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

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 liveliness, 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-SUBSTITution) 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 \( \mathbb{N} \) denote the set of all natural numbers, \( \mathbb{N}_0 \) the set of all nonnegative integer numbers, \( \emptyset \) the empty set, \( \sim \) the logical negation operator, \( |A| \) the cardinality of the given set \( A \), where \( |\mathbb{N}| = \mathbb{N}_0, |P(A)| \) the family of all the subsets of a given set \( A \), and IDENT the set of all identifiers over a certain alphabet.

A multiset \( M \) over a nonempty set \( S \) is a function \( m : S \rightarrow \mathbb{N}_0 \). The nonnegative number \( m(a) \in \mathbb{N}_0 \), 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 \( \sum_{a \in S} m(a) a \). \( S_{\text{MS}} \) denotes the set of all nonempty multisets over the set \( S \) and \( S_{\text{MSE}} \) denotes the set of all multisets over the set \( S \).

Let \( A \) be a nonempty set. By the (nonempty) sequence \( \sigma \) over the set \( A \) a function \( \sigma \) can be understood, \( \sigma : \{1, 2, \ldots, n\} \rightarrow A \), where \( n \in \mathbb{N} \). The function \( \varepsilon : \emptyset \rightarrow A \) 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 \( A_{\text{SQ}} \) and the set of all finite (and possible empty) sequences over the set \( A \) by the notation \( A_{\text{SQE}} \). The length of the sequence \( \sigma = (a_1, a_2, \ldots, a_n) \), where \( n \in \mathbb{N} \), 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 \( \text{elem} : A_{\text{SQE}} \rightarrow P(A) \), such that \( \text{elem}(\sigma) = \{a | \exists i : a = \sigma(i)\}, \) where \( 1 \leq i \leq \text{length}(\sigma) \), \( \text{elem}(\varepsilon) = \emptyset \), assigns to each sequence \( \sigma \in A_{\text{SQE}} \) the set of all its elements.

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

Let \( V = [(V(1), V(2), \ldots, V(n)) | V(i) \in \mathbb{N}_0 \text{ for } 1 \leq i \leq n] \) and \( W = [(W(1), W(2), \ldots, W(m)) | W(j) \in \mathbb{N}_0 \text{ for } 1 \leq j \leq m] \). Then

\[
V \otimes W = \{ (V(1), \ldots, V(n), W(1), \ldots, W(m)) | V(i) \in \mathbb{N}_0 \text{ for } 1 \leq i \leq n, W(j) \in \mathbb{N}_0 \text{ for } 1 \leq j \leq m \},
\]

if \( (V \neq \emptyset) \land (W \neq \emptyset) \),

\[
= V, \quad \text{if } (V \neq \emptyset) \land (W = \emptyset),
\]

\[
= W, \quad \text{if } (V = \emptyset) \land (W \neq \emptyset),
\]

\[
= \emptyset, \quad \text{if } (V = \emptyset) \land (W = \emptyset).
\]

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

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 \( \text{NET} = (P, T, A) \), 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 if it is not composed of two or more disjoint and nonempty nets.

Some commonly used notations for the nets are \( \mathcal{S} = \{x | (x, y) \in A\} \) for the preset and \( \mathcal{Y} = \{x | (y, x) \in A\} \) 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 \( \text{SNTN} = (P, T, A, AF, TP) \), where

(i) \((P, T, A)\) is a connected net, \( \forall p \in P : \triangleright p \subseteq \triangleright p, \)
(ii) $AF$ is the arc function, $AF : (P \times T) \cup (T \times P) \to (\mathbb{N}_0)_{\leq 0}, AF(x, y) \in (\mathbb{N}_0)_{\leq 0}$ iff $(x, y) \in A, AF(x, y) = 0$ iff $(x, y) \notin A$, where $x, y \in P \cup T$, 
\[ \forall t \in T \exists k \in \mathbb{N} \ \forall p \in (\ast \cup t \ast) : \text{length}(AF(p, t)) = \text{length}(AF(t, p)) = k, \]

(iii) $TP$ is the transition priority function, $TP : T \to \mathbb{N}$.

