

## Research Article

# Automation of Presentation Record Production Based on Rich-Media Technology Using SNT Petri Nets Theory

**Ivo Martiník**

*Faculty of Economics, VŠB-Technical University of Ostrava, Sokolská třída 33, 701 21 Ostrava 1, Czech Republic*

Correspondence should be addressed to Ivo Martiník; [ivo.martinik@vsb.cz](mailto:ivo.martinik@vsb.cz)

Received 5 December 2014; Accepted 20 March 2015

Academic Editor: Andrzej Materka

Copyright © 2015 Ivo Martiník. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Rich-media describes a broad range of digital interactive media that is increasingly used in the Internet and also in the support of education. Last year, a special pilot audiovisual lecture room was built as a part of the MERLINGO (*MEdia-rich Repository of LearnING Objects*) project solution. It contains all the elements of the modern lecture room determined for the implementation of presentation recordings based on the rich-media technologies and their publication online or on-demand featuring the access of all its elements in the automated mode including automatic editing. Property-preserving Petri net process algebras (PPPA) were designed for the specification and verification of the Petri net processes. PPPA does not need to verify the composition of the Petri net processes because all their algebraic operators preserve the specified set of the properties. These original PPPA are significantly generalized for the newly introduced class of the SNT Petri process and agent nets in this paper. The PLACE-SUBST and ASYNC-PROC algebraic operators are defined for this class of Petri nets and their chosen properties are proved. The SNT Petri process and agent nets theory were significantly applied at the design, verification, and implementation of the programming system ensuring the pilot audiovisual lecture room functionality.

## 1. Introduction

The term rich-media describes a broad range of digital interactive media, through which it is possible to share and transfer information and communicate in various ways. Moreover, rich-media enable interactivity, that is, bidirectional communication. The characteristic feature of the rich-media technologies is their accessibility online or on-demand, followed by the support of the dynamics of changes. An example can be online streaming video reporting, which is updated during broadcast, or a record of presentation placed on a website jointly with the synchronized slide show, which the user can interactively work with. Currently, there are several theories dealing with various aspects of the rich-media implementation, such as *Media Richness Theory* [1], *Media Naturalness Theory* [2], and *Social Presence Theory* [3].

The issue of majority aspects in implementing rich-media technologies at selected universities in the Czech Republic is dealt with in the MERLINGO project (*MEdia-rich Repository of LearnING Objects*) [4]. One of the key objectives of this project last year was the implementation of the cutting

edge rich-media technologies in order to build a specialized audiovisual lecture room containing all the elements of a modern lecture room determined for the implementation of presentation recordings and their publication online or on-demand featuring the integrated ergonomic and intuitive control and access in the automated mode of all its elements including automatic editing. Next objective was to provide a possibility of parallel sharing and feedback of lectures presented in the lecture room outfitted as such in real time even at other audiovisual auditoriums while providing the services of the central repository of educational objects MERLINGO and the possibility of their adaptation for the students with special needs. The implementation of those functionalities significantly contributed in decrease of costs required for the operation of high-capacity lecture halls including reduction of time and space demands mainly required for teaching subjects determined for large groups of students. Currently, the operation of the central repository MERLINGO dramatically simplifies the implementation of the new revolutionary technologies *EduArt* [5] and *MediaInTouch* [6], featuring unique characteristics in this area complying with demanding

requirements of teachers in the availability and quality of presentation recordings.

Petri nets [7–14] represent a popular formalism connecting advantages of the graphic representation of a modeled system with the possibilities of its simulation and the formal analyzability. Property-preserving Petri net process algebras (PPPA) [15] were originally designed for the specification and verification of manufacturing systems. PPPA also follows many ideas that can be originally seen in the class of workflow nets [16]. The elements of PPPA are the Petri net processes, ordinary connected Petri nets with a unique entry place, a unique exit place, and a set of places for handling resource sharing with their initial marking allowed. Among other features, PPPA does not need to verify composite components because all its operators preserve many properties. Hence, if the primitive modules satisfy the desirable properties, each of the composite components, including the system itself, also satisfies these properties.

PPPA has five types of operators: extensions, compositions, refinements, reductions, and place-merging. All the operators can preserve about twenty properties (some under additional conditions), such as liveness, boundedness, reversibility, traps, siphons, and proper termination.

The main goals of this paper are to generalize PPPA for the newly introduced class of the SNT Petri process and agent nets (SNTPAN) and to define the new SNTPAN operators called PLACE-SUBST (*PLACE-SUBSTitition*) and ASYNC-PROC (*ASYNChronous-PROcessing*) with the property of the SNTPAN soundness preservation. This class of SNTPANs with the support of the PLACE-SUBST and ASYNC-PROC operators that preserve the soundness property during the SNTPAN components composition process has been applied at the design, verification, simulation, and implementation phases of the software support of the audiovisual lecture room enabling automated recordings of the presentations including automated editing.

## 2. Materials and Methods

### 2.1. SNT Petri Process and Agent Nets and Their Properties.

Let  $\mathbf{N}$  denote the set of all natural numbers,  $\mathbf{N}_0$  the set of all nonnegative integer numbers,  $\emptyset$  the empty set,  $\neg$  the logical negation operator,  $|A|$  the cardinality of the given set  $A$ , where  $|\mathbf{N}| = \aleph_0$ ,  $\mathbf{P}(A)$  the family of all the subsets of a given set  $A$ , and IDENT the set of all identifiers over a certain alphabet.

*Multiset*  $M$  over a nonempty set  $S$  is a function  $m : S \rightarrow \mathbf{N}_0$ . The nonnegative number  $m(a) \in \mathbf{N}_0$ , where  $a \in S$ , denotes the number of occurrences of the element  $a$  in the multiset  $m$ . The multiset  $m$  is usually represented by the formal sum  $\sum_{a \in S} m(a)'a$ .  $S_{MS}$  denotes the set of all nonempty multisets over the set  $S$  and  $S_{MSE}$  denotes the set of all multisets over the set  $S$ .

Let  $A$  be a nonempty set. By the (nonempty finite) *sequence*  $\sigma$  over the set  $A$  a function  $\sigma$  can be understood,  $\sigma : \{1, 2, \dots, n\} \rightarrow A$ , where  $n \in \mathbf{N}$ . The function  $\varepsilon : \emptyset \rightarrow A$  is called the *empty sequence* over the set  $A$ . The sequence  $\sigma : \{1, 2, \dots, n\} \rightarrow A$  is usually represented by the notation  $\sigma = \langle a_1, a_2, \dots, a_n \rangle$  or  $\sigma = a_1 a_2 \cdots a_n$  of the elements of the set  $A$ , where  $a_i = \sigma(i)$  for  $1 \leq i \leq n$ . Empty sequence

$\varepsilon : \emptyset \rightarrow A$  over the set  $A$  is usually represented by the notation  $\varepsilon = \langle \rangle$ . The set of all *finite nonempty* sequences over the set  $A$  is denoted by the notation  $A_{SQ}$  and the set of all *finite* (and possible empty) sequences over the set  $A$  by the notation  $A_{SQE}$ . The *length of the sequence*  $\sigma = \langle a_1, a_2, \dots, a_n \rangle$ , where  $n \in \mathbf{N}$ , is equal to the natural number  $n$  and the length of the empty sequence  $\varepsilon$  is equal to the number 0. The length of the sequence  $\sigma$  is represented by the notation  $\text{length}(\sigma)$ . The function  $\text{elem} : A_{SQE} \rightarrow \mathbf{P}(A)$ , such that  $\text{elem}(\sigma) = \{a \mid \exists i : a = \sigma(i), \text{ where } 1 \leq i \leq \text{length}(\sigma)\}$ ,  $\text{elem}(\varepsilon) = \emptyset$ , assigns to each sequence  $\sigma \in A_{SQE}$  the set of all of its *elements*.

Let  $A = \{a_1, a_2, \dots, a_n\}$ , where  $n \in \mathbf{N}$ , be a finite nonempty set. *Vector*  $V$  is a function  $V : A \rightarrow \mathbf{N}_0$ . We will denote a vector  $V$  by the statement  $V = (V(1), V(2), \dots, V(n))$ .

Let  $\mathbf{V} = \{(V(1), V(2), \dots, V(n)) \mid V(i) \in \mathbf{N}_0 \text{ for } 1 \leq i \leq n\}$  and  $\mathbf{W} = \{(W(1), W(2), \dots, W(m)) \mid W(j) \in \mathbf{N}_0 \text{ for } 1 \leq j \leq m\}$ . Then

$$\begin{aligned} \mathbf{V} \otimes \mathbf{W} &= \{(V(1), \dots, V(n), W(1), \dots, W(m)) \mid V(i) \\ &\in \mathbf{N}_0 \text{ for } 1 \leq i \leq n, W(j) \in \mathbf{N}_0 \text{ for } 1 \leq j \leq m\}, \\ &\quad \text{if } (\mathbf{V} \neq \emptyset) \wedge (\mathbf{W} \neq \emptyset), \\ &= \mathbf{V}, \quad \text{if } (\mathbf{V} \neq \emptyset) \wedge (\mathbf{W} = \emptyset), \\ &= \mathbf{W}, \quad \text{if } (\mathbf{V} = \emptyset) \wedge (\mathbf{W} \neq \emptyset), \\ &= \emptyset, \quad \text{if } (\mathbf{V} = \emptyset) \wedge (\mathbf{W} = \emptyset). \end{aligned} \quad (1)$$

It is clear that the operation  $\otimes$  is associative (i.e.,  $(\mathbf{V} \otimes \mathbf{W}) \otimes \mathbf{U} = \mathbf{V} \otimes (\mathbf{W} \otimes \mathbf{U})$ ).

A *semigroup*  $(S, \circ)$  is a set  $S$  together with an associative binary operation  $\circ : S \times S \rightarrow S$ . A *monoid*  $(M, \circ, u)$  is a semigroup  $(M, \circ)$  with an element  $u \in M$  such that  $\forall x \in M : u \circ x = x = x \circ u$ .

A *net* is an ordered triple  $NET = (P, T, A)$ , where

- (i)  $P$  is a finite nonempty set of *places*,
- (ii)  $T$  is a finite set of *transitions*,  $P \cap T = \emptyset$ ,
- (iii)  $A$  is a finite set of *arcs*,  $A \subseteq (P \times T) \cup (T \times P)$ .

A given net is then described with a bipartite graph containing a finite nonempty set  $P$  of places used for expressing conditions of a modeled system (we usually use circles for their representation), a finite set  $T$  of transitions describing changes in the modeled system (we usually draw them in the form of rectangles) and a finite set  $A$  of arcs being principally oriented while connecting a place with a transition or a transition with a place and we usually draw them as lines with arrows. We say then a net  $(P, T, A)$  is *connected* iff it is not composed of two or more disjoint and nonempty nets.

Some commonly used notations for the nets are  $\bullet y = \{x \mid (x, y) \in A\}$  for the *preset* and  $y \bullet = \{x \mid (y, x) \in A\}$  for the *postset* of a net element  $y$  (i.e., place or transition).

A *SNT net* (*single-number tokens net*) is an ordered 5-tuple  $SNTN = (P, T, A, AF, TP)$ , where

- (i)  $(P, T, A)$  is a connected net,  $\forall p \in P : \succ p \subseteq \succ p$ ,

- (ii)  $AF$  is the arc function,  $AF : (P \times T) \cup (T \times P) \rightarrow (\mathbf{N}_0)_{SQE}$ ,  $AF(x, y) \in (\mathbf{N}_0)_{SQ}$  iff  $(x, y) \in A$ ,  $AF(x, y) = \langle \rangle$  iff  $(x, y) \notin A$ , where  $x, y \in P \cup T$ ,  
 $\forall t \in T \exists k \in \mathbf{N} \forall p \in (\bullet t \cup t\bullet)$ :  $\text{length}(AF(p, t)) = \text{length}(AF(t, p)) = k$ ,
- (iii)  $TP$  is the transition priority function,  $TP : T \rightarrow \mathbf{N}$ .

The arc function  $AF$  of a given SNT net assigns each arc with a (nonempty) sequence of the nonnegative integer numbers expressing a removed or added token from or to the place associated with that arc when executing a particular transition. There must be the only one length  $k$  ( $k \in \mathbf{N}$ ) of all the values of the arc function  $AF$  (i.e., for the sequences of nonnegative integer numbers) for the given transition  $t$  and all of its associated input and output arcs. Transition priority function  $TP$  of a given SNT net assigns with each transition the natural number value expressing its priority (transition priority function has the default value of 1 if it is not explicitly indicated in the SNT net diagram).

Let  $SNTN = (P, T, A, AF, TP)$  be a SNT net and  $p \in P$  a place. We will define the sets  $\triangleright p \in \mathbf{P}(\mathbf{N})$  of the *place input tokens* and  $\triangleright p \in \mathbf{P}(\mathbf{N})$  of the *place output tokens* by the following way:

- $\triangleright p =_{\text{def}} \{\text{elem}(AF(t, p)) \mid t \in \bullet p\}$ , if  $\bullet p \neq \emptyset$ ,
- $\triangleright p =_{\text{def}} \triangleright p$ , if  $(\bullet p = \emptyset) \wedge (p\bullet \neq \emptyset)$ ,
- $\triangleright p =_{\text{def}} \mathbf{N}$ , if  $(\bullet p = \emptyset) \wedge (p\bullet = \emptyset)$ ,
- $\triangleright p =_{\text{def}} \{\text{elem}(AF(p, t)) \mid t \in p\bullet\}$ , if  $p\bullet \neq \emptyset$ ,
- $\triangleright p =_{\text{def}} \triangleright p$ , if  $(p\bullet = \emptyset) \wedge (\bullet p \neq \emptyset)$ ,
- $\triangleright p =_{\text{def}} \mathbf{N}$ , if  $(p\bullet = \emptyset) \wedge (\bullet p = \emptyset)$ .

Then the condition  $\triangleright p \supseteq \triangleright p$  must be true for all the places  $p$  of the given SNT net  $SNTN$ .

*Marking*  $M$  of the SNT net  $SNTN$  is a mapping  $M : P \rightarrow (\mathbf{N}_0)_{MSE}$  such that  $\forall p \in P \forall e \in M(p) : e \in \triangleright p$  (i.e., every token  $e$  represents the chosen single nonnegative integer number and every token  $e$  that is associated with the place  $p$  in the marking  $M$  must be an element of the set  $\triangleright p$  of the place output tokens). Marking  $M$  of a given SNT net then assigns each place with the multiset over the set  $\mathbf{N}_0$  of all the nonnegative integer numbers and we use the form of  $\langle m \rangle$  for the single token representation in the net diagrams, where  $m \in \mathbf{N}_0$ . Marking  $M$  of the SNT net then expresses the current status of a process that is modeled by the given SNT net. If  $P = \{P1, P2, \dots, Pn\}$ , where  $n \in \mathbf{N}$ , then marking  $M$  of given SNT net can be written as a  $|P|$ -vector  $M = (M(P1), M(P2), \dots, M(Pn))$ .

