions for  $C$ , meaning that  $T^\nabla = T$ . The state of set  $P^\nabla$  at the  $i$ -th step will be  $P^\nabla[i]$ . The description of input function  $I^\nabla$  for transition  $t$  shall be designated as  $I^\nabla[i](t)$ , the description of input function  $O^\nabla$  for transition  $t$  shall be designated as  $O^\nabla[i](t)$ . The process can be described as a sequence of steps.

Step 0. Let  $i = 0$ ;  $P^\nabla[i] = \emptyset$ ;  $I^\nabla[i](t) = \emptyset$ ,  $O^\nabla[i](t) = \emptyset$ ,  $\forall t \in T^\nabla$ .

Step 1. Let  $i := i + 1$ , if  $i > N$ , then skip to step 4.

Step 2. Let  $t_{j1}, t_{j2}, \dots, t_{jk}, \dots, t_{jm}$  be a set of transitions for which  $p_i \in I^l(t_{ik})$  or  $p_i \in O^l(t_{ik})$ , i.e. for which position  $p_i$  is an input or output position, where  $m \leq M < \infty$ ,  $k = \overline{1, m}$ ,  $l = \overline{1, m}$ .

$$\text{Let } k_i = \sum_{l=1}^L \sigma_l(p_i),$$

$$\text{where } \sigma_l(p_i) = \begin{cases} 1, & \text{if } \exists l, \text{ such as } p_i \in I^l(t_{jk}), \\ 0, & \text{otherwise.} \end{cases}$$

In other words,  $k_i$  is a number of colors of the arcs incoming to and outgoing from position  $p_i$ . Then

$$P^\nabla[i] = \left( \bigcup_{j=1}^{k_i} p_{ij} \right) \cup P^\nabla[i-1], \text{ where position } p_{ij} \text{ is added to}$$

set  $P^\nabla$  at the  $i$ -th iteration. There is a total of  $k_i$  of such added positions. Index  $j$  corresponds to a certain color, but does not coincide with it if  $k_i \neq 1$ . If  $p_i$  contains a certain number of tokens of any color, i.e.  $\mu_i = n \neq 0$ , then in the corresponding position  $p_{ij}$  an equal number of regular tokens is placed, which can be designated as follows:

$$\mu^\nabla(p_{ij}) = n;$$

$$I^\nabla[i](t_j) = \left( \bigcup_{l=1}^L I^l(t_{ij}) \right) \cup I^\nabla[i-1](t_{ij});$$

$$O^\nabla[i](t_j) = \left( \bigcup_{l=1}^L O^l(t_{ij}) \right) \cup O^\nabla[i-1](t_{ij}).$$

This procedure is applied to all transitions  $t_{jl}$ , for which  $p_i \in I^l(t_{ijl})$  or  $p_i \in O^l(t_{ijl})$ .

Step 3. Skip to step 1.

Step 4. As a result,

$$(P^\nabla[N], T, I^\nabla[N], O^\nabla[N]) = (P^\nabla, T^\nabla, I^\nabla, O^\nabla) = C^\nabla,$$

where  $C^\nabla$  is a regular PN,  $|P^\nabla| = \sum_{i=1}^N k_i$ ,  $|T^\nabla| = M$ ,

$$|I^\nabla(t_j)| = \sum_{l=1}^L |I^l(t_j)|, \quad \forall t_j \in T \text{ and}$$

$$|O^\nabla(t_j)| = \sum_{l=1}^L |O^l(t_j)|, \quad \forall t_j \in T^\nabla.$$

This method is applicable to both marked and unmarked CPNs. A simplified example of application of the method is demonstrated on CPNs shown in Fig. 2 and Fig. 3.

The next step is to demonstrate the language equivalence of CPN  $C$  and PN  $C^\nabla$ , as it makes it possible to study the dynamic properties of  $C^\nabla$  and extend the findings of such study to  $C$ .

**Definition 13.** An elementary subnet

$$C_i = (p_i, \{t_j / t_j \in I^l(p_i) \vee t_j \in O^l(p_i)\}, I, O)$$

of Petri net  $C = (P, T, I, O)$  is the collection of any position  $p_i \in P$  of the net, all of its input and output transitions, i.e. transitions that satisfy  $t_j \in I^l(p_i)$  and  $t_j \in O^l(p_i)$ , and the input  $I$  and output  $O$  functions of such transitions.

The whole net is made of elementary subnets, which are such that  $\bigcup p_i = P$ ,  $\bigcap p_i = \emptyset$ .

The decoloration algorithm is applied consecutively to each position, i.e. to each elementary subnet  $C_i$ . An elementary subnet  $C_i$  has its own language  $\alpha(C_i)$ . It is evident that, if after decoloration an elementary subnet retains its language, the language of the whole net is retained as well.