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 $k \in \mathbb{N}$ 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 $p, p_\ast \in P(\mathbb{N})$ of the place input tokens and $p_\circ \in P(\mathbb{N})$ of the place output tokens by the following way:

\[ p = \text{def} \{ \text{elem}(AF(p, t)) | t \in \ast \cdot p \}, \text{if } \ast \cdot p \neq 0, \]
\[ p = \text{def} \{ \text{elem}(AF(p, t)) | t \in \circ \cdot p \}, \text{if } \circ \cdot p \neq 0, \]
\[ p = \text{def} \{ \text{elem}(AF(p, t)) | t \in \star \cdot p \}, \text{if } \star \cdot p \neq 0, \]

Then the condition $p \subseteq p_\circ$ must be true for all the places $p$ of the given SNT net $SNTN$.

Marking $M$ of the SNT net $SNTN$ is a mapping $M : P \to (\mathbb{N}_0)_{\leq 0}$ such that $\forall p \in P \forall e \in M(p) : e \in p_\ast$ (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 $p_\ast$ of the place output tokens). Marking $M$ of a given SNT net then assigns each place with the multiset over the set $\mathbb{N}_0$ of all the nonnegative integer numbers and we use the form of $(m)$ for the single token representation in the net diagrams, where $m \in \mathbb{N}_0$. Marking $M$ of the SNT net then expresses the current status of a process that is modeled by the given SNT net. If $P = \{P_1, P_2, \ldots, P_n\}$, where $n \in \mathbb{N}$, then marking $M$ of given SNT net can be written as a $|P|$-vector $M = (M(P_1), M(P_2), \ldots, M(P_n))$.

A SNT Petri net is an ordered couple $SNTPN = (SNTN, M_0)$, where $SNTN = (P, T, A, AF, TP)$ is a SNT net and $M_0 : P \to (\mathbb{N}_0)_{\leq 0}$ is an initial marking of the SNT net. We will then 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 $\exists \text{ind} \in N \forall p \in \ast \cdot p \exists e \in M(p) : e = AF(p, t)$ (and we denote that fact symbolically in the form of $t \in M$). 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 $\sigma = t_1 t_2 \cdots t_n$ of the transitions, where $t_1, t_2, \ldots, t_n \in T, n \in \mathbb{N}$, such that $M_1 t_1 M_2 t_2 \cdots M_n t_n M_{n+1}$. 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 $[M\infty]$; that is, $[M\infty] = \{t_1 t_2 \cdots t_n | \exists M_{n+1} \in [M] : M_1 t_1 t_2 \cdots t_n M_{n+1}, n \in \mathbb{N}\}$.

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

\[ (i) P = \{P_1, P_2, P_3, P_4\}, \]
\[ (ii) T = \{T_1\}, \]
\[ (iii) A = \{(P_1, T_1), (P_2, T_1), (T_1, P_3), (T_1, P_4)\}, \]
\[ (iv) AF = \{(\{P_1, T_1\}, (2, 1)), (\{P_2, T_1\}, (2, 3)), (\{T_1, P_3\}, (1, 2)), (\{T_1, P_4\}, (1, 1))\}, \]
\[ (v) TP = \{(T_1, 1)\}. \]
\[ (vi) M_0 = (\{3, 1\}^t, 1^t, 1^t, 0^t). \]

It can be seen that $\forall t \in T \exists k \in \mathbb{N} \forall p \in (\ast \cup t \ast)$: length$(AF(p, t)) = length(AF(t, p)) = k$ for the net $N1$, because for the only one transition $T_1$ in $T = \{T_1\} \subseteq \mathbb{N} \forall p \in (\ast T_1 \cup T_1 \ast)$: length$(AF(p, t)) = length(AF(t, p)) = 2$; 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 $P_1^\cdot = \{1, 2\}$ (because $P_1 \neq \emptyset$ and $P_1 = \{\text{elem}(AF(P_1, T_1))\} = \{1, 2\}$), $P_1 = \{1, 2\}$ (because $(\ast P_1 = \emptyset) \land (P_1 \neq \emptyset)$, $P_1^\cdot = \{1, 2\}$), $P_2^\cdot = \{2, 3\} = \{P_2, P_3\}$, and $P_4^\cdot = \{1\} = \star P_4$. We can also verify that the condition $\forall p \in P \forall e \in M_0(p) : e \in p_\circ$ is fully satisfied for the net $N1$.