A *SNT Petri net* is an ordered couple  $SNTPN = (SNTN, M_0)$ , where  $SNTN = (P, T, A, AF, TP)$  is a SNT net and  $M_0 : P \rightarrow (\mathbf{N}_0)_{MSE}$  is an initial marking of the net  $SNTN$ . We will then also denote a SNT Petri net by an ordered 6-tuple  $SNTPN = (P, T, A, AF, TP, M_0)$ .

Transition  $t \in T$  is *enabled* in the marking  $M$  of the SNT Petri net  $SNTPN$  if  $\exists \text{ind} \in N \forall p \in \bullet t \exists e \in M(p) : e = AF(p, t)(\text{ind})$  and we denote that fact symbolically in the form of  $t \text{ en } M$ . *Firing of the transition*  $t \in T$  results in changing the marking  $M$  into the marking  $M'$ , where

$\forall p \in P : M'(p) = (M(p) \setminus \{AF(p, t)(\text{ind})\}) \cup \{AF(t, p)(\text{ind})\}$ , that is, denoted by  $M[t]M'$ . Marking  $M_{n+1}$  is *reachable* from the marking  $M_1$  iff there exists a finite sequence  $\sigma = t_1 t_2 \dots t_n$  of the transitions, where  $t_1, t_2, \dots, t_n \in T$ ,  $n \in \mathbf{N}$ , such that  $M_1[t_1] M_2[t_2] M_3 \dots M_n[t_n] M_{n+1}$ . The set of all the markings reachable from the marking  $M$  will be denoted by the symbol  $[M]$ . The set of all finite and infinite transition sequences that are reachable from the marking  $M$  will be denoted by the symbol  $[M]^\gg$ ; that is,  $[M]^\gg = \{t_1 t_2 \dots t_n \mid \exists M_{n+1} \in [M] : M[t_1 t_2 \dots t_n] M_{n+1}, n \in \mathbf{N}\}$ .

Figure 1 illustrates the SNT Petri net  $N1 = (P, T, A, AF, TP, M_0)$ , where

- (i)  $P = \{P1, P2, P3, P4\}$ ,
- (ii)  $T = \{T1\}$ ,
- (iii)  $A = \{(P1, T1), (P2, T1), (T1, P3), (T1, P4)\}$ ,
- (iv)  $AF = \{((P1, T1), \langle 2, 1 \rangle), ((P2, T1), \langle 2, 3 \rangle), ((T1, P3), \langle 1, 2 \rangle), ((T1, P4), \langle 1, 1 \rangle)\}$ ,
- (v)  $TP = \{(T1, 1)\}$ ,
- (vi)  $M_0 = (3'1, 1'3, 1'2, \emptyset)$ .

It can be seen that  $\forall t \in T \exists k \in \mathbf{N} \forall p \in (\bullet t \cup t\bullet)$ :  $\text{length}(AF(p, t)) = \text{length}(AF(t, p)) = k$  for the net  $N1$ , because for the only one transition  $T1 \in T \exists 2 \in \mathbf{N} \forall p \in (\bullet T1 \cup T1\bullet)$ :  $\text{length}(AF(p, t)) = \text{length}(AF(t, p)) = 2$ ; that is, for any given transition  $t \in T$  and for all its input arcs  $(p, t)$  and all its output arcs  $(t, p)$  must be all the values of the arc functions  $AF(p, t)$  and  $AF(t, p)$  the sequences with the equal length  $k$ .

We can also determine that  $P1 \triangleright = \{1, 2\}$  (because  $P1\bullet \neq \emptyset$  and  $P1\bullet = \{\text{elem}(AF(P1, T1))\} = \{1, 2\}$ ),  $\triangleright P1 = \{1, 2\}$  (because  $(\bullet P1 = \emptyset) \wedge (P1\bullet \neq \emptyset)$ ,  $\triangleright P1 = P1\bullet$ ),  $P2 \triangleright = \{2, 3\} = \triangleright P2$ ,  $P3 \triangleright = \{1, 2\} = \triangleright P3$ , and  $P4 \triangleright = \{1\} = \triangleright P4$ . We can also verify that the condition  $\forall p \in P \forall e \in M_0(p) : e \in \triangleright p$  is fully satisfied for the net  $N1$ .

Transition  $T1$  is enabled in the marking  $M_0$  of the SNT Petri net  $N1$  because  $\exists \text{ind} \in N \forall p \in \bullet T1 \exists e \in M(p) : e = AF(p, T1)(\text{ind})$ , where  $\text{ind} = 2$ ,  $\bullet T1 = \{P1, P2\}$ ,  $AF(P1, T1)(2) = 1 \in 2'1 + 1'4 = M_0(P1)$ , and  $AF(P2, T1)(2) = 3 \in 1'3 = M_0(P2)$ . Firing of the transition  $T1$  results in changing of the net marking  $M_0$  into the marking  $M_1, M_0[T1]M_1$ , where  $M_1 = (1'1 + 1'4, \emptyset, 1'2 + 1'5, 1'1)$ , because we will remove the token  $\langle 1 \rangle$  from the place  $P1$  and the token  $\langle 3 \rangle$  from the place  $P2$  and then we will add the token  $\langle 2 \rangle$  (i.e.,  $AF(T1, P3)(2)$ ) into the place  $P3$  and the token  $\langle 1 \rangle$  (i.e.,  $AF(T1, P4)(2)$ ) into the place  $P4$  according to the firing of the transition general rule.

When enabling individual transitions of a given SNT Petri net so called *conflicts* can originate in its certain markings (or conflict transitions). At the enabling of transitions  $t_1$  and  $t_2$  of the given SNT Petri net in its marking  $M$ , where  $t_1, t_2 \in T$ , the conflict occurs, if  $((\bullet t_1 \cap \bullet t_2 \neq \emptyset) \wedge (t_1 \text{ en } M) \wedge (t_2 \text{ en } M) \wedge \neg(\{t_1, t_2\} \text{ en } M))$ . The term of the conflict transitions can be obviously easily generalized for the case of the finite set  $t_1, t_2, \dots, t_n$  ( $n \in \mathbf{N}$ ) of transitions of a given SNT Petri net.

A typical example of conflict transitions in the particular marking of the SNT Petri net  $N2$  is shown in Figure 2.

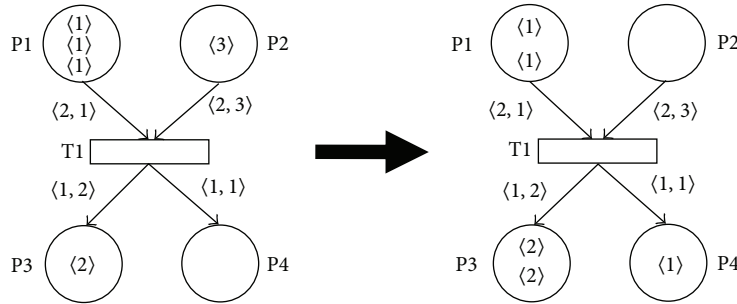


FIGURE 1: Firing of transition in SNT Petri net N1.

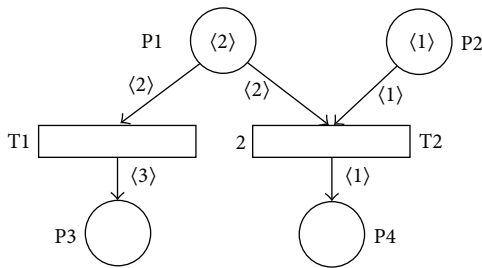


FIGURE 2: Conflict transitions in SNT Petri net N2.

The transitions T1 and T2 have the common input place P1; both are enabled but not enabled in parallel. When solving such transitions conflicts we will therefore follow the rule which determines, informally said, from the set of conflict transitions the one which will be enabled, whose value of the transition priority function  $TP$  is the highest. If such transition from the set of conflict transitions does not exist, the given conflict would have to be solved by other means. Our studied example will be then on the basis of that rule the transition T2 enabled (because  $TP(T1) = 1$  and  $TP(T2) = 2$ ).

A *SNT process net* (PN) is an ordered 9-tuple  $PN = (P, T, A, AF, TP, RP, IP, OP, \mathbf{RM}_s)$ , where

- (i)  $(P, T, A, AF, TP)$  is a *connected SNT net*,
- (ii)  $RP$  is a finite set of *resource places*,  $RP \subset P$ ,
- (iii) the *input place*  $IP$  is the only one place  $IP \in (P \setminus RP)$  such that  $\bullet IP = \emptyset$ ,
- (iv) the *output place*  $OP$  is the only one place  $OP \in (P \setminus RP)$  such that  $OP \bullet = \emptyset$ ,
- (v)  $\mathbf{RM}_s$  is the set of all the allowed *resource static markings* (see below).

The finite set of the resource places  $RP$  is used for expressing the conditions of a modeled system containing some initial resources and we use circles with the double line for their representation. The input place  $IP$  is the only one place from the set  $P \setminus RP$  of the nonresource places that has no input transition and similarly the output place  $OP$  is the only one place from the set  $P \setminus RP$  of the nonresource places of the SNT process net that has no output transition. The class of all the SNT process nets will be denoted by the symbol PNET.

*Marking*  $M$  of the SNT process net  $PN$  is again a mapping  $M : P \rightarrow (\mathbf{N}_0)_{MSE}$  such that  $\forall p \in P \forall e \in M(p) : e \in p$ . If  $RP = \{R1, R2, \dots, Rm\}$ , where  $m \in \mathbf{N}$ ,  $P \setminus RP = \{IP, P1, P2, \dots, Pn, OP\}$ , where  $n \in \mathbf{N}$ , then the marking  $M$  can be written as a  $|P|$ -vector  $M = (M(IP), M(P1), M(P2), \dots, M(Pn), M(R1), M(R2), \dots, M(Rm), M(OP))$ .

*Static marking*  $M_s$  of the SNT process net  $PN$  is a marking  $M_s$  such that  $\forall p \notin RP : M_s(p) = \emptyset; \forall t \in T : \neg(t \in M_s)$ ; that is, informally said, in the static marking  $M_s$  the tokens are allowed in the chosen resource places only and no transition is enabled in this static marking  $M_s$ . Static marking  $M_s$  of the SNT process net  $PN$  can then be written as a  $|P|$ -vector  $M_s = (\emptyset, \emptyset, \emptyset, \dots, \emptyset, M_s(R1), M_s(R2), \dots, M_s(Rm), \emptyset)$ . The set of all the allowed static markings  $M_s$  of the SNT process net  $PN$  we will denote by the symbol  $\mathbf{M}_s$ .

If  $M_s$  is the given static marking of the SNT process net  $PN$ , then by the symbol  $RM_s$  we denote the *resource static marking* associated with the static marking  $M_s$ , that is, the function  $RM_s : RP \rightarrow (\mathbf{N}_0)_{MSE}$ , where  $\forall p \in RP : RM_s(p) = M_s(p)$  (i.e., informally said, resource static marking  $RM_s$  associated with the static marking  $M_s$  of the SNT process net  $PN$  is the vector containing only the resource places markings of the static marking  $M_s$  and the static marking  $M_s$  of all the remaining places of the net  $PN$  is then equal to the empty set  $\emptyset$ ). If  $RP = \{R1, R2, \dots, Rm\}$ , where  $m \in \mathbf{N}$ , then the resource static marking  $RM_s$  of the SNT process net  $PN$  can be written as a  $|RP|$  vector  $RM_s = (RM_s(R1), RM_s(R2), \dots, RM_s(Rm))$ . The set of all the allowed resource static markings  $RM_s$  associated with the set  $\mathbf{M}_s$  of all allowed static markings  $M_s$  of the SNT process net  $PN$  will be denoted by the symbol  $\mathbf{RM}_s$ .

*Input marking*  $M_i$  of the SNT process net  $PN$  is a marking  $M_i$  such that  $M_i(IP) \neq \emptyset, M_i(OP) = \emptyset, \exists M_s \in \mathbf{M}_s \forall p \in (P \setminus \{IP, OP\}) : M_i(p) = M_s(p)$ ; that is, informally said, in the input marking  $M_i$  the input place  $IP$  must contain at least one token, the output place  $OP$  must not contain any token, and the marking  $M_i$  of the remaining places corresponds with the chosen static marking  $M_s$  from the set of all the allowed static markings  $\mathbf{M}_s$ . We will denote this chosen static marking  $M_s \in \mathbf{M}_s$  associated with the given input marking  $M_i$  by the symbol  $[M_s]_i$ . Input marking  $M_i$  of the process net  $PN$  can then be written as a  $|P|$ -vector  $M_i = (M_i(IP), \emptyset, \emptyset, \dots, \emptyset, [M_s]_i(R1), [M_s]_i(R2), \dots, [M_s]_i(Rm), \emptyset)$ .

*Output (exit) marking*  $M_x$  reachable from the input marking  $M_i$  of the SNT process net  $PN$  is a marking  $M_x$  such that  $M_x(IP) = \emptyset, M_x \in [M_i], |M_x(OP)| = |M_i(IP)|$ ,

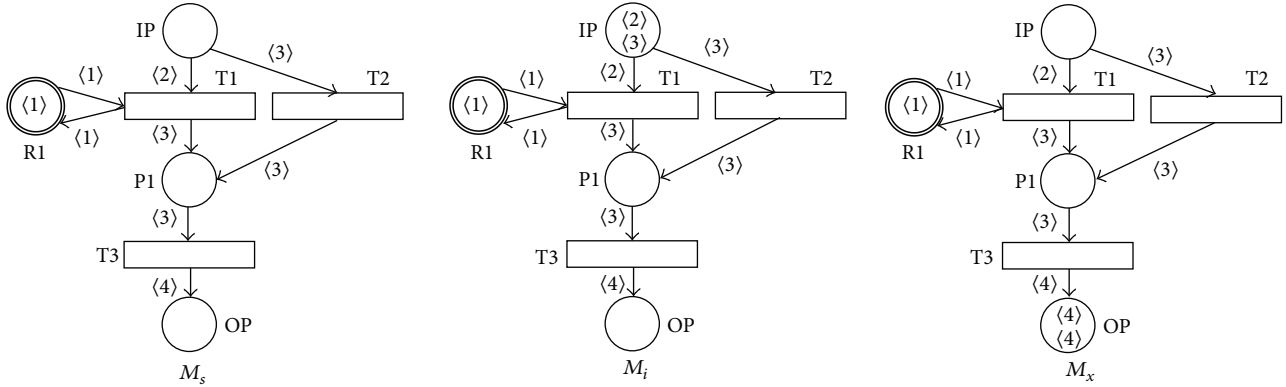


FIGURE 3: Static, input, and output markings of the SNT process net  $N3$ .

and  $\exists M'_s \in \mathbf{M}_s \forall p \in (P \setminus \{IP, OP\}) : M_x(p) = M'_s(p)$ ; that is, informally said, in the output marking  $M_x$  the input place  $IP$  must not contain any token, the output marking  $M_x$  is reachable from the input marking  $M_i$ , the output place  $OP$  must contain the same number of tokens in the output marking  $M_x$  as the input place  $IP$  in the input marking  $M_i$  and the markings of the remaining places correspond with the chosen static marking  $M'_s$  from the set of allowed static markings  $\mathbf{M}_s$ . We will denote this chosen static marking  $M'_s \in \mathbf{M}_s$  associated with the given output marking  $M_x$  by the symbol  $[M'_s]_x$ . Output marking  $M_x$  of the SNT process net  $PN$  can then be written as a  $|P|$ -vector  $M_x = (\emptyset, \emptyset, \emptyset, \dots, \emptyset, [M'_s]_x(R1), [M'_s]_x(R2), \dots, [M'_s]_x(Rm), M_x(OP))$ . The set of all the output markings  $M_x$  that are reachable from the input marking  $M_i$  of the SNT process net  $PN$  will be denoted by the symbol  $[M_i]_x$ .