**Theorem 1.** Decoloration (decomposition) of an elementary subnet of CPN  $C_i$  does not change the language of the subnet.

**Proof of theorem 1**

Let  $C_i$  be an elementary subnet with language  $\alpha(C_i)$ . Application of the decoloration algorithm results in an elementary subnet  $C_i^\nabla$  with language  $\alpha(C_i^\nabla)$ . It is necessary to demonstrate that  $\alpha(C_i) = \alpha(C_i^\nabla)$ .

A set of allowed sequences of transition firings (a language) is determined by the trigger conditions for each transition and enabling conditions for transition firing. Decoloration preserves the set of transitions of the elementary subnet. Decoloration does not affect the activity of input transitions  $t_j \in I^l(p_i)$  of position  $p_i$  of the elementary subnet, since their enabled status is determined by external factors. Thus, it only remains to be demonstrated that decoloration of an elementary subnet does not affect the enabled status of output transitions  $t_j \in O^l(p_i)$  of position  $p_i$ .

The enabled status of a transition is determined by its input function. As prescribed by the second step of the decoloration algorithm, the input function of any transition  $t_i$  of an elementary subnet is defined as

$$I^\nabla(t_j) = \left( \bigcup_{l=1}^L I^l(t_j) \right).$$

The total number of tokens required for the transition to fire and arcs incoming to the transition remains the same:

$$|I^\nabla(t_j)| = \sum_{l=1}^L |I^l(t_j)|, \quad \forall t_j \in O^l(p_j);$$

$$|\mu^\nabla(p_i)| = \sum_{l=1}^L |\mu^l(p_i)|,$$

i.e. the conditions for firing of transition  $t_i$  do not change with application of the decoloration algorithms, i. e. there is no change in language  $\alpha$  of the Petri net:  $\alpha(C_i) = \alpha(C_i^\nabla)$ .

The expression can be written for the whole net as  $\alpha(C) = \alpha(f(C))$ , where  $\alpha$  is the Petri net language,  $C$  is the colored Petri net,  $f$  is the decoloration algorithm.

Therefore, application of transformation  $f$  to the CPN makes it possible, abstracting from particular interpretations of the CPN, to analyze the PN  $f(C)$  relying on conventional methods. However, if  $C$  possesses the properties of partial safety, restriction, persistence with respect to a certain set of colors,  $f(C)$  may not possess the same properties for the PN.

## VI. METHOD FOR REGULAR NET TO COLORED PETRI NET TRANSITION

For the convenience of presentation, below is a discussion of the input and output functions of positions, rather than the input and output functions of transitions.

Let a Petri net be represented by a quadruple  $C = (P, T, I, O)$  (see formula (1)).

CPN  $C^\nabla$  shall be defined as  $(P, T, I^\nabla, O^\nabla)$ , where  $I^\nabla = (I^1, I^2, \dots, I^N)$ .

Function  $I^j$  is defined only for position  $p_i \in P$ ,  $I^j(p_i) = \{t_{j1}, t_{j2}, \dots, t_{jm}\}$ ,  $m \leq M$ ;  $O^\nabla = (O^1, O^2, \dots, O^N)$ .

Function  $O^i$  is defined only for position  $p_i \in P$ :

$$O^i(p_i) = \{t_{i1}, t_{i2}, \dots, t_{ik}\}, \quad k = M;$$

$\mu_i^\nabla(p_i) = \mu(p_i)$ ,  $|P| = N$ ;  $|T| = M$ ;  $|D| = L$ ;  $D$  is a set of colors.

CPN  $C^{\nabla\nabla}$  shall be defined as  $(P^\nabla, T, I^\nabla, O^\nabla)$ , where  $P^\nabla = \{p\}$ ,  $|P^\nabla| = 1$ . A graph of this CPN consists of a set of transitions  $T$ , which coincides with the set of transitions of the original PN (Fig. 5.1), and one position  $p$ , with colored arcs incoming to and outgoing from it, linking position  $p$  to transitions (Fig. 5.2). Since each position  $p_i$  in the original PN is put in correspondence with a unique color, then there are no multiple colored arcs in  $C^{\nabla\nabla}$  if there were no multiple arcs in the original PN. The equivalence problem of such CPN can be regarded as the equivalence problem (up to the symbols) of

input and output functions, which can also be useful for PN comparison.

Following the reasoning presented in paragraph 5, in this case the language of Petri nets is also preserved:  $\alpha(C) = \alpha(C^\nabla) = \alpha(C^{\nabla\nabla})$ .