Transition $T_1$ is enabled in the marking $M_0$ of the SNT Petri net $N1$ because $\exists \text{ind} \in N \forall p \in \ast T_1 \exists e \in M(p) : e = AF(p, T_1)(\text{ind})$, where ind = 2, $T_1 = \{P_1, P_2\}$, $AF(P_1, T_1)(2) = 1 \in 2^t + 1^t$ $M_0(P_1)$, and $AF(P_2, T_1)(2) = 2 \in 1^t + 3^t = M_0(P_2)$. Firing of the transition $T_1$ results in changing of the net marking $M_0$ into the marking $M_1$, $M_0^\cdot T_1 M_1$, where $M_1 = (1^t + 1^t, 0^t, 1^t + 1^t)$. Because we will remove the token 1 from the place $P_1$ and the token 2 from the place $P_2$ and then we will add the token 2 (i.e., $AF(T_1, P_3)(2)$) into the place $P_3$ and the token 1 (i.e., $AF(T_1, P_4)(2)$) into the place $P_4$ 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 $t_1 \cdot t_2 \in T$, the conflict occurs, if $(\ast t_1 \cap \ast t_2 \neq \emptyset) \land (t_1 \in M) \land (t_2 \in M) \land (\ast t_2 \cap \ast t_1 \neq \emptyset)$. The term of the conflict transitions can be obviously easily generalized for the case of the finite set $t_1, t_2, \ldots, t_n (n \in \mathbb{N})$ 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.
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 $PN = (P, T, A, AF, TP, RP, IP, OP, RM_s)$, 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 $IP = 0$,
(iv) the output place OP is the only one place $OP \in (P \setminus RP)$ such that $OP = 0$,
(v) $RM_s$ 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 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 $M : P \rightarrow (N_0)_{MSE}$ such that $\forall p \in P \forall e \in M(p) : e \in p$. If $RP = \{R1, R2, \ldots, Rm\}$, where $m \in N, P \setminus RP = \{IP, P1, P2, \ldots, Pn, OP\}$, where $n \in N$, then the marking $M$ can be written as a $|P|$-vector $M = (M(IP), M(P1), M(P2), \ldots, M(Pn), M(R1), M(R2), \ldots, M(Rm), M(OP))$.

Static marking $M_s$ of the SNT process net $PN$ is a marking $M_s$ such that $\forall p \notin RP : M_s(p) = 0; \forall t \in T : \tau(t \; en \; M_s)$; 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 $M_s = (0, 0, 0, \ldots, 0, M_s(R1), M_s(R2), \ldots, M_s(Rm), 0)$. The set of all the allowed static markings $M_s$ of the SNT process net $PN$ we will denote by the symbol $RM_s$.

If $M_i$ is the given static marking of the SNT process net $PN$, then by the symbol $RM_i$ we denote the resource static marking associated with the static marking $M_i$, that is, the function $RM_i : RP \rightarrow (N_0)_{MSE}$, where $\forall p \in RP : RM_i(p) = M_i(p)$ (i.e., informally said, resource static marking $RM_i$ associated with the static marking $M_i$ of the SNT process net $PN$ is the vector containing only the resource places markings of the static marking $M_i$ and the static marking $M_i$ of all the remaining places of the net $PN$ is then equal to the empty set $\emptyset$). If $RP = \{R1, R2, \ldots, Rm\}$, then the resource static marking $RM_s$ of the SNT process net $PN$ can be written as a $|RP|$-vector $RM_s = (RM_s(R1), RM_s(R2), \ldots, RM_s(Rm))$. The set of all the allowed resource static markings $RM_s$ associated with the set $M_s$ of all allowed static markings $M_s$ of the SNT process net $PN$ will be denoted by the symbol $RM_s$.

Input marking $M_i$ of the SNT process net $PN$ is a marking $M_i$ such that $M_i(IP) \neq 0, M_i(OP) = 0, \exists M_s \in RM_s, \forall p \in (P \setminus \{IP, OP\}) : M_s(p) = M_i(p)$; 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 $M_s$ of the remaining places corresponds with the chosen static marking $M_s$ from the set of all the allowed static markings $M_s$. We will denote this chosen static marking $M_s \in RM_s$ associated with the given input marking $M_i$ by the symbol $[M_i]$. Input marking $M_i$ of the process net $PN$ can then be written as a $|P|$-vector $M_i = (M_i(IP), 0, 0, \ldots, 0, [M_i](R1), [M_i](R2), \ldots, [M_i](Rm), 0)$.