Figure 3 illustrates the SNT process net  $N3$ , where  $RP = \{R1\}$  is the only one resource place of the net,  $IP$  is the input place  $IP$ ,  $OP$  is the output place  $OP$ , and the set  $\mathbf{RM}_s$  of all its allowed resource static markings  $\mathbf{RM}_s = \{(1'1)\}$ , in its static  $M_s$ , input  $M_i$ , and output  $M_x$  markings (note the resource place  $R1$  with the initial token in the static marking  $M_s$ ) where no transition must be enabled in (any) static marking  $M_s$ .

A SNT process Petri net is an ordered couple  $PPN = (PN, M_i)$ , where  $PN = (P, T, A, AF, TP, RP, IP, OP, \mathbf{RM}_s)$  is a SNT process net and  $M_i$  is an input marking of the  $PN$ . We will then also denote a SNT process Petri net by an ordered 10-tuple  $PPN = (P, T, A, AF, TP, RP, IP, OP, \mathbf{RM}_s, M_i)$ .

A SNT agent net (AN) is an ordered 10-tuple  $AN = (P, T, A, AF, TP, RP, IP, OP, \mathbf{RM}_s, OT)$ , where

- (i)  $(P, T, A, AF, TP)$  is a *connected SNT net*,
- (ii)  $RP$  is a finite set of *resource places*,  $RP \subset P$ ,
- (iii) the *input place*  $IP$  is the only one place  $IP \in (P \setminus RP)$  such that  $\bullet IP = \emptyset$ ,
- (iv) the *output place*  $OP$  is the selected place  $OP \in (P \setminus RP)$  such that  $OP \bullet = \{OT\}$ ,
- (v)  $\mathbf{RM}_s$  is the set of all the allowed *resource static markings*,
- (vi) the *output transition*  $OT$  is the selected transition  $OT \in T$  such that  $\bullet OT = \{OP\}$ ,  $TP(OT) = 1$ .

The concept of the SNT agent net such extends the concept of the SNT process net. So, informally said, the output place  $OP$  is the selected place of the SNT agent net that has the single output transition  $OT$  and the output transition  $OT$  is the selected transition of the SNT agent net that has the single input place  $OP$ ; it has no one output place and its transition priority function  $TP$  value is always the natural number 1. The meanings of the marking  $M$ , the static marking  $M_s$ , the input marking  $M_i$ , and the output marking  $M_x$  of the SNT agent net  $AN$  are the same as in the case of the SNT process net  $PN$ . The class of all the SNT agent nets will be denoted by the symbol ANET.

Figure 4 illustrates the SNT agent net  $N3$ , where  $RP = \{R1\}$  is the only one resource place,  $IP$  is the input place  $IP$ ,  $OP$  is the output place  $OP$ ,  $OT$  is the output transition  $OT$ , and the set  $\mathbf{RM}_s$  of all its allowed resource static markings  $\mathbf{RM}_s = \{(1'1)\}$ , in its input marking  $M_i$ .

Let  $AN = (P, T, A, AF, TP, RP, IP, OP, \mathbf{RM}_s, OT)$  be a SNT agent net. The SNT process net  $\underline{AN}$  associated with the SNT agent net  $AN$  is the 9-tuple  $\underline{AN} = (P, T, A, AF, TP, RP, IP, OP, \mathbf{RM}_s)$ , where  $P = P$ ,  $T = T \setminus \{OT\}$ ,  $A = A \setminus \{(OP, OT)\}$ ,  $AF = AF \setminus \{(OP, OT), k \mid k \in \mathbf{N}\}$ ,  $TP = TP \setminus \{(OT, 1)\}$ ,  $RP = RP$ ,  $IP = IP$ ,  $OP = OP$ , and  $\mathbf{RM}_s = \mathbf{RM}_s$ .

A SNT agent Petri net is an ordered couple  $APN = (AN, M_i)$ , where  $AN = (P, T, A, AF, TP, RP, IP, OP, \mathbf{RM}_s, OT)$  is a SNT agent net and  $M_i$  is an input marking of the  $AN$ . We will then also denote a SNT agent Petri net by an ordered 11-tuple  $APN = (P, T, A, AF, TP, RP, IP, OP, \mathbf{RM}_s, OT, M_i)$ . A SNT process Petri net  $\underline{APN} = (\underline{AN}, M_i)$  is associated with the SNT agent Petri net  $APN = (AN, M_i)$ .

Let  $PPN = (P, T, A, AF, TP, RP, IP, OP, \mathbf{RM}_s, M_i)$  be a SNT process Petri net. A place  $p \in P$  is said to be  $k$ -bounded iff  $\exists k \in \mathbf{N} \forall M \in [M_i] : |M(p)| \leq k$ . The  $PPN$  is said to be  $k$ -bounded iff every place of the  $PPN$  is  $k$ -bounded. We say that  $PPN$  *terminates properly* if  $(\forall M \in [M_i]) \exists M_x \in [M_i]_x : M_x \in [M] \wedge (|M_x| < \aleph_0)$ .

A SNT process net  $PN = (P, T, A, AF, TP, RP, IP, OP, \mathbf{RM}_s)$  is said to be *sound* iff the  $PPN = (P, T, A, AF, TP, RP, IP, OP, \mathbf{RM}_s, M_i)$  is  $k$ -bounded and it terminates properly for its every input marking  $M_i$ .

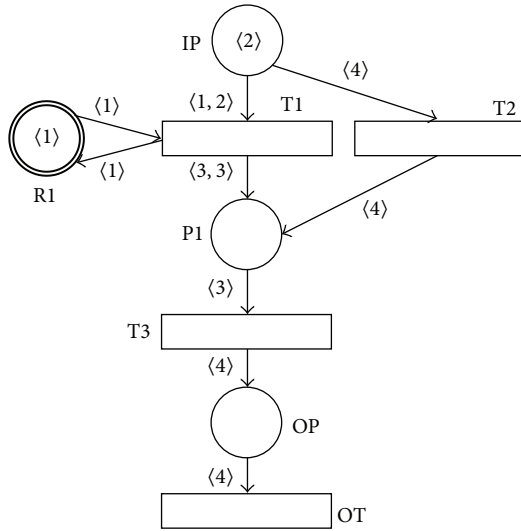


FIGURE 4: SNT agent net N4.

A SNT agent net  $AN = (P, T, A, AF, TP, RP, IP, OP, \mathbf{RM}_s, OT)$  is said to be *sound* iff the SNT process Petri net  $APN = (\underline{P}, \underline{T}, \underline{A}, \underline{AF}, \underline{TP}, \underline{RP}, \underline{IP}, \underline{OP}, \underline{\mathbf{RM}}_s, M_i)$  associated with the SNT agent Petri net  $APN = (P, T, A, AF, TP, RP, IP, OP, \mathbf{RM}_s, OT, M_i)$  is sound.

From many properties of the SNT process nets, respectively, SNT agent nets, we are especially interested in their soundness (in the terminology of the programming systems this property informally means that the programming system modeled by given process net will not cause deadlock [17] and memory or device overflow).

We will start our study of the sound SNT process nets by introduction of so called *base SNT process nets* (BPN). BPN represent the class of PNs that are elementary sound (i.e., soundness of these PNs can be trivially proved).

Figure 5 shows six simple BPNs in their static markings  $M_s \in \mathbf{M}_s$  (where  $\langle \underline{x} \rangle_n$  means nonempty finite sequence of the  $n$  nonnegative integer numbers; that is,  $\langle \underline{x} \rangle_n \in (\mathbf{N}_0)_{SQ}$ ,  $\text{length}(\langle \underline{x} \rangle_n) = n$ ,  $n \in \mathbf{N}$ , and  $\langle i \rangle_n$  means nonempty finite sequence of the  $n$  nonnegative integer numbers  $i \in \mathbf{N}_0$ ; that is,  $\langle i \rangle_n \in (\mathbf{N}_0)_{SQ}$ ,  $\langle i \rangle_n = \langle i, i, \dots, i \rangle$ , and  $\text{length}(\langle i \rangle_n) = n$ ,  $n \in \mathbf{N}$ ). The BPN *BASE1* is the simplest BPN at all and it contains only one place IOP that is simultaneously its input and output place (definition of PNs allows this case). BPN *BASE3* can be used for the modeling of the programming statement IF... THEN... ELSE... The BPN *BASE4* then represents the model of the programming critical section, where typically  $|M_s(R1)| = 1$  in the resource place R1. The BPN *BASE5* represents the synchronous method called mechanism where the called method is represented by the place P2. The BPN *BASE6* can be used for the modelling of the programming barrier.

**2.2. Place-Substitution Operator and Its Properties.** The place-substitution operator PLACE-SUBST is defined for the classes of PNs and ANs and it is shown to be conditionally preserving the soundness property. The sound PNs and ANs

then form a closed set (i.e., the application of this operator on any two PNs or any couple of PN and AN with the soundness property resulting in another PN or AN with the soundness property). The design of a given programming system with the support of the place-substitution operator will such typically start with the BPN *BASE1* and then it follows with the several place-substitution operations (i.e., substitution of the selected BPN over the place in the actual PN) with the support of selected BPNs that result into the complex SNT process net that models the part of the whole programming system.

**Definition 1.** Let  $PN1 = (P_1, T_1, A_1, AF_1, TP_1, RP_1, IP_1, OP_1, \mathbf{RM}_{s1})$  and  $PN2 = (P_2, T_2, A_2, AF_2, TP_2, RP_2, IP_2, OP_2, \mathbf{RM}_{s2})$  be the PNs, and let  $p \in P_1 \setminus RP_1$  be a place of the PN  $PN1$  such that

$$\begin{aligned} & ((\triangleright p \subseteq \triangleright IP_2) \wedge (OP_2 \triangleright \subseteq p \triangleright)) \vee (\triangleright p = p \triangleright = \mathbf{N}) \\ & \vee (\triangleright IP_2 = OP_2 \triangleright = \mathbf{N}). \end{aligned} \quad (2)$$

The function PLACE-SUBST:  $\text{PNET} \times \text{PNET} \rightarrow \text{PNET}$  of the PN  $PN2$  substitution over the place  $p$  of the PN  $PN1$  will be denoted by  $PN1.p \langle \rangle PN2$ . The execution of this function results in the new PN  $PN = (P, T, A, AF, TP, RP, IP, OP, \mathbf{RM}_s)$ , where

- (i)  $P = (P_1 \setminus \{p\}) \cup P_2$ ,
- (ii)  $T = T_1 \cup T_2$ ,
- (iii)  $A = A_2 \cup \{(x, y) \in A_1 \mid (x \neq p) \vee (y \neq p)\} \cup \{(x, IP_2) \mid (x, p) \in A_1\} \cup \{(OP_2, y) \mid (p, y) \in A_1\}$ ,
- (iv)  $AF = AF_2 \cup \{((x, y), v) \in AF_1 \mid (x \neq p) \vee (y \neq p)\} \cup \{((x, IP_2), v) \mid ((x, p), v) \in AF_1\} \cup \{((OP_2, y), v) \mid ((p, y), v) \in AF_1\}$ ,
- (v)  $TP = TP_1 \cup TP_2$ ,
- (vi)  $RP = RP_1 \cup RP_2$ ,
- (vii)  $IP = IP_1$ , if  $p \neq IP_1$ ;  $IP = IP_2$ , if  $p = IP_1$ ,
- (viii)  $OP = OP_1$ , if  $p \neq OP_1$ ;  $OP = OP_2$ , if  $p = OP_1$ ,
- (ix)  $\mathbf{RM}_s = \mathbf{RM}_{s1} \otimes \mathbf{RM}_{s2}$ .

**Example 2.** The result of the place-substitution operation  $SN1.P2 \langle \rangle SN2$  where the place  $P2$  of the BPN  $SN1$  (see Figure 5) was substituted by the BPN  $SN2$  can be shown in Figure 6. It is clear that all the necessary requests for the place-substitution operation are satisfied in this case, that is,  $P2$  is not a resource place of the PN  $SN1$ ,  $\triangleright P2 = \{4\} \subseteq \{4\} = \triangleright IP_2$ ,  $IP_2 \triangleright = \{4\} \subseteq \{4\} = P2 \triangleright$ . It is easy to verify that in this case the resulting PN  $SN1.P2 \langle \rangle SN2$  is also sound.