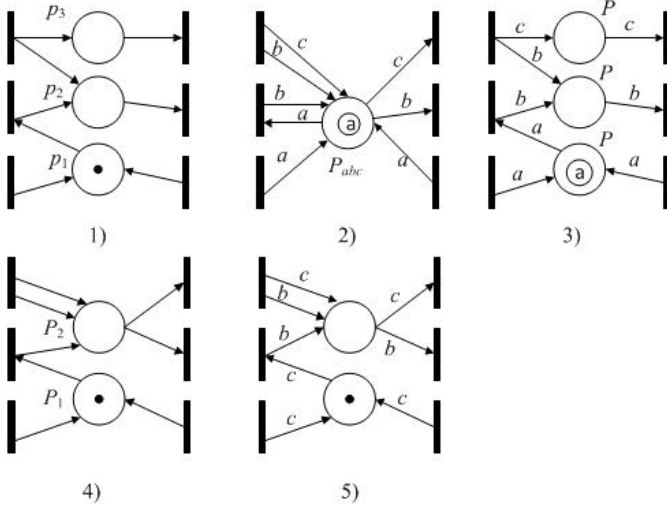


Fig. 5. Equivalence of PNs and CPNs 1) original Petri net; 2) CPN with a single position, equivalent to the original Petri net; 3) CPN with three positions, equivalent to the original Petri net; 4) PN non-equivalent to the CPNs; 5) CPN with two positions, equivalent to the original Petri net.

If the definition of  $C^\nabla$  does not place the requirement that functions  $I^l(p_i)$  и  $O^l(p_i)$  should be defined only for  $p_i = p$ , then graph  $C^{\nabla\nabla}$  can have multiple colored arcs (Fig. 5.3) when color  $b$  coincides with color  $c$ . Then, upon application of the method for CPN to PN transition (paragraph 5), the CPN shown in Fig. 5.2 will correspond to the PN shown in Fig. 5.4. In order to avoid that, it is necessary to put identical colors in correspondence with the positions that have no intersecting sets of input transitions and intersecting sets of output positions. An alternative methods consists in such redefinition of  $P^\nabla$  for  $C^{\nabla\nabla}$ , that  $|P^\nabla| \neq 1$  and, if  $p_i$  and  $p_j (j \neq i)$  belong to  $P^\nabla$ , then their output (input) sets of transitions corresponding to one color do not intersect (Fig. 5.5). It should be noted, however, that input and output sets can intersect among themselves.

Therefore, it follows from the above that different CPNs can have the same structure at the lower level. Specific colors, their combinations of the CPN structure convey the information in models of specific real systems or processes.

These methods can be applied to CPNs and PNs with restricting arcs without any adjustments or additions. An interesting factor that should be noted is that the ratio of a restricting arc in Petri nets cannot exceed 1, while in colored Petri nets the number of restricting arcs linking the position to the transition can be more than 2, but their colors will not coincide.

## VII. ELEMENTS OF CPN-BASED CONTROL STRUCTURE MODELS

PN-based modeling is discussed, among others, in [10, 11]. Structural simplification (reduction of the number of positions) of a PN graph using colors (CPN), as discussed in paragraph 6, is not the only effect of the CPN apparatus. The existence of language-equivalent regular PNs suggests that the approach is consistent when using CPNs. Moreover, it becomes possible, for the purpose of analysis of a specific class of systems, to move away from actual execution of the net, which is a requirement in development of the accessibility tree or in recording of matrix equations. In this case, the dynamic properties of the system are studied through analysis of this structure. It offers the opportunity to give identical interpretation to identical elements of the model without any additional verbal description, which makes it possible to build models that provide a fuller and more accurate description of real processes.

Models of business processes are often developed using the terms of the PN and CPN apparatus. Such models make it possible to describe not only the structure of the process, but its dynamics as well. A natural requirement is that formally identical elements of the model should correspond to the identical elements of the real system. In our interpretation, positions correspond to the conditions, the fulfillment of which results in selection of certain actions which will then be performed.

In case of complex interaction between elements, additional positions need to be introduced in the PN model in order to reflect the subtle aspects that have to be taken into account in process management. Apart from hindering interpretation of the model, it complicates its spatial structure, negatively affecting the very process of model development, as the developers will need to solve the problems associated with graphical representation of the model. In order to eliminate these shortcomings, we have developed and studied one of PN expansions – CPNs. Here models are developed using the terms of CPNs from the start. A model of a business process developed in this way allows the developer to select abstract types of sub-processes which are then specified. A combination of such types is a specific implementation of the process execution rules, and together with the content of work stages is the model of the business process.

It should be kept in mind, however, that we only deal with a model of the process. If we wish to discuss the activity of the net modeled, then in order to simplify its graphic representation, constructions that are similar with respect to the process structure can be represented in the model only once. If a CPN model of a business process structure is completed as a whole, it is possible to define transitions within the model, which will fire automatically as the net is executed, and transitions that must be fired from outside. As a general rule, the latter situations make it possible to resolve any contentions in the net. Such contentions are resolved through management of net execution, which ultimately is model management.