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 $M_x(IP) = 0, M_x \in [M_i], [M_x(OP)] = [M_i(IP)]$, where $[.]$ represents the mapping $[.] : M_i \rightarrow (N_0)_{MSE}$, $\forall p \in P \forall e \in M_x(p) : e \in p$.
and \( \exists M'_y \in M_y \forall p \in (P \setminus \{IP, OP\}) : M'_y(p) = M'_y(p); \)
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 \( M_t \), the output place \( OP \) must contain the same number of tokens in the output marking \( M_t \) as the input place \( IP \) in the input marking \( M_t \) and the markings of the remaining places correspond with the chosen static marking \( M'_y \) from the set of allowed static markings \( M_y \). We will denote this chosen static marking \( M'_y \) of \( M_y \) with the given output marking \( M_x \) by the symbol \([M'_y]_x\). Output marking \( M_x \) of the SNT process net \( PPN \) can then be written as a \(|P|\)-vector \( M_x = (0,0, \ldots ,0, [M'_y]_x(R1), [M'_y]_x(R2), \ldots , [M'_y]_x(Rm), M_x(OP)) \). The set of all the output markings \( M_x \) that are reachable from the input marking \( M_t \) of the SNT process net \( PPN \) will be denoted by the symbol \([M'_y]_x\).

Figure 3 illustrates the SNT process net \( N3 \), where \( RM = \{R1\} \) is the only one resource place, \( OP \) is the output place \( IP \), \( OP \) is the output place \( OP \), \( OT \) is the output transition of the \( SNT \) process net \( PPN \). The class of all the SNT agent nets will be denoted by the symbol \( AN \).

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 \( M_t \), the static marking \( M_y \), the input marking \( M_t \), and the output marking \( M_x \) of the SNT agent net \( AN \) are the same as in the case of the SNT process net \( PPN \). Let AN = \((P, T, A, AF, TP, RP, IP, OP, RM_x, OT)\) be a SNT agent net. The SNT process net \( AN \) associated with the SNT agent net \( AN \) is the 9-tuple \( AN = \langle P, T, A, AF, TP, RP, IP, OP, RM_x, OT \rangle \), where \( P \) is the input place \( IP \), \( IP \) is the input place \( IP \) and \( RM_x \) is the set of all the allowed resource static markings of all its allowed resource static markings \( RM_x = \{(1'1)\} \), in its input marking \( M_t \).

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

(i) \((P, T, A, AF, TP)\) is a \textit{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 \( \bullet IP = 0 \),

(iv) the output place \( OP \) is the selected place \( OP \in (P \setminus RP) \) such that \( OP = \{OT\} \),

(v) \( RM_x \) is the set of all the allowed resource static markings,

(vi) the output transition \( OT \) is the selected transition \( OT \in T \) such that \( \bullet OT = \{OP\} \), \( TP(OT) = 1 \).
A SNT agent net \( AN = (P, T, A, AF, TP, RP, IP, OP, RM_s, OT) \) is said to be sound iff the SNT process Petri net \( APN = (P, T, A, AF, TP, RP, IP, OP, RM_s, M) \) associated with the SNT agent Petri net \( APN = (P, T, A, AF, TP, RP, IP, OP, RM_s, OT, M) \) 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 \( M_s \), where \( (\underline{x})_n \) means nonempty finite sequence of the \( n \) nonnegative integer numbers; that is, \( (\underline{x})_n \in (N^0)_S \), length\((\underline{x})_n\) = \( n \), \( n \in N \), and \( (i)_n \) means nonempty finite sequence of the \( n \) nonnegative integer numbers \( i \in N \), that is, \( (i)_n \in (N^0)_S \), length\((i)_n\) = \( n \), \( n \in N \). The BPN \( BASE1 \) is the simplest BPN at all and it contains only one place IP 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 \( PN_1 = (P_1, T_1, A_1, AF_1, TP_1, RP_1, IP_1, OP_1, RM_{s1}) \) and \( PN_2 = (P_2, T_2, A_2, AF_2, TP_2, RP_2, IP_2, OP_2, RM_{s2}) \) be the PNs, and let \( p \in P_1 \setminus RP_1 \) be a place of the PN \( PN_1 \) such that

\[
\begin{align*}
\left( (p \subseteq IP_2) \land (OP_2 \subseteq p) \right) \lor (p = p \equiv N) \lor (IP_1 = RP_2 = N).
\end{align*}
\]