**Lemma 3.** Let  $PN1 = (P_1, T_1, A_1, AF_1, TP_1, RP_1, IP_1, OP_1, \mathbf{RM}_{s1})$ ,  $PN2 = (P_2, T_2, A_2, AF_2, TP_2, RP_2, IP_2, OP_2, \mathbf{RM}_{s2})$ , and  $PN3 = (P_3, T_3, A_3, AF_3, TP_3, RP_3, IP_3, OP_3, \mathbf{RM}_{s3})$  be the three PNs, let  $p1 \in P_1$  be a place of the PN  $PN1$  such that  $p1 \notin RP_1$ ,  $((\triangleright p1 \subseteq \triangleright IP_2) \wedge (OP_2 \triangleright \subseteq p1 \triangleright)) \vee (\triangleright p1 = p1 \triangleright = \mathbf{N}) \vee (\triangleright IP_2 = OP_2 \triangleright = \mathbf{N})$ , and let  $p2 \in P_2$  be a place

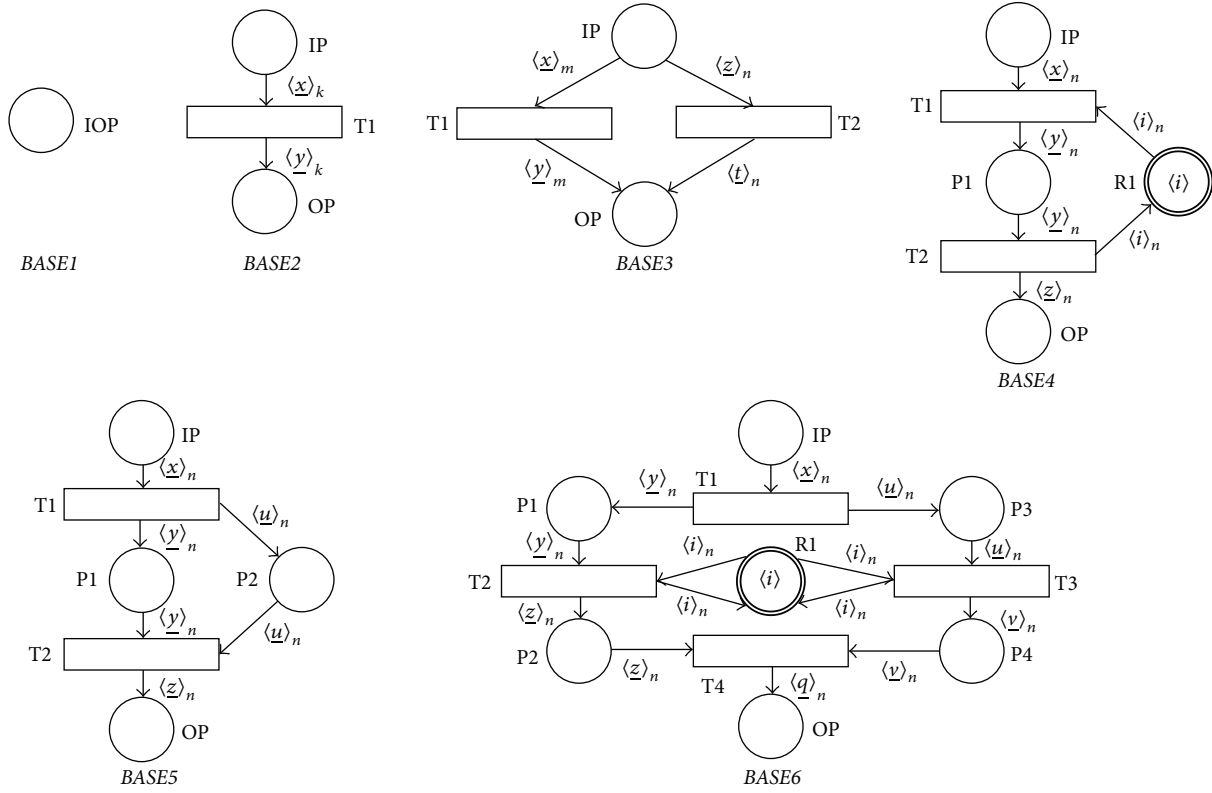


FIGURE 5: Examples of base process nets.

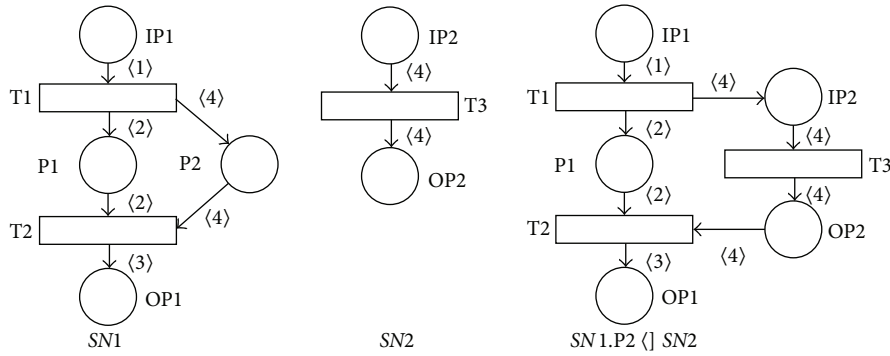


FIGURE 6: Result of place-substitution operation  $SN1.P2 \{\} SN2$ .

of the PN  $PN2$ , such that  $p2 \notin RP_2$ ,  $(\langle p2 \subseteq \rangle IP_3) \wedge (OP_3 \langle p2 \rangle) \vee (\langle p2 = p2 \rangle = \mathbf{N}) \vee (\langle IP_3 = OP_3 \rangle = \mathbf{N})$ . Then

$$\begin{aligned}
 &PN1.p1 \{\} (PN2.p2 \{\} PN3) \\
 &= (PN1.p1 \{\} PN2).p2 \{\} PN3.
 \end{aligned}
 \tag{3}$$

*Proof.* From Definition 1 it directly follows that  $PN1.p1 \{\} (PN2.p2 \{\} PN3) = (PN1.p1 \{\} PN2).p2 \{\} PN3 = (P, T, A, AF, TP, RP, IP, OP, \mathbf{RM}_s)$  where

- (i)  $P = (P_1 \setminus \{p_1\}) \cup (P_2 \setminus \{p_2\}) \cup P_3$ ,
- (ii)  $T = T_1 \cup T_2 \cup T_3$ ,

$$\begin{aligned}
 \text{(iii)} \quad A &= A_3 \cup \{(x, y) \in A_2 \mid (x \neq p_2) \vee (y \neq p_2)\} \cup \\
 &\{(x, IP_3) \mid (x, p_2) \in A_2\} \cup \{(OP_3, y) \mid (p_2, y) \in A_2\} \cup \\
 &\{(x, y) \in A_1 \mid (x \neq p_1) \vee (y \neq p_1)\} \cup \{(x, IP_2) \mid \\
 &(x, p_1) \in A_1\} \cup \{(OP_2, y) \mid (p_1, y) \in A_1\},
 \end{aligned}$$

$$\begin{aligned}
 \text{(iv)} \quad AF &= AF_3 \cup \{(x, y, v) \in AF_2 \mid (x \neq p_2) \vee (y \neq p_2)\} \cup \\
 &\{(x, IP_2, v) \mid ((x, p_2), v) \in AF_2\} \cup \{(OP_2, y, v) \mid \\
 &((p_2, y), v) \in AF_2\} \cup \{(x, y, v) \in AF_1 \mid (x \neq \\
 &p) \vee (y \neq p)\} \cup \{(x, IP_2, v) \mid ((x, p), v) \in AF_1\} \cup \\
 &\{(OP_2, y, v) \mid ((p, y), v) \in AF_1\},
 \end{aligned}$$

$$\text{(v)} \quad TP = TP_1 \cup TP_2 \cup TP_3,$$

$$\text{(vi)} \quad RP = RP_1 \cup RP_2 \cup RP_3,$$

- (vii)  $IP = IP_1$ , if  $p_1 \neq IP_1$ ;  $IP = IP_2$ , if  $(p_1 = IP_1) \wedge (p_2 \neq IP_2)$ ;  $IP = IP_3$ , if  $(p_1 = IP_1) \wedge (p_2 = IP_2)$ ,
- (viii)  $OP = OP_1$ , if  $p_1 \neq OP_1$ ;  $OP = OP_2$ , if  $(p_1 = OP_1) \wedge (p_2 \neq OP_2)$ ;  $OP = OP_3$ , if  $(p_1 = OP_1) \wedge (p_2 = OP_2)$ ,
- (ix)  $\mathbf{RM}_s = \mathbf{RM}_{s_1} \otimes \mathbf{RM}_{s_2} \otimes \mathbf{RM}_{s_3}$ .

□

**Lemma 4.** *The triple (PNET, PLACE-SUBST, BASE1) is a monoid.*

*Proof.* Lemma 4 is the direct consequence of Lemma 3 and it is clear that the BPN BASE1 (see Figure 5) is the neutral element of the monoid. □

*Definition 5.* Let  $AN1 = (P_1, T_1, A_1, AF_1, TP_1, RP_1, IP_1, OP_1, \mathbf{RM}_{s_1}, OT_1)$  be the AN and  $PN2 = (P_2, T_2, A_2, AF_2, TP_2, RP_2, IP_2, OP_2, \mathbf{RM}_{s_2})$  the PNs, and let  $p \in P_1 \setminus RP_1$  be a place of the AN AN1 such that

$$\begin{aligned} & ((\triangleright p \subseteq \triangleright IP_2) \wedge (OP_2 \triangleright \subseteq p)) \vee (\triangleright p = p \triangleright = \mathbf{N}) \\ & \vee (\triangleright IP_2 = OP_2 \triangleright = \mathbf{N}). \end{aligned} \quad (4)$$

The function PLACE-SUBST:  $ANET \times PNET \rightarrow ANET$  of the PN PN2 substitution over the place  $p$  of the AN AN1 will be denoted by  $AN1.p \langle \rangle PN2$ . The execution of this function results in the new AN  $AN = (P, T, A, AF, TP, RP, IP, OP, \mathbf{RM}_s, OT)$ , where

- (i)  $P = (P_1 \setminus \{p\}) \cup P_2$ ,
- (ii)  $T = T_1 \cup T_2$ ,
- (iii)  $A = A_2 \cup \{(x, y) \in A_1 \mid (x \neq p) \vee (y \neq p)\} \cup \{(x, IP_2) \mid (x, p) \in A_1\} \cup \{(OP_2, y) \mid (p, y) \in A_1\}$ ;
- (iv)  $AF = AF_2 \cup \{(x, y, v) \in AF_1 \mid (x \neq p) \vee (y \neq p)\} \cup \{((x, IP_2), v) \mid ((x, p), v) \in AF_1\} \cup \{((OP_2, y), v) \mid ((p, y), v) \in AF_1\}$ ,
- (v)  $TP = TP_1 \cup TP_2$ ,
- (vi)  $RP = RP_1 \cup RP_2$ ,
- (vii)  $IP = IP_1$ , if  $p \neq IP_1$ ;  $IP = IP_2$ , if  $p = IP_1$ ,
- (viii)  $OP = OP_1$ , if  $p \neq OP_1$ ;  $OP = OP_2$ , if  $p = OP_1$ ,
- (ix)  $\mathbf{RM}_s = \mathbf{RM}_{s_1} \otimes \mathbf{RM}_{s_2}$ ,
- (x)  $OT = OT_1$ .

Definition 5 is similar to Definition 1 in the sense of overloading of the operator  $\langle \rangle$ :  $PNET \times PNET \rightarrow PNET$  with the operator  $\langle \rangle$ :  $ANET \times PNET \rightarrow ANET$  of the PN PN2 substitution over the place  $p$  of the AN AN1 which is denoted by  $AN1.p \langle \rangle PN2$ . It is easy to formulate similar lemmas like Lemmas 3 and 4 for the case of this overloaded operator  $\langle \rangle$ .

We will investigate the necessary conditions for the soundness of the resulting SNT process or agent net in the following paragraphs. The first condition for the soundness of the resulting net is already in Definitions 1 and 5; that is, the substituted place  $p \in P$  must not be a member of the set of the net resource places  $RP$ ,  $p \notin RP$ . Figure 7 shows the result of the resource place R1 (that contains the token

$\langle 1 \rangle$  in the nonempty static marking  $M_s$ ) of the sound net SN3 substitution by the sound net SN4, that is, the net SN3.R1  $\langle \rangle$  SN4. The resulting net then does not satisfy the necessary conditions for its static marking  $M_s$ , because the token  $\langle 1 \rangle$  is associated in this static marking  $M_s$  with the nonresource place IP2, and the transition T2 is then enabled in this static marking  $M_s$  which is not allowed. If we remove the token  $\langle 1 \rangle$  from the place T2 then the resulting net is not sound again. So it is generally not possible to substitute any resource place of a given PN or AN by another PN.

Figure 8 shows the result of the place P2 of the sound net SN5 substitution by the sound net SN6, that is, the PN SN5.P2  $\langle \rangle$  SN6, that is not sound. It can be seen for its input marking  $M_i$ , where  $M_i(IP_1) = 1'1$ . In this input marking  $M_i$  the transition T1 is enabled and its firing results into the marking  $M_1$ , where  $M_i[T1]M_1$ . There is token  $\langle 3 \rangle$  in the place P1 and the token  $\langle 1 \rangle$  in the place P2 in the marking  $M_1$ , and the transitions T2 and T5 are enabled in parallel in the marking  $M_1$  and after their parallel firing the PN results in the marking  $M_2$ , where  $M_1[T2 T5]M_2$ . In the marking  $M_2$  there will be the token  $\langle 1 \rangle$  in the place OP2 and the token  $\langle 3 \rangle$  in the place OP1. No transition is enabled in the nonoutput marking  $M_2$  and so the resulting net SN5.P2  $\langle \rangle$  SN6 is not sound.

The loss of the net soundness then follows from the existence of the arc (P1, T2) in the resulting net. The necessary condition for the soundness of the resulting net is in this case the following:  $\forall t \in P2 \bullet : ((q \in \bullet t) \wedge (q \neq IP_1)) \Rightarrow (q \bullet = \{t\})$  and it does not hold for the place P2 of the PN SN5, because transition T3  $\in P2 \bullet$  holds that  $((P1 \in \bullet T3) \wedge (P1 \neq IP1))$  and  $(P1 \bullet = \{T2, T3\})$ . It is then clear that if the arc (P1, T2) is not present then the resulting net is sound.

Figure 9 shows the result of the place P2 of the sound net SN7 substitution by the sound net SN8, that is, the SNT process net SN7.P2  $\langle \rangle$  SN8, that is again not sound. The loss of the soundness (it can be easily seen for any input marking  $M_i$  of the resulting PN) then follows from the fact that  $P2 \triangleright = \{1, 4\}$ ; that is,  $|P2 \triangleright| > 1$ . The necessary condition for the soundness of the resulting PN is in this case the following:  $|P2 \triangleright| = 1$ .