Let us consider the most frequently used models of

substructures used to build general models of systems. A simplest example is a degenerated structure (Fig. 6).

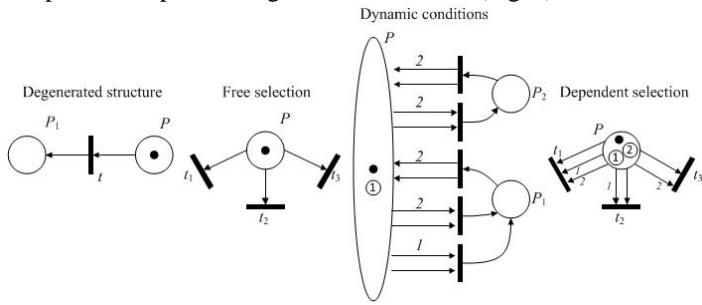


Fig. 6. Models of some management substructures

Transition  $t$  is the only output transition of position  $P$  and is fired upon fulfillment of the condition modeled by position  $P$ . The transition  $t$  firing event is registered in position  $P_1$ . If further progress of the process requires a decision to be made, then the firing of transition  $t$  can be interpreted as the decision-making procedure.

Fig. 6 also shows some more complex structures where transition enabling conditions can change dynamically in the process of net execution.

The PN and CPN apparatus makes it possible to develop models describing the system with a varying degree of detail. Every position and/or transition can be decompiled and represented as a subnet modeling the subprocess.

## VIII. CONCLUSION

Analysis of models of socio-economic systems involves a broad spectrum of analysts having a variety of mathematical backgrounds. That is why one of the important criteria considered in selection of a business process model is its illustrative power. And that makes the Petri net apparatus particularly compelling for presentation of management processes in socio-economic systems.

Structural simplification (reduction of the number of positions) of a PN graph using colors, as demonstrated in this paper, is not the only effect of the CPN apparatus. The existence of language-equivalent regular PNs suggests that the approach is consistent and applicable when using CPNs. Moreover, it becomes possible, for the purpose of analysis of a specific class of systems, to minimize the effort associated with actual execution of the net. In such case, the dynamic properties of the system are studied through analysis of its structure. It offers the opportunity to give identical interpretation to identical elements of the model without any additional verbal description, which makes it possible to build models that provide a sufficient and accurate description of real processes in socio-economic systems.

This article is designed as part of the national project of the Ministry of Education and Science of the Russian Federation № 3653 “Models, algorithms and software to support decision-making in risk management in the socioeconomic and industrial-technological system”.

## REFERENCES

- [1] V.E. Kotov, “Petri Nets”, Moscow: Nauka, 1984. (in Russian)
- [2] W. Reisig, “Petri Nets: An Introduction”, EATCS Monographs on Theoretical Computer Science, Springer-Verlag, Vol. 4, 1985.
- [3] I.M. Makarov, V.M. Nazaretov, A.V. Kulba and A.R. Shvetsov, “Petri Nets with colored markers”, Moscow: Tekhnicheskaya Kibernetika, 1987, Issue 6, p. 101-107. (in Russian)
- [4] G.J. Nutt, “Evaluation nets for computer system performance analysis”, Fall Joint Computer Conf, 1972, p.279-286.
- [5] V. A. Silich and M.P. Silich, “Business process reengineering”, Tomsk State University of Control Systems and Radioelectronics, 2007. (in Russian)
- [6] J. Peterson “Theory of Petri nets and simulation of systems”, Moscow, Mir, 1984. (in Russian)
- [7] Yu. P. Ekhlakov, V. F. Tarasenko, O. I. Zhukovsky, P. V. Senchenko and Yu. B. Gritenko “Color Petri nets in simulation of social and economic systems”, Reports of TUSUR, vol. 29, issue 3, 2013. pp. 83-92. (In Russian)
- [8] Yu. P. Ekhlakov, O. I. Zhukovsky, P. V. Senchenko and V. F. Tarasenko, “Petri nets in business process modeling. Theoretical bases and applications”, Tomsk: Tomsk State University of Control Systems and Radioelectronics, 2007. (in Russian)
- [9] B. Ya. Sovetov, O. I. Kutuzov, Yu. A. Golovin, and Yu. V. Avetov, “Application of microprocessor means in information transmission systems, Moscow: Vysshaya Shkola Publishers, 1987. (In Russian)
- [10] N. F. Shakirova “Simulation by Petri nets of behavior of players in the financial market”, Moscow: MFTI, 2006, Available: <http://zhurnal.ape.relarn.ru/articles/2002/185.pdf>. (In Russian)
- [11] A. Mehrez, M. Muzumdar, W. Acar and G. Weinroth “Petri Net Model View of Decision-Making: an Operational Management Analysis”, OMEGA The International Journal of Management Science, vol.23, Number 1, 1995, pp. 63-79.