The function PLACE-SUBST: \( PNET \times PNET \rightarrow PNET \) of the PN \( PN_2 \) substitution over the place \( p \) of the PN \( PN_1 \) will be denoted by \( PN_1.p[P_2]PN_2 \). The execution of this function results in the new PN \( PN = (P, T, A, AF, TP, RP, IP, OP, RM_s) \), where

\[
\begin{align*}
(i) \ & P = (P_1 \setminus \{ p \}) \cup P_2, \\
(ii) \ & T = T_1 \cup T_2, \\
(iii) \ & A = A_1 \cup \{ (x, y) \in A_1 \mid (x \neq p) \lor (y \neq p) \} \cup \{ (x, IP_2) \mid (x, p) \in A_1 \} \cup \{ (IP_2, y) \mid (y, p) \in A_1 \}, \\
(iv) \ & AF = AF_2 \cup \{ (x, y, v) \in AF_1 \mid (x, y) \neq p \lor (x, p) \neq \{ (x, IP_2, v) \} \cup \{ v, (x, IP_2, v) \} \cup \{ (x, v) \in AF_2 \} \cup \{ (v, y, v) \in AF_1 \}, \\
(v) \ & TP = TP_1 \cup TP_2, \\
(vi) \ & RP = RP_1 \cup RP_2, \\
(vii) \ & IP = IP_1, \text{ if } p \neq IP_1; IP = IP_2, \text{ if } p = IP_1, \\
(viii) \ & OP = OP_1, \text{ if } p \neq OP_1; OP = OP_2, \text{ if } p = OP_1, \\
(ix) \ & RM_s = RM_s1 \odot RM_s2.
\end{align*}
\]

**Example 2.** The result of the place-substitution operation \( SN1.P2 \setminus SN2 \) where the place \( P2 \) of the PN \( SN1 \) (see Figure 5) was substituted by the BPN \( SN2 \) can be shown in Figure 6. It is clear that all the necessary requirements for the place-substitution operation are satisfied in this case, that is, \( P2 \) is not a resource place of the PN \( SN1 \), \( P2 \subseteq \{ 4 \} \subseteq \{ 4 \} = 1P2, IP_2 = \{ 4 \} \subseteq \{ 4 \} = 2P2. \) It is easy to verify that in this case the resulting PN \( SN1.P2 \setminus SN2 \) is also sound.

**Lemma 3.** Let \( PN1 = (P_1, T_1, A_1, AF_1, TP_1, RP_1, IP_1, OP_1, RM_{s1}) \), \( PN2 = (P_2, T_2, A_2, AF_2, TP_2, RP_2, IP_2, OP_2, RM_{s2}) \), and \( PN3 = (P_3, T_3, A_3, AF_3, TP_3, RP_3, IP_3, OP_3, RM_{s3}) \) be the three PNs, let \( p_1 \in P_1 \) be a place of the PN \( PN1 \) such that \( p_1 \neq RP_1 \), \( (\{ p_1 \subseteq IP_2 \} \land (OP_2 \subseteq p_1)) \lor (p_1 = p_1 \equiv N) \lor (IP_2 = OP_2 \equiv N) \), and let \( p_2 \in P_2 \) be a place
of the PN \( PN2 \), such that \( p_2 \not\in RP_2 \), \((\langle x_2 \rangle \leq \langle IP_3 \rangle \land \langle OP_3 \rangle \leq p_2)\) \lor \((\langle x_2 \rangle = p_2) = N\) \lor \((\langle IP_3 \rangle = \langle OP_3 \rangle = N\). Then

\[
PN1.p1 \mid (PN2.p2 \mid PN3) = (PN1.p1 \mid (PN2).p2 \mid PN3.
\]

Proof. From Definition 1 it directly follows that

\[
PN1.p1 \mid (PN2.p2 \mid PN3) = (PN1.p1 \mid (PN2).p2 \mid PN3 = (P,T,A,AF,TP,RP,IP,OP,RM_1) where
\]

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

(iii) \( A = A_3 \cup \{(x_2,y) \in A_2 \mid (x \neq p_2) \lor (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_3 \cup \{(x_2,y) \in A_1 \mid (x \neq p_1) \lor (y \neq p_1)\} \cup \{(x,OP_3,\langle x_2 \rangle) \mid (x,p_1) \in A_1 \cup \{(OP_3,y) \mid (p_1,y) \in A_2,\}