**Lemma 6.** *Let  $PN1 = (P_1, T_1, A_1, AF_1, TP_1, RP_1, IP_1, OP_1, \mathbf{RM}_{s_1})$  and  $PN2 = (P_2, T_2, A_2, AF_2, TP_2, RP_2, IP_2, OP_2, \mathbf{RM}_{s_2})$  be two sound PNs, and let  $p \in P_1 \setminus RP_1$  be a place of the PN PN1 such that  $((\triangleright p \subseteq \triangleright IP_2) \wedge (OP_2 \triangleright \subseteq p)) \vee (\triangleright p = p \triangleright = \mathbf{N}) \vee (\triangleright IP_2 = OP_2 \triangleright = \mathbf{N})$ ,  $|p \triangleright| = 1$ ,  $\forall t \in p \bullet : ((q \in \bullet t) \wedge (q \neq IP_1)) \Rightarrow (q \bullet = \{t\})$ . Then the PN  $PN1.p \langle \rangle PN2$  is also sound.*

*Comment.* If the conditions  $p \in P_1 \setminus RP_1$ ,  $((\triangleright p \subseteq \triangleright IP_2) \wedge (OP_2 \triangleright \subseteq p)) \vee (\triangleright p = p \triangleright = \mathbf{N}) \vee (\triangleright IP_2 = OP_2 \triangleright = \mathbf{N})$ , and  $|p \triangleright| = 1$ ,  $\forall t \in p \bullet : ((q \in \bullet t) \wedge (q \neq IP_1)) \Rightarrow (q \bullet = \{t\})$  for the substituted place  $p \in P_1$  of the net PN1 are not satisfied the resulting net  $PN1.p \langle \rangle PN2$  does not need to be sound generally as follows from the previous three examples.

*Proof.* It is clear for  $p = IP_1$  or  $p = OP_1$  and it follows directly from the definition of the sound PN. Let  $p \neq IP_1$  and  $p \neq OP_1$ . Because the PN PN1 is sound, it holds that  $\forall M_i \exists k \in \mathbf{N} \forall q \in P_1 \forall M \in [M_i] : M(q) \leq k$ . This holds also for the substituted



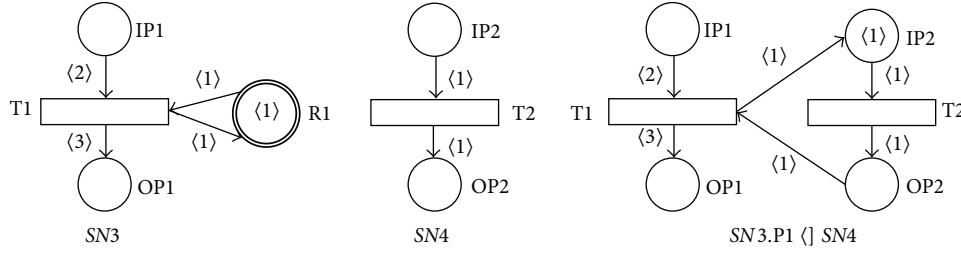


FIGURE 7: Result of place-substitution operation SN3.R1 {} SN4.

place  $p \in P_1$ . From this follows that also the place  $IP_2$  in the resulting PN is  $k$ -bounded because the number of tokens in any reachable marking of the process  $PN1.p \langle \rangle PN2$  is at the highest equaling the number of tokens in the place  $p$ . Because the PN  $PN2$  is also  $k$ -bounded and it terminates properly for any of its input markings  $M_i$ , from these facts follows that also the place  $OP_2$  in the resulting PN is  $k$ -bounded and so the whole PN  $PN1.p \langle \rangle PN2$  must be also  $k$ -bounded for any of its input markings  $M_i$ . Similarly, both PN  $PN1$  and  $PN2$  terminate properly for any of its input markings  $M_i$ . In particular for the PN  $PN1$  holds true that  $\forall M \in [M_i]$ ,  $M(p) \neq \emptyset$ , and  $\exists M_x : M_x \in [M]$  and for the PN  $PN2$  holds true that  $\forall M \in [M_i] \exists M_x : M_x \in [M]$ . The conditions  $|p\rangle = 1, \forall t \in p\bullet : ((q \in \bullet t) \wedge (q \neq IP_1)) \Rightarrow (q\bullet = \{t\})$  are satisfied for the PN  $PN1$  and from these conditions follows that  $|OP_2\rangle = 1, \forall t \in OP_2\bullet : ((q \in \bullet t) \wedge (q \neq IP_1)) \Rightarrow (q\bullet = \{t\})$  in the resulting net  $PN1.p \langle \rangle PN2$ . Then it follows that there must exist transition  $t, t \in OP_2\bullet$ , and the marking  $M$  of the resulting net  $PN1.p \langle \rangle PN2$  such that  $(M(OP_2) \neq \emptyset) \wedge (t \text{ en } M)$ , because the PN  $PN1$  terminates properly. So the whole PN  $PN1.p \langle \rangle PN2$  must also terminate properly for any of its input markings  $M_i$  and it is sound.  $\square$

It immediately follows from the previous lemma that the application of the operator PLACE-SUBST that satisfies all the necessary requests formulated in Lemma 6 for two given sound PNs results into another sound PN.

**Lemma 7.** Let  $AN1 = (P_1, T_1, A_1, AF_1, TP_1, RP_1, IP_1, OP_1, \mathbf{RM}_{s1}, OT_1)$  be the sound AN and  $PN2 = (P_2, T_2, A_2, AF_2, TP_2, RP_2, IP_2, OP_2, \mathbf{RM}_{s2})$  the sound PN, and let  $p \in P_1 \setminus RP_1$  be a place of the AN  $AN1$  such that  $((\rangle p \subseteq \rangle IP_2) \wedge (OP_2 \subseteq p\rangle)) \vee (\rangle p = p\rangle = \mathbf{N}) \vee (\rangle IP_2 = OP_2 \rangle = \mathbf{N}), |p\rangle = 1, \forall t \in p\bullet : ((q \in \bullet t) \wedge (q \neq IP_1)) \Rightarrow (q\bullet = \{t\})$ . Then the AN  $AN1.p \langle \rangle PN2$  is also sound.

*Proof.* Proof directly follows from the proof of Lemma 6.  $\square$

**2.3. Asynchronous-Processing Operator and Its Properties.** The asynchronous-processing operator ASYNC-PROC is the second operator defined for the class of all PNs and ANs and it is also shown to be preserving the soundness property. The sound PNs and ANs then form a closed set (i.e., the application of this operator on any couple of PN and AN or two ANs with the soundness property results in another PN or AN with the soundness property).

**Definition 8.** Let  $PN1 = (P_1, T_1, A_1, AF_1, TP_1, RP_1, IP_1, OP_1, \mathbf{RM}_{s1})$  be the PN and  $AN2 = (P_2, T_2, A_2, AF_2, TP_2, RP_2, IP_2, OP_2, \mathbf{RM}_{s2}, OT_2)$  the AN, and let  $t_1, t_2, \dots, t_n \in T_1, n \in \mathbf{N}$ , be the transitions of the PN  $PN1$ . The function ASYNC-PROC:  $\text{PNET} \times \text{ANET} \rightarrow \text{PNET}$  of the asynchronous-processing of the AN  $AN2$  from the PN  $PN1$  will be denoted by  $PN1.t_1.t_2 \dots t_n \text{ - } AN2$ . The execution of this function results in the new PN  $PN = (P, T, A, AF, TP, RP, IP, OP, \mathbf{RM}_s)$ , where

- (i)  $P = P_1 \cup P_2$ ,
- (ii)  $T = T_1 \cup T_2$ ,
- (iii)  $A = A_1 \cup A_2 \cup \{(t_1, IP_2)\} \cup \{(t_2, IP_2)\} \cup \dots \cup \{(t_n, IP_2)\}$ ,
- (iv)  $AF = AF_1 \cup AF_2 \cup \{(t_1, IP_2, k)\} \cup \{(t_2, IP_2, m)\} \cup \dots \cup \{(t_n, IP_2, n)\}$ , where  $k, m, n, \dots \in \rangle IP_2$ ,
- (v)  $TP = TP_1 \cup TP_2$ ,
- (vi)  $RP = RP_1 \cup RP_2$ ,
- (vii)  $IP = IP_1$ ,
- (viii)  $OP = OP_1$ ,
- (ix)  $\mathbf{RM}_s = \mathbf{RM}_{s1} \otimes \mathbf{RM}_{s2}$ .

**Example 9.** The result of the asynchronous-processing operation  $PN1.T1.T2 \text{ - } AN2$  where both PN  $PN1$  and AN  $AN2$  are sound can be shown in Figure 10. It is clear that all the necessary requests for the asynchronous-processing operation are satisfied in this case. It is also easy to verify that in this case the resulting PN  $PN1.T1.T2 \text{ - } AN2$  is also sound.

**Lemma 10.** Let  $PN1 = (P_1, T_1, A_1, AF_1, TP_1, RP_1, IP_1, OP_1, \mathbf{RM}_{s1})$  is a PN,  $AN2 = (P_2, T_2, A_2, AF_2, TP_2, RP_2, IP_2, OP_2, \mathbf{RM}_{s2}, OT_2)$  and  $AN3 = (P_3, T_3, A_3, AF_3, TP_3, RP_3, IP_3, OP_3, \mathbf{RM}_{s3}, OT_3)$  are two ANs, let  $t11, t21, \dots, t1n \in T_1$  are the transition of the PN  $PN1, n \in \mathbf{N}$ , and  $t12, t22, \dots, tm2 \in T_2, m \in \mathbf{N}$ , are the transition of the AN  $AN2$ . Then

$$PN1.t11 \dots t1n \text{ - } (AN2.t12 \dots tm2 \text{ - } AN3) = (PN1.t11 \dots t1n \text{ - } PN2) . t12 \dots tm2 \text{ - } PN3. \quad (5)$$

*Proof.* The proof is clear and follows directly from Definition 8.  $\square$

**Lemma 11.** The couple  $(\text{PNET} \cup \text{ANET}, \text{ASYNC-PROC})$  is a semigroup.

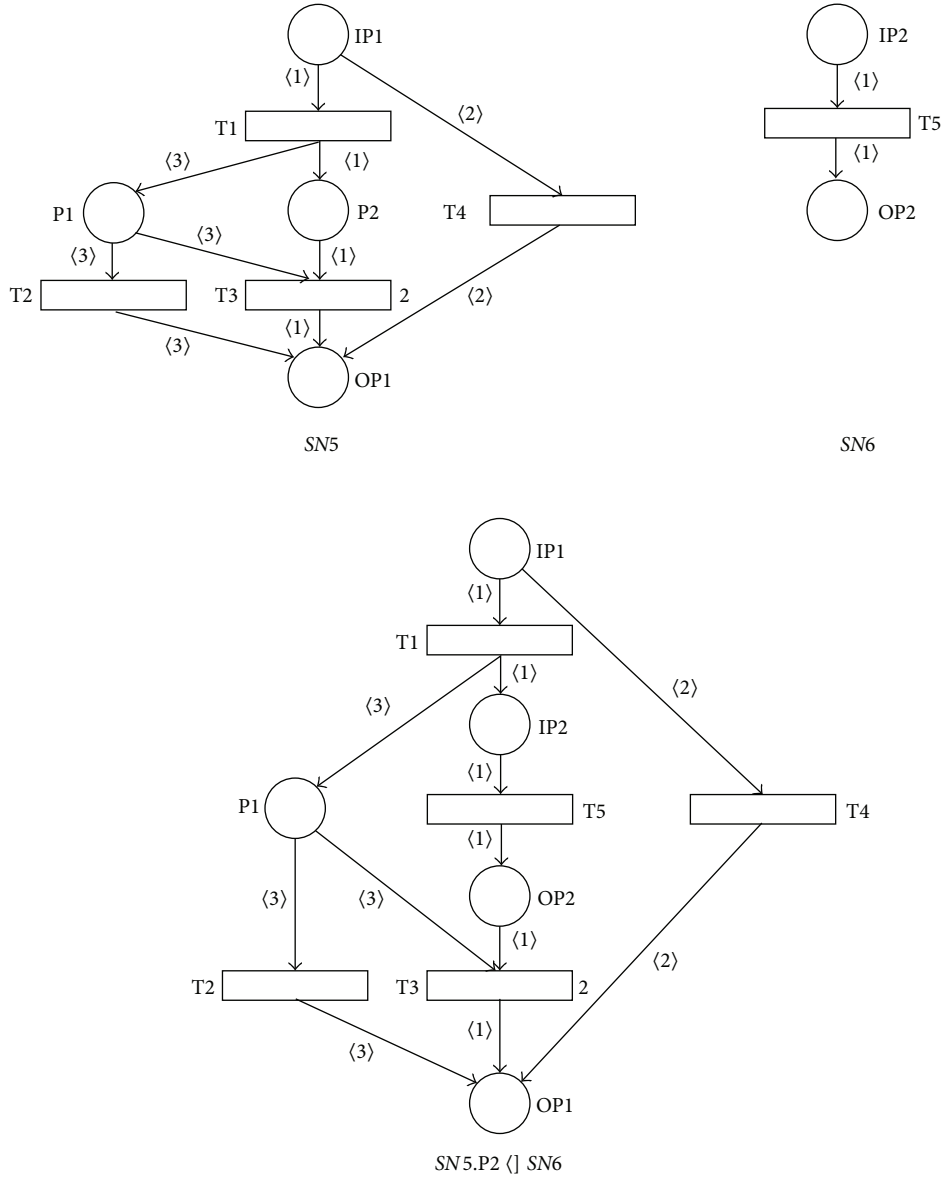


FIGURE 8: Result of place-substitution operation  $SN5.P2 [ ] SN6$ .

*Proof.* Lemma 11 is the direct consequence of Lemma 10.  $\square$

**Definition 12.** Let  $AN1 = (P_1, T_1, A_1, AF_1, TP_1, RP_1, IP_1, OP_1, \mathbf{RM}_{s_1}, OT_1)$  and  $AN2 = (P_2, T_2, A_2, AF_2, TP_2, RP_2, IP_2, OP_2, \mathbf{RM}_{s_2}, OT_2)$  be the ANs, and let  $t_1, t_2, \dots, t_n \in T_1, n \in \mathbf{N}$ , be the transitions of the AN  $AN1, AN1 \cap AN2 = \emptyset$ . The function  $ASYNC-PROC: ANET \times ANET \rightarrow ANET$  of the asynchronous-processing of the AN  $AN2$  from the AN  $AN1$  will be denoted by  $AN1.t_1.t_2. \dots .t_n - ] AN2$ . The execution of this function results in the new AN  $AN = (P, T, A, AF, TP, RP, IP, OP, \mathbf{RM}_s, OT)$ , where

- (i)  $P = P_1 \cup P_2$ ,
- (ii)  $T = T_1 \cup T_2$ ,
- (iii)  $A = A_1 \cup A_2 \cup \{(t_1, IP_2)\} \cup \{(t_2, IP_2)\} \cup \dots \cup \{(t_n, IP_2)\}$ ,

- (iv)  $AF = AF_1 \cup AF_2 \cup \{(t_1, IP_2), k\} \cup \{(t_2, IP_2), m\} \cup \dots \cup \{(t_n, IP_2), n\}$ , where  $k, m, n, \dots \in \mathbf{N}$ ,
- (v)  $TP = TP_1 \cup TP_2$ ,
- (vi)  $RP = RP_1 \cup RP_2$ ,
- (vii)  $IP = IP_1$ ,
- (viii)  $OP = OP_1$ ,
- (ix)  $\mathbf{RM}_s = \mathbf{RM}_{s_1} \otimes \mathbf{RM}_{s_2}$
- (x)  $OT = OT_1$ .

Definition 12 is similar to Definition 8 in the sense of the overloading of the operator  $-]$ :  $PNET \times ANET \rightarrow PNET$  with the operator  $-]$ :  $ANET \times ANET \rightarrow ANET$  of the AN  $AN2$  asynchronous-processing from the AN  $AN1$ . It is easy

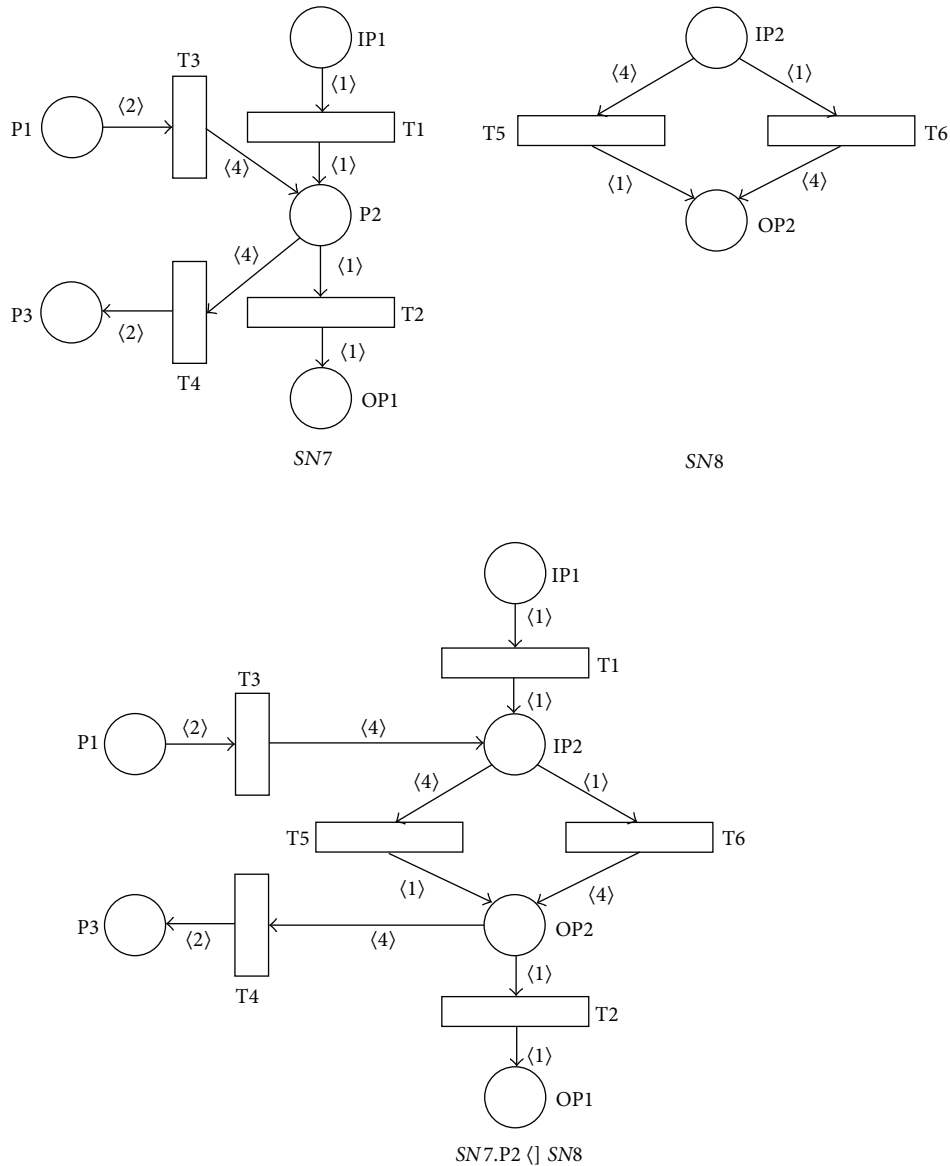


FIGURE 9: Result of place-substitution operation  $SN7.P2 \{\} SN8$ .

to formulate similar lemmas like Lemmas 10 and 11 for the case of this overloaded operator  $\cdot$ .

**Lemma 13.** *Let  $PN1 = (P_1, T_1, A_1, AF_1, TP_1, RP_1, IP_1, OP_1, \mathbf{RM}_{s1})$  be a sound PN and  $AN2 = (P_2, T_2, A_2, AF_2, TP_2, RP_2, IP_2, OP_2, \mathbf{RM}_{s2}, OT_2)$  a sound AN, and let  $t_1, t_2, \dots, t_n \in T_1$ ,  $n \in \mathbf{N}$ , be the transitions of the PN  $PN1$ . Then the PN  $PN1.t_1.t_2. \dots .t_n \cdot$  AN2 is also sound.*

*Proof.* Clear. Because the PN  $PN1$  is sound, it holds that  $\forall M_i \exists k \in \mathbf{N} \forall q \in P_1 \forall M \in [M_i] : M(q) \leq k$ . It also holds that  $\forall M_i \exists t_1 t_2 \dots t_m : M_i[t_1] M_1[t_2] M_2 \dots [t_m] M_x$ ,  $m \in \mathbf{N}$ ,  $t_1, t_2, \dots, t_m \in T_1$ . From these facts follows that every of the transition  $t_1, t_2, \dots, t_n \in T_1$ ,  $n \in \mathbf{N}$ , will be fired finite times in every transition sequence  $M_i[t_1] M_1[t_2] M_2 \dots [t_m] M_x$ .

From these immediately follows that in the resulting PN  $PN1.t_1.t_2. \dots .t_n \cdot$  AN2 is satisfied the condition  $\forall M_i \exists k \in \mathbf{N} \forall M \in [M_i] : M(IP_2) \leq k$ . Because of the soundness of the AN AN2 the whole PN  $PN1.t_1.t_2. \dots .t_n \cdot$  AN2 must be bounded. Because the PN  $PN1$  terminates properly for any of its input markings  $M_i$  and also the AN AN2 terminates properly and their firing of the transitions are independent it is clear that also the resulting PN  $PN1.t_1.t_2. \dots .t_n \cdot$  AN2 must terminate properly for any of its input marking and it is also sound.  $\square$

**2.4. Audiovisual Lecture Room with the Automatic Editing System.** Within the MERLINGO project implementation the selected standard audiovisual lecture room was additionally outfitted with the automated editing system consisting of three good quality cameras *Sony BPE NEX-FS100EK* series

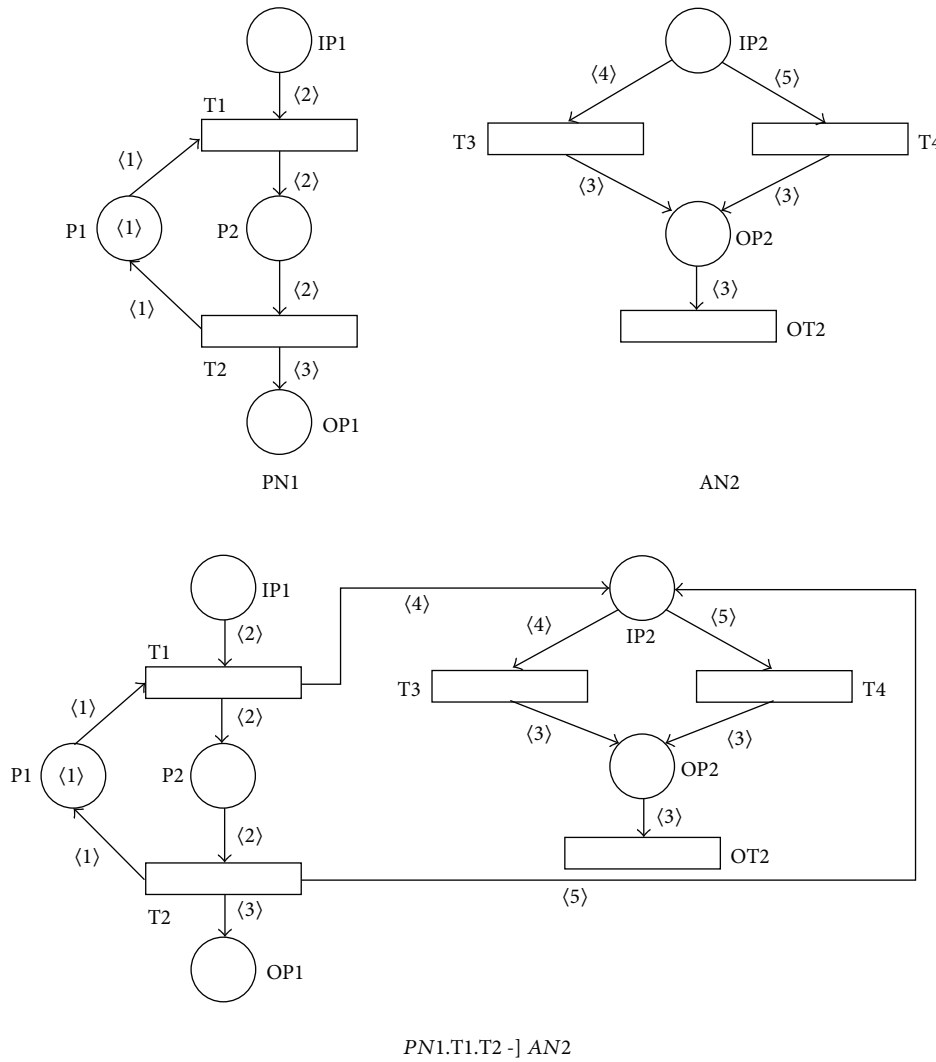


FIGURE 10: Result of asynchronous-processing operation PN1.T1.T2 -] AN2.

with the *object lens E18-200 mm* to eliminate the problem of insufficient lighting in lecture rooms during recording (see Figure 11), followed by editing unit *Roland VR-5* including three spatial microphones *Roland R-26* and control system with the motion sensors of *AMX 3100* line with the converter *RS232/MiDi*. That system ensures editing of video and audio signal read by the camera and microphone in the sectors where the lecturer is moving.

Additionally, this audiovisual lecture room has been outfitted with the programming system *EduArt* enabling transfer of image (video) of the scanned person and transfer of presentation from the work station, interactive board, visualizer, and sound in the high resolution via the computer network of the sufficient bandwidth. The output is a recording viewable by the web browser. The integration of the whole unit is ensured by the control system of the *AMX* product line automated in such manner that the lecturer can concentrate on the context of their message and does not need to pay attention to control of presentation tools and means for the recording



FIGURE 11: Three cameras Sony BPE NEX-FS100EK.

and transfer of the presentation into the resources of the central repository *MERLINGO*. There is no need for other persons to make a recording which is a cost saving feature and, additionally, the lecturer is not disturbed by them.

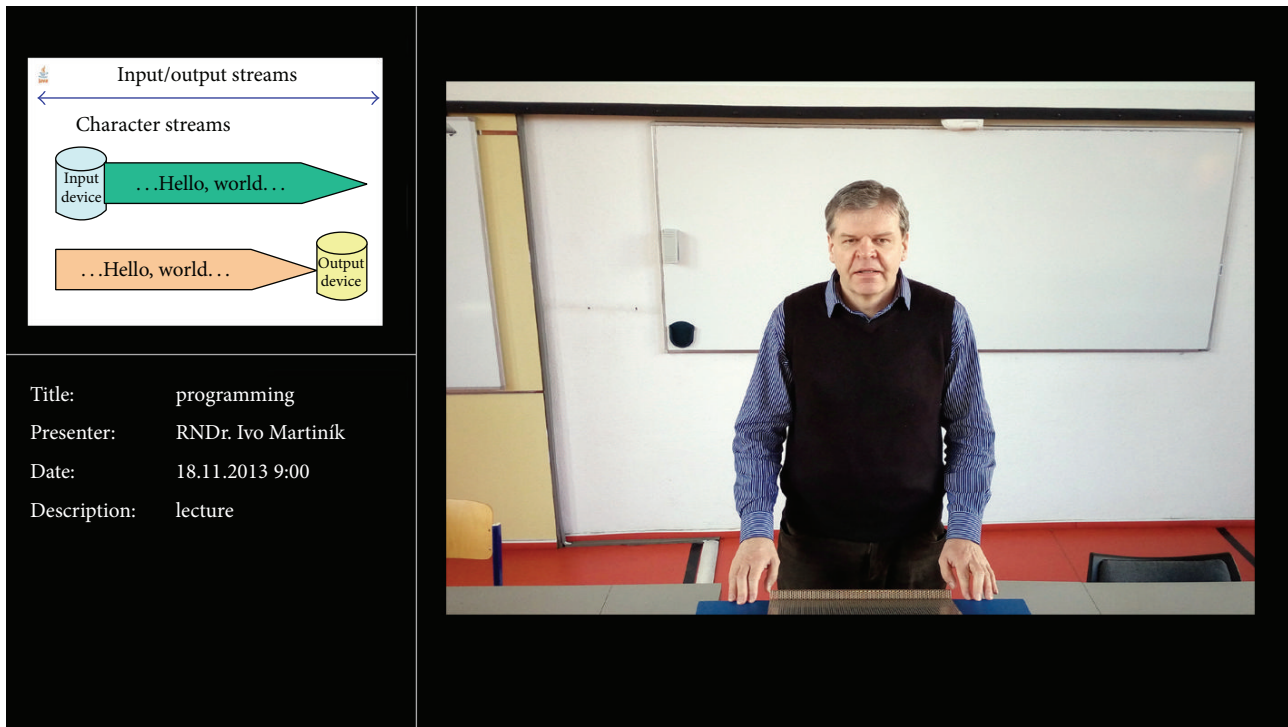


FIGURE 12: Shot of *detail-view* type in audiovisual lecture room.

For the realization of the automated editing were selected and implemented three basic types of shots:

- (i) *detail-view*: lecturer (one or more) standing by the speaker's desk; see Figure 12,
- (ii) *board-view*: lecturer (one or more) are found in the zone of the board or in the zone between the speaker's desk and board,
- (iii) *general-view*: lecturer (one or more) are found outside the shot *detail-view* or *board-view*; see Figure 13.