(iv) \( AF = AF_1 \cup \{(x,y),v) \in AF_2 \mid (x \neq p_2) \lor (y \neq p_2)\} \cup \{(x,IP_3),v) \mid (x,p_2),v) \in AF_2 \cup \{(OP_3,y) \mid (p_2,y),v \in AF_2 \cup \{(x,y),v) \in AF_1 \mid (x \neq p_2) \lor (y \neq p_2)\} \cup \{(x,IP_3),v) \mid (x,p_2),v) \in AF_2 \cup \{(OP_3,y) \mid (p_2,y),v \in AF_1 \}

(v) \( TP = TP_1 \cup TP_2 \cup TP_3, \)
(vi) \( RP = RP_1 \cup RP_2 \cup RP_3, \)
The triple \((\text{PNET}, \text{PLACE-SUBST}, \text{BASE1})\) is a monoid.

Lemma 4. The triple \((\text{PNET}, \text{PLACE-SUBST}, \text{BASE1})\) 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 = \((P_1, T_1, A_1, AF_1, TP_1, RP_1, IP_1, OP_1, RM_{a1}, OT_1)\) be the AN and PN2 = \((P_2, T_2, A_2, AF_2, TP_2, RP_2, IP_2, OP_2, RM_{a2})\) the ANs, and let \(p \in P_1 \setminus RP_1\) be a place of the AN AN1 such that

\[
\left(\left(\left(p \subseteq \text{IP}_2\right) \land \left(\text{OP}_2 \subseteq \text{p}\right)\right) \lor \left(\left(p = \text{p} = \text{N}\right)\right)\right) \lor \left(\left(\text{IP}_2 = \text{OP}_2 = \text{N}\right)\right).
\]

(4)

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

(i) \(P = (P_1 \setminus \{p\}) \cup P_2\),

(ii) \(T = T_1 \cup T_2\),

(iii) \(A = A_2 \cup \{(x, y) \in A_1 \mid (x = p) \lor (y = p)\} \cup \{(x, IP_2) \mid (x, p) \in A_1\} \cup \{(OP_2, y) \mid (p, y) \in A_1\}\),

(iv) \(AF = AF_2 \cup \{(x, y, v) \in AF_1 \mid (x = p) \lor (y = p)\} \lor \{(x, IP_2, v) \mid (x, p) \in AF_1\} \lor \{(OP_2, y, v) \mid (p, y, v) \in AF_1\}\),

(v) \(TP = TP_1 \cup TP_2\),

(vi) \(RP = RP_1 \cup RP_2\),

(vii) \(IP = IP_1\) if \(p \not\subseteq \text{IP}_1; IP = IP_2\) if \(p = \text{IP}_1\),

(viii) \(OP = OP_1\) if \(p \not\subseteq \text{OP}_1; OP = OP_2\) if \(p = \text{OP}_1\),

(ix) \(RM_{a} = RM_{a1} \otimes RM_{a2}\),

(x) \(OT = OT_1\).

Definition 5 is similar to Definition 1 in the sense of overloading of the operator \(|: \text{PNET} \times \text{PNET} \rightarrow \text{PNET}\) with the operator \(|: \text{ANET} \times \text{PNET} \rightarrow \text{ANET}\) of the AN PN2 substitution over the place \(p\) of the AN AN1 which is denoted by AN1.p\(\downarrow\)PN2. It is easy to formulate similar lemmas like Lemmas 3 and 4 for the case of this overloaded operator \(|\).

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 \(RP, p \not\subseteq RP\). Figure 7 shows the result of the resource place \(R_1\) (that contains the token in the nonempty static marking \(M_{s}\) of the sound net SN3 substitution by the sound net SN4, that is, the net SN3.R1 \(\downarrow\) SN4. The resulting net then does not satisfy the necessary conditions for its static marking \(M_{s}\), because the token \(1\) is associated in this static marking \(M_{s}\) with the nonresource place \(P_2\), and the transition \(T_2\) is then enabled in this static marking \(M_{s}\) which is not allowed. If we remove the token \(1\) from the place \(T_2\) 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 \(P_2\) of the sound net SN5 substitution by the sound net SN6, that is, the PN SN5.P2 \(\downarrow\) SN5, that is not sound. It can be seen for its input marking \(M_{s}\), where \(M_{s}(IP_1) = 1\). In this input marking \(M_{s}\) the transition \(T_1\) is enabled and its firing results into the marking \(M_{s}\), where \(M_{s}(T1) = M_{s}\). There is token \(3\) in the place \(P_1\) and the token \(1\) in the place \(P_2\) in the marking \(M_{s}\), and the transitions \(T_2\) and \(T_5\) are enabled in parallel in the marking \(M_{s}\) and after their parallel firing the PN results in the marking \(M_{s}\), where \(M_{s}(T2 \ T5)M_{s}\). In the marking \(M_{s}\) there will be the token \(1\) in the place \(OP_2\) and the token \(3\) in the place \(OP_1\). No transition is enabled in the nonoutput marking \(M_{s}\) and so the resulting net SN5.P2 \(\downarrow\) SN6 is not sound.

The loss of the net soundness then follows from the existence of the arc \((P_1, T_2)\) in the resulting net. The necessary condition for the soundness of the resulting net is in this case the following:

\[
\forall t \in P_2: (t \in \{u \mid (u \in \text{IP}_1 \lor (u \in \text{IP}_2)\}) \lor (t \in \{q \mid (q \in \text{IP}_1)\})\]

and it does not hold for the place \(P_2\) of the PN SN5, because transition \(T_3\) in \(P_2\) holds that \((P_1 \in \{T3\} \land \{P_1 \neq \text{IP}_1\})\) and \((P1+ = \{T2, T3\})\). It is then clear that if the arc \((P_1, T_2)\) is not present then the resulting net is sound.

Figure 9 shows the result of the place \(P_2\) of the sound net SN7 substitution by the sound net SN8, that is, the SNT process net SN7.P2 \(\downarrow\) SN8, that is again not sound. The loss of the soundness (it can be easily seen for any input marking \(M_{s}\) of the resulting PN) then follows from the fact that \(P_2\) = \(\{1, 4\}\); that is, \(|P_2| > 1\). The necessary condition for the soundness of the resulting PN is in this case the following:

\[
|P_2| = 1.
\]

Lemma 6. Let PN1 = \((P_1, T_1, A_1, AF_1, TP_1, RP_1, IP_1, OP_1, RM_{a1})\) and PN2 = \((P_2, T_2, A_2, AF_2, TP_2, RP_2, IP_2, OP_2, RM_{a2})\) be two sound ANs, and let \(p \in P_1 \setminus RP_1\) be a place of the PN PN1 such that \((\langle p \subseteq \text{IP}_2\rangle \land (\text{OP}_2 \subseteq \text{p})\) \lor (\langle p = \text{p} = \text{N}\rangle)\) \lor (\langle p = \text{p} = \text{N}\rangle) \lor (\langle q = \text{q} = \text{I}\rangle)\) \lor (\langle q = \text{q} = \text{I}\rangle). Then the PN PN1.p \(\downarrow\) PN2 is also sound.

Comment. If the conditions \(p \in P_1 \setminus RP_1\) hold, then \((\langle p \subseteq \text{IP}_2\rangle \land (\text{OP}_2 \subseteq \text{p})\) \lor (\langle p = \text{p} = \text{N}\rangle)\) \lor (\langle p = \text{p} = \text{N}\rangle) \lor (\langle q = \text{q} = \text{I}\rangle)\) \lor (\langle q = \text{q} = \text{I}\rangle)\) follows for the substituted place \(p \in P_1\) of the net PN1 are not satisfied the resulting net PN1.p \(\downarrow\) PN2 does not need to be sound generally as follows from the previous three examples.

Proof. It is clear for \(p = \text{IP}_1\) or \(p = \text{OP}_1\) and it follows directly from the definition of the sound PN. Let \(p \not\subseteq \text{IP}_1\) and \(p \not\subseteq \text{OP}_1\). Because the PN PN1 is sound, it holds that \(V M_{s} \exists q \in N \forall q \in P_1 \forall M \in \{M_{s}\} : M(q) \leq k\). This holds also for the substituted
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 $PN1.p \parallel PN2$ 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 $PN1.p \parallel PN2$ 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 $\forall M \in \{M_i\}_i$, $M(p) \neq 0$, and $\exists M_n : M_n \in \{M\}$ and for the PN $PN2$ holds true that $\forall M \in \{M\}_i \exists M_n : M_n \in \{M\}$. The conditions $|p| = 1$, $\forall t \in p^* : (q \in t) \land (q \not\in IP_2)$ are satisfied for the PN $PN1$ and from these conditions follows that $\exists OP_2 : ((q \in t) \land (q \not\in IP_2)) \Rightarrow (f = \{t\})$ in the resulting net $PN1.p \parallel PN2$. Then it follows that there must exist transition $t$, $t \in OP_2^*$, and the marking $M$ of the resulting net $PN1.p \parallel PN2$ such that $(M(\overline{OP_2}) \neq \emptyset) \land (t \in M)$, because the PN $PN1$ terminates properly. So the whole PN $PN1.p \parallel PN2$ must also terminate properly for any of its input markings $M_i$ and it is sound.