### 3. Results and Discussion

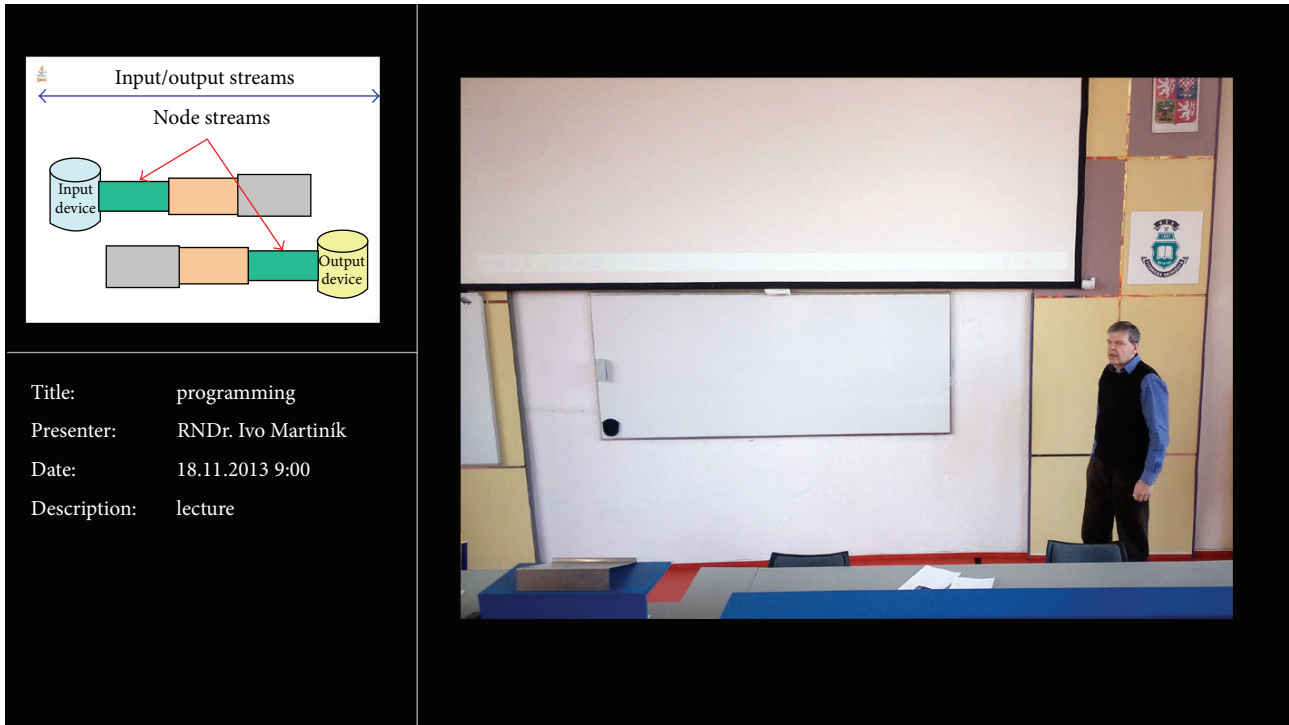
The SNT process and agent net formal theory have been used at the design, verification, simulation, and implementation phases of software support of the audiovisual lecture room enabling automated recordings of the presentations including automated editing.

When the speaker enters or exits one of the *general-view*, *board-view*, or *detail-view* areas the corresponding area sensor in the audiovisual lecture room sends the message describing this event. We will model these events by the tokens <1>–<6> in the resulting PN *RNET* by the following way:

- <1> entry of the speaker into the *general-view* area (from the non-scanned area of the lecture room),
- <2> exit of the speaker outside the *general-view* area (into the non-scanned area of the lecture room),
- <3> exit of the speaker outside the *general-view* area and entry of this speaker into the *board-view* area,

- <4> exit of the speaker outside the *board-view* area and entry of this speaker into the *general-view* area,
- <5> exit of the speaker outside the *board-view* area and entry of this speaker into the *detail-view* area,
- <6> exit of the speaker outside the *detail-view* area and entry of this speaker into the *board-view* area.

The PN *ROOM1* in Figure 14(a) then models the basic functionalities of the lecture room during the presenter(s) entry or exit into or outside the *general-view* area from or into the non-scanned area of the lecture room. The individual speaker that enters into the *general-view* area is represented by the token <10> in the input place *IP1* of the PN *ROOM1* in its input marking  $M_i$ . The input place *IP1* can contain in its input marking  $M_i$  any finite number of these tokens. The firing of the transition *T1* then represents an entry of the selected speaker into the *general-view* area from the non-scanned area of the lecture room. This entry then generates the activity of the chosen *general-view* area sensor that is modeled by the message represented by the token <1> in the resulting PN. Speaker can then leave out the *general-view* area into the non-scanned area of the lecture room by the firing of the transition *T3* (the message represented by the token <2> will be sent by the selected *general-view* area sensor as the consequence of this action). The transition *T2* is executed when the speaker leaves out the *general-view* and then he enters into the remaining areas of the lecture room. The token <10> that represents the chosen speaker inside the lecture room will be changed into the token <20> after the firing of the transition *T2*. This change of the token after the firing of the transition *T2* will such disable the possibility



Title: programming  
 Presenter: RNDr. Ivo Martinik  
 Date: 18.11.2013 9:00  
 Description: lecture

FIGURE 13: Shot of *general-view* type in audiovisual lecture room.

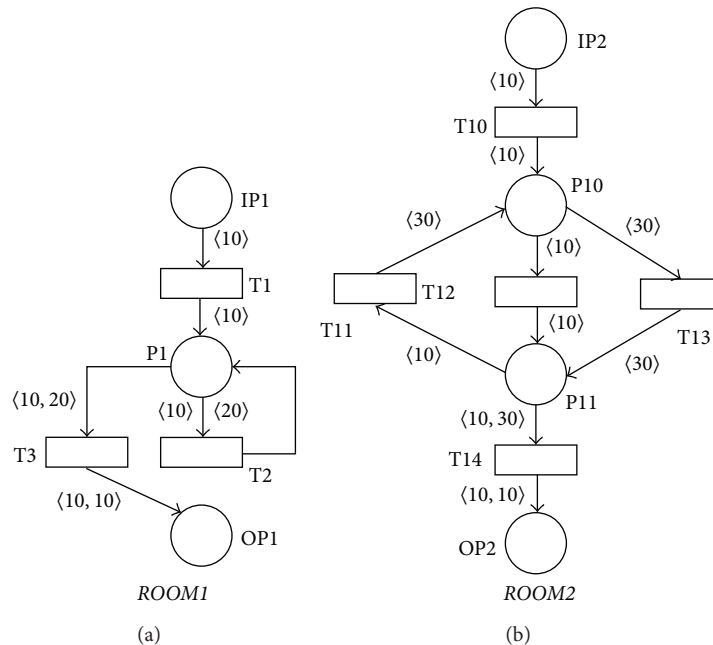


FIGURE 14: SNT process nets *ROOM1* and *ROOM2*.

of the infinite transitions sequence  $T_2 T_2 \dots$  that can be potentially fired by every token  $\langle 10 \rangle$  in the marking of the place  $P_2$ . Every token  $\langle 20 \rangle$  in the actual marking of the place  $P_2$  can then enable only the transition  $T_3$  that models leaving out the *general-view* area into the nonscanned area of the lecture room by the given speaker. It is clear that the PN *ROOM1* is sound for every one of its input markings  $M_i$ .

The PN *ROOM2* in Figure 14(b) then models the basic functionalities of the lecture room during the presenter(s) entry or exit into or outside *board-view* (represented by the place  $P_{10}$  and output place  $OP_2$ ) and *detail-view* (represented by the place  $P_{11}$ ) areas. The token(s)  $\langle 10 \rangle$  in the marking of the input place  $IP_2$  represent(s) individual speaker(s) in the *general-view* area of the lecture room. The firing of

the transition T10 models the entry into the *board-view* area by the individual speaker and sending of the message represented by the token  $\langle 3 \rangle$ . The entry into the *detail-view* area by the individual speaker and sending of the message represented by the token  $\langle 5 \rangle$  is then modeled by the firing of the transition T12. Every presenter has the possibility of revisiting the *board-view* area during its presentation (firing of the transition T11 and sending of the message  $\langle 6 \rangle$ ) and returning again into the *detail-view* area (firing of the transition T13 and sending of the message  $\langle 5 \rangle$ ). Every such speaker is then represented by the token  $\langle 30 \rangle$  after the firing of the transition T11 to prevent the possibility of the infinite transitions sequence T11 T13 T11 T13  $\dots$ . The firing of the transition T14 and sending of the message represented by the token  $\langle 4 \rangle$  then model an entry of the selected speaker back into the *board-view* area from the *detail-view* area of the lecture room. It is again clear that the PN ROOM2 is sound for every one of its input markings  $M_i$ .

PN ROOM that models all the main functionalities of the audiovisual lecture room is then the PN ROOM = ROOM1.P1  $\langle \rangle$  ROOM2 that is also sound according to Lemma 6.

Design and implementation of the programming systems that accept the messages generated by the area sensors of the audiovisual lecture room (these messages are represented by the tokens  $\langle 1 \rangle$ – $\langle 6 \rangle$  in the resulting PN) and that realize the automatic switching of the camera systems and the automatic editing functionality during recordings of presentations are represented by the AN CAM1 in Figure 15. The input place IP3 of the AN CAM1 accepts all the messages represented by the tokens  $\langle 1 \rangle$ – $\langle 6 \rangle$  and generated by the area sensors of the audiovisual lecture room. Resource place R20 with the token  $\langle 0 \rangle$  of the only one allowed static marking  $M_s$  of this AN (this static marking  $M_s$  is such the only-one element of the set  $M_s$  of the AN CAM1) represents the critical section mechanism of the whole programming system. This critical section guarantees that at most one message will be serviced by the programming system at the given time. The place P2 then represents the algorithm which realizes the automatic switching of the camera systems. The firing of the output transition OT3 then discards every accepted message from the given area sensor. It is again clear that the AN CAM1 is sound for every one of its input markings  $M_i$ .

Figure 16 shows the resulting PN RNET1 that is composed of the PN ROOM = ROOM1.P1  $\langle \rangle$  ROOM2 and the AN CAM1 using the ASYNC-PROC operator such that RNET1 = ROOM.T1.T3.T10.T11.T12.T13.T14 -] CAM1. The PN RNET1 introduces the mechanism of the messages generated by the area sensors of the audiovisual lecture room processing (these messages are represented by the tokens  $\langle 1 \rangle$ – $\langle 6 \rangle$  in the PN RNET1). The whole PN RNET1 is again sound according to Lemma 13.

The first inner part of the programming systems that accepts the messages generated by the area sensors of the audiovisual lecture room is modeled by the PN CAM2 in Figure 17. Every token of the actual static marking  $M_s$  of the only one resource place R30 of this PN represents the speakers that are actually presenting in the *general-view*, *board-view*, or *detail-view* areas at the given time. Every (anonymous) speaker that is presenting in the *general-view*,

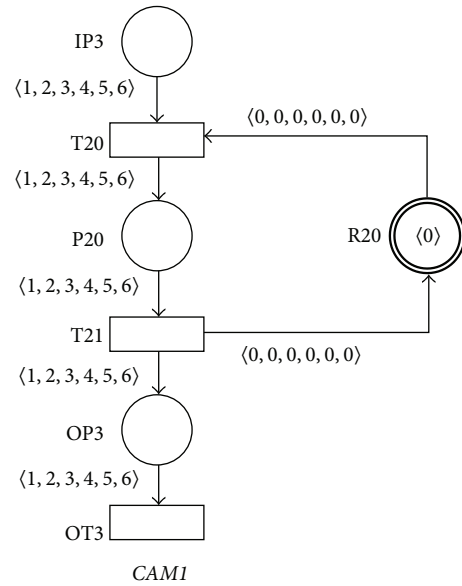
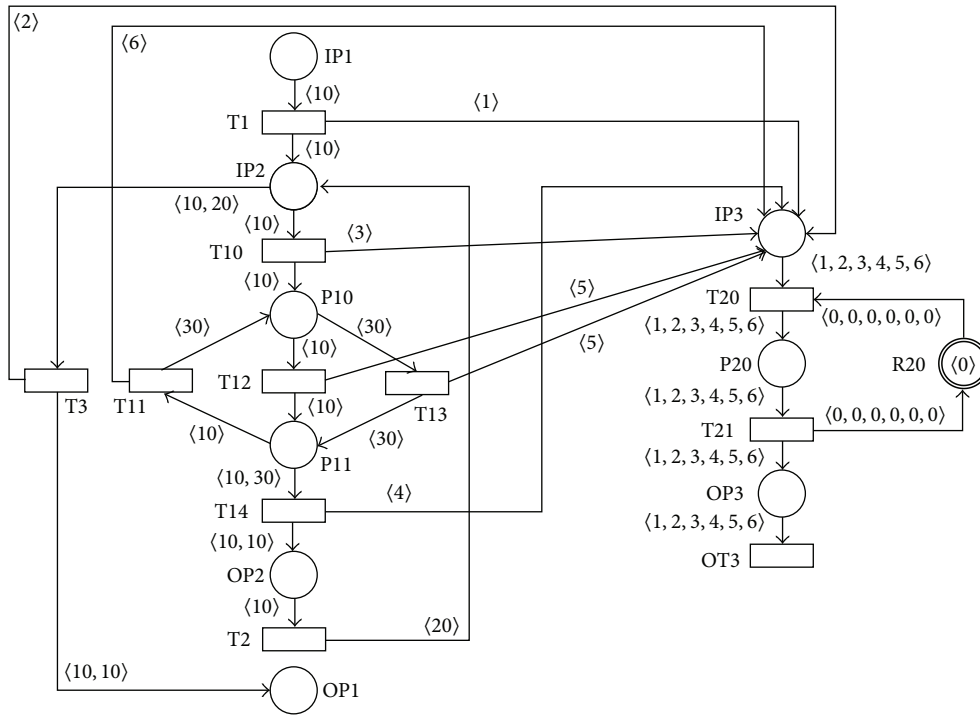


FIGURE 15: SNT agent net CAM1.

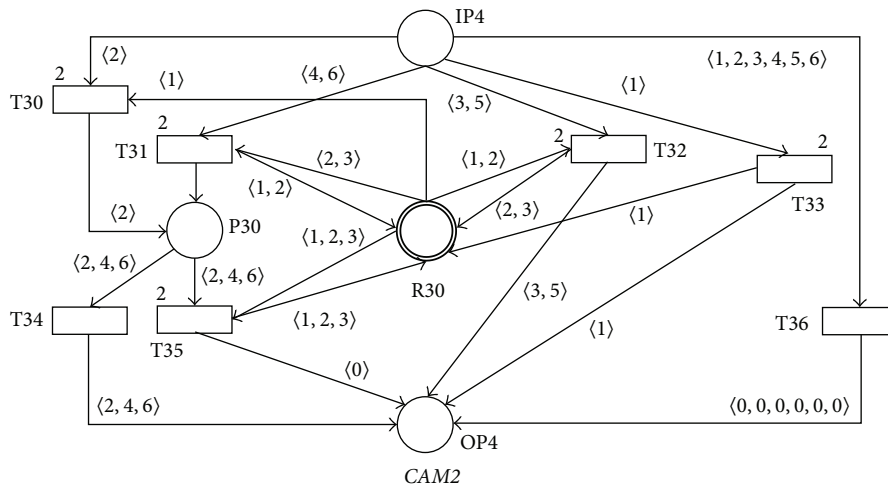
*board-view*, *detail-view* areas, respectively, at the given time is then modeled by the tokens  $\langle 1 \rangle$ ,  $\langle 2 \rangle$ , and  $\langle 3 \rangle$ , respectively, in the actual static marking  $M_s$  of the resource place R30. The set of all the allowed resource markings of this PN is then  $RM_s = \{(i' \langle 1 \rangle + j' \langle 2 \rangle + k' \langle 3 \rangle) \mid i \in \mathbf{N}_0, j \in \mathbf{N}_0, k \in \mathbf{N}_0\}$ . The functionalities of the PN CAM2 are as in the following:

- (i) Acceptance of the message  $\langle 1 \rangle$  (entry of the speaker into the *general-view* area from the nonscanned area of the lecture room) results in the firing of the transition T33 and adding of the token  $\langle 1 \rangle$  (entry of the speaker into the *general-view* area) into the resource place R30.
- (ii) Acceptance of the message  $\langle 3 \rangle$  (exit of the speaker outside the *general-view* area and entry of this speaker into the *board-view* area) results in the firing of the transition T32, removing the token  $\langle 1 \rangle$  (exit of the speaker outside the *general-view* area) from the resource place R30 and adding the token  $\langle 2 \rangle$  (entry of the speaker into the *board-view* area) into the resource place R30.
- (iii) Acceptance of the message  $\langle 5 \rangle$  (exit of the speaker outside the *board-view* area and entry of this speaker into the *detail-view* area) results in the firing of the transition T32, removing of the token  $\langle 2 \rangle$  (exit of the speaker outside the *board-view* area) from the resource place R30, and adding of the token  $\langle 3 \rangle$  (entry of the speaker into the *detail-view* area) into the resource place R30.
- (iv) Acceptance of the message  $\langle 2 \rangle$  (exit of the speaker outside the *general-view* area into the nonscanned area of the lecture room) results in the firing of the transition T32 and removing of the token  $\langle 1 \rangle$  (exit of the speaker outside the *general-view* area into the nonscanned area) from the resource place R30.



ROOM.T1.T3.T10.T11.T12.T13.T14 -] CAM1

FIGURE 16: SNT process net RNET1.



CAM2

FIGURE 17: SNT process net CAM2.

the resource place R30 still contains at least one token  $\langle 1 \rangle$  (i.e., at least one speaker is still presented in the *general-view* area) the transition T35 then fires and the token  $\langle 2 \rangle$  is changed into the token  $\langle 0 \rangle$  (i.e., no camera switching will be performed). The transition T34 will fire otherwise and the camera switching can be performed.

- (v) Acceptance of the message  $\langle 4 \rangle$  (exit of the speaker outside the *board-view* area and entry of this speaker into the *general-view* area) results in the firing of

the transition T31, removing of the token  $\langle 2 \rangle$  (exit of the speaker outside the *board-view* area) from the resource place R30 and adding of the token  $\langle 1 \rangle$  (entry of this speaker into the *general-view* area) into the resource place R30. If the resource place R30 still contains at least one token  $\langle 2 \rangle$  (i.e., at least one speaker is still presented in the *board-view* area) the transition T35 then fires and the token  $\langle 4 \rangle$  is changed into the token  $\langle 0 \rangle$  (i.e., no camera switching will be performed). The transition T34 will fire otherwise and the camera switching can be performed.



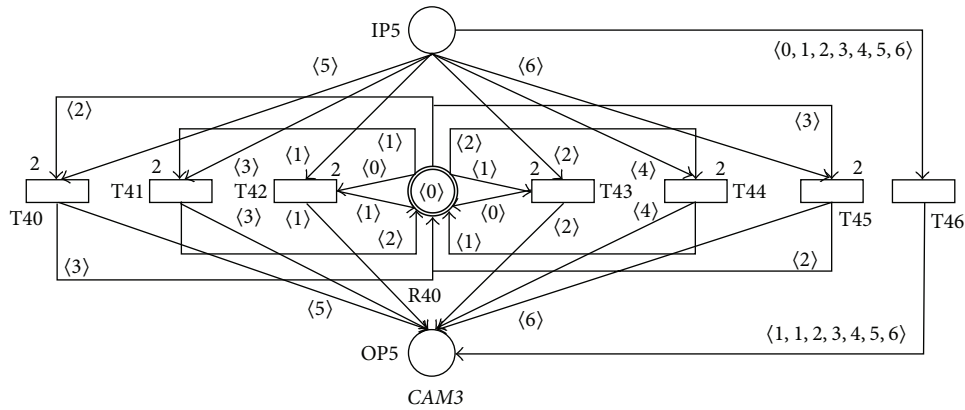


FIGURE 18: SNT process net CAM3.

- (vi) Acceptance of the message  $\langle 6 \rangle$  (exit of the speaker outside the *detail-view* area and entry of this speaker into the *board-view* area) results in the firing of the transition T31, removing of the token  $\langle 3 \rangle$  (exit of the speaker outside the *detail-view* area) from the resource place R30, and adding of the token  $\langle 2 \rangle$  (entry of this speaker into the *board-view* area) into the resource place R30. If the resource place R30 still contains at least one token  $\langle 3 \rangle$  (i.e., at least one speaker is still presented in the *detail-view* area) the transition T35 then fires and the token  $\langle 6 \rangle$  is changed into the token  $\langle 0 \rangle$  (i.e., no camera switching will be performed). The transition T34 will fire otherwise and the camera switching can be performed.
- (vii) The transition T35 will fire otherwise and no camera switching will be performed.

It is not difficult to prove that the whole PN CAM2 is sound for every one of its input markings  $M_i$ .

The second inner part of the programming systems that accepts the messages generated by the area sensors of the audiovisual lecture room is modeled by the PN CAM3 in Figure 18. Every token of the actual static marking  $M_s$  of the only one resource place R40 of this PN represents the actual state of the cameras for the *general-view*, *board-view*, or *detail-view* areas, such that

- (i) the token  $\langle 0 \rangle$  represents the state where all the three cameras are nonactive,
- (ii) the token  $\langle 1 \rangle$  represents the state where only the shot for the *general-view* area is actively scanned and all of the remaining cameras are nonactive,
- (iii) the token  $\langle 2 \rangle$  represents the state where only the shot for the *board-view* area is actively scanned and all of the remaining cameras are nonactive,
- (iv) the token  $\langle 3 \rangle$  represents the state where only the shot for the *detail-view* area is actively scanned and all of the remaining cameras are nonactive.

The set of allowed resource markings of this PN is then  $RM_s = \{ \langle k \rangle \mid k \in \{0, 1, 2, 3\} \}$ . The functionalities of the PN CAM2 are as in the following:

- (i) Acceptance of the message  $\langle 1 \rangle$  (entry of the speaker into the *general-view* area from the non-scanned area of the lecture room) results in the firing of the transition T42 only if all the three cameras are in nonactive states (i.e., the token  $\langle 0 \rangle$  is actually presented in the marking of the resource place R40) and the token  $\langle 1 \rangle$  will be then added into the resource place R40 (i.e., only the camera for the *general-view* area is in active state and all of the remaining cameras are in nonactive states), otherwise the transition T46 will fire and the actual states of all the cameras will be unchanged,
- (ii) Acceptance of the message  $\langle 2 \rangle$  (exit of the speaker from the *general-view* area and entry into the non-scanned area of the lecture room) results in the firing of the transition T43 if only the camera for the *general-view* area is in active state and all of the remaining cameras are in nonactive states (i.e., the token  $\langle 1 \rangle$  is actually presented in the marking of the resource place R40) and the token  $\langle 0 \rangle$  will be then added into the resource place R40 (i.e., all the cameras are in nonactive states), otherwise the transition T46 will fire and the actual states of all the cameras will be unchanged,
- (iii) ... and similarly for the acceptance of the messages  $\langle 3 \rangle$ – $\langle 6 \rangle$ .

It is again clear that the PN CAM3 is sound for every one of its input markings  $M_i$ .

We will then construct the resulting PN RNET with the support of the PLACE-SUBST operator applications such that

$$RNET = (RNET1.P20 \langle \rangle CAM2).OP4 \langle \rangle CAM3, \quad (6)$$

where  $RNET1 = ROOM.T1.T3.T10.T11.T12.T13.T14 - ] CAM1$  and  $ROOM = ROOM1.P1 \langle \rangle ROOM2$ ; that is,

$$RNET = (((ROOM1.P1 \langle \rangle ROOM2).T1.T3.T10.T11.T12.T13.T14 - ] CAM1).P20 \langle \rangle CAM2).OP4 \langle \rangle CAM3. \quad (7)$$

The resulting PN *RNET* is also sound according to Lemmas 6 and 13.

#### 4. Conclusions

Design and implementation of the programming systems which realize the automatic switching of the camera systems and the automatic editing functionality during recordings of presentations in the audiovisual lecture room involved the use of the SNT Petri process and agent nets formal theory. The simplified SNTPAN model of the given system is shown in Figures 14–18 (a more detailed model is much more complicated and its design used the theory of the timed SNTPANs [18]).

There is an opportunity to define additional operators for the class of SNTPANs and to generalize it (for instance, the SYNC-PROC composition operator that models synchronous nonrecursive calling of chosen method represented by a PN *PN2* of some programming system from the body of another method of the same programming system represented by a PN *PN1*). Selected PN or AN then represents typically a method in the model of the programming system realized with the support of the SNTPAN formalism.

At the Faculty of Economics, VŠB-Technical University of Ostrava, the recordings of presentations were realized with the support of highly mentioned lecture room with the automated editing. The rich-media recordings were published through the *MediaInTouch* programming system that is also integrated with the LMS system *Moodle* in the present time. This integration is based on the presence of the accessibility of the *MediaInTouch* application programming interface based on the web services technology. Thus it is possible to enable (i.e., insert or share) the recordings of presentations from the *MediaInTouch* repository as the organic part of every eLearning course content in the *Moodle* environment. The recordings are such then also the parts of the study materials for the mentioned subjects in the Bachelor and Master studies in the *Moodle* environment.

The lecture room with the support of automated editing enables *all-day fully automated* realization of all presentation recordings in the present time. There is then significant financial, personal, and time cost reduction joined with the creation and management of rich-media learning objects in comparison of the usual ways of recording. There is also verifiable improvement of the study results of the students (and particularly of the students with special needs) in the context of the availability of the presentation recordings and their accessibility online or on-demand as the standard part of the virtual university resources.

Those technologies can be also crucially beneficial during qualitative extension of provided services for the students with special needs, mainly at the establishment of “barrier-free” information access to recordings of presentations adapted to needs mainly for students with locomotive, visual aural disability while using internationally valid standards. *EduArt* and *MediaInTouch* programming systems are also extensively applied at the practical application of the methodology of adaptation of existing and newly created learning

objects which are adapted for students with special needs. The main results achieved in this area currently involve

- (i) technical solution of the parallel recording of translation of the lecturer into the sign language and text form. The programming implementation of the stated feature is based on the addition of next video stream containing translation of the lecturer into sign language or text form in the resulting record and their synchronizing with performed presentation,
- (ii) the transcription of standard eLearning text study supports in the audio form and their availability obtained via podcasting as a part of the MERLINGO portal services,
- (iii) automated transcription of spoken text of the lecture recorded by the recording and assistance service into the written text.

#### Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

#### Acknowledgment

This paper has been elaborated in the framework of the project CZ.1.07/2.3.00/20.0296 supported by the European Social Fund.

#### References

- [1] R. L. Daft and R. H. Lengel, “Organizational information requirements, media richness and structural design,” *Management Science*, vol. 32, no. 5, pp. 554–571, 1986.
- [2] N. Kock, D. A. Hantula, S. C. Hayne, G. Saad, P. M. Todd, and R. T. Watson, “Introduction to Darwinian perspectives on electronic communication,” *IEEE Transactions on Professional Communication*, vol. 51, no. 2, pp. 133–146, 2008.
- [3] J. Short, E. Williams, and B. Christie, *The Social Psychology of Telecommunications*, John Wiley & Sons, London, UK, 1976.
- [4] MERLINGO, *Media-Rich Repository of Learning Objects*, 2014, <http://www.merlingo.cz>.
- [5] EduArt, *Education Art*, 2014, <http://www.polymedia.cz/eduart.php>.
- [6] MediaInTouch, *Média na dosah*, 2014, <http://www.polymedia.cz/mediaintouch.php>.
- [7] G. A. Agha, F. de Cindio, and G. Rozenberg, *Concurrent Object-Oriented Programming and Petri Nets: Advances in Petri Nets*, Springer, Berlin, Germany, 2001.
- [8] R. David and H. Alla, *Petri Nets and Grafcet*, Prentice Hall, New York, NY, USA, 1992.
- [9] R. David and H. Alla, *Discrete, Continuous and Hybrid Petri Nets*, Springer, Berlin, Germany, 2010.
- [10] J. Desel and J. Esparza, *Free Choice Petri Nets*, Cambridge University Press, Cambridge, UK, 1995.
- [11] M. Diaz, *Fundamental Models, Verification and Applications*, John Wiley & Sons, London, UK, 2009.
- [12] C. Girault and R. Valk, *Petri Nets for Systems Engineering*, Springer, Berlin, Germany, 2001.

- [13] W. Reisig and G. Rozenberg, *Lectures on Petri Nets I: Basic Models*, Springer, Heidelberg, Germany, 1998.
- [14] W. Reisig, *Elements of Distributed Algorithms*, Springer, Berlin, Germany, 1998.
- [15] H. Huang, L. Jiao, T. Cheung, and W. M. Mak, *Property-Preserving Petri Net Process Algebra in Software Engineering*, World Scientific, Singapore, 2012.
- [16] W. van der Alst and K. van Hee, *Workflow Management: Models, Methods and Systems*, MIT Press, Cambridge, Mass, USA, 2002.
- [17] G. Juhás, I. Kazlov, and A. Juhásová, "Instance deadlock: a mystery behind frozen programs," in *Applications and Theory of Petri Nets*, J. Lilius and W. Penczek, Eds., vol. 6128 of *Lecture Notes in Computer Science*, pp. 1–17, Springer, Berlin, Germany, 2010.
- [18] L. Popova-Zeugmann, *Time and Petri Nets*, Springer, Heidelberg, Germany, 2013.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