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 $AN1 = (P_1, T_1, A_1, AF_1, TP_1, RP_1, IP_1, OP_1, RM_{AN1}, OT_{AN1})$ be the sound AN and $PN2 = (P_2, T_2, A_2, AF_2, TP_2, RP_2, IP_2, OP_2, RM_{PN2})$ the sound PN, and let $p \in P_1 \setminus RP_1$ be a place of the AN $AN1$ such that $((p \subseteq \overline{IP_2}) \land (OP_2 \subseteq p)) \lor ((p = p^* = N) \lor (IP_2 = OP_2 = N))$, $|p| = 1$, $\forall t \in p^* : (q \in t) \land (q \not\in IP_2) \Rightarrow (f = \{t\})$. Then the AN $AN1.p \parallel PN2$ is also sound.

**Proof.** Proof directly follows from the proof of Lemma 6.

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

(i) $P = P_1 \cup P_2$,
(ii) $T = T_1 \cup T_2$,
(iii) $A = A_1 \cup A_2 \cup \{\langle t_1, IP_2 \rangle \} \cup \{\langle t_2, IP_2 \rangle \} \cup \cdots \cup \{\langle t_n, IP_2 \rangle \}$,
(iv) $AF = AF_1 \cup AF_2 \cup \{\langle (t_1, IP_2), k \rangle \} \cup \{\langle (t_2, IP_2), m \rangle \} \cup \cdots \cup \{\langle (t_n, IP_2), n \rangle \}$, where $k, m, n, \ldots \in \mathbb{I}P_2$,
(v) $TP = TP_1 \cup TP_2$,
(vi) $RP = RP_1 \cup RP_2$,
(vii) $IP = IP_1$,
(viii) $OP = OP_1$,
(ix) $RM = RM_{AN1} \otimes RM_{AN2}$.

**Example 9.** The result of the asynchronous-processing operation $PN1.T1.T2 \overset{\text{ASYNC-PROC}}{\rightarrow} AN2$ 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 $PN1.T1.T2 \overset{\text{ASYNC-PROC}}{\rightarrow} AN2$ is also sound.

**Lemma 10.** Let $PN1 = (P_1, T_1, A_1, AF_1, TP_1, RP_1, IP_1, OP_1, RM_{PN1})$ be a PN, $AN2 = (P_2, T_2, A_2, AF_2, TP_2, RP_2, IP_2, OP_2, RM_{AN2}, OT_{AN2})$ and $AN3 = (P_3, T_3, A_3, AF_3, TP_3, RP_3, IP_3, OP_3, RM_{AN3}, OT_{AN3})$ are two ANs, let $t_1, t_2, \ldots, t_n \in T_1$ be the transition of the PN $PN1$, $n \in \mathbb{N}$, and $t_1, t_2, \ldots, t_m \in T_2$, $m \in \mathbb{N}$, are the transition of the AN $AN2$. Then

$$PN1.t_1.t_2.\ldots.t_n \overset{\text{ASYNC-PROC}}{\rightarrow} AN2.t_1.t_2.\ldots.t_m$$

(5)

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

**Lemma 11.** The couple $(PNET \cup ANET, ASYNC-PROC)$ is a semigroup.
Figure 8: Result of place-substitution operation $SN5. P2 \{ T \}$ $SN6$. 

**Definition 12.** Let $AN1 = (P_1, T_1, A_1, AF_1, TP_1, RP_1, IP_1, OP_1, RM_1, OT_1)$ and $AN2 = (P_2, T_2, A_2, AF_2, TP_2, RP_2, IP_2, OP_2, RM_2, OT_2)$ be the ANs, and let $t_1, t_2, \ldots, t_n \in T_1, n \in \mathbb{N}$, 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 $AN1t_1, t_2, \ldots, t_n \_ AN2$. The execution of this function results in the new AN $AN = (P, T, A, AF, TP, RP, IP, OP, RM, OT)$, where

(i) $P = P_1 \cup P_2$,  
(ii) $T = T_1 \cup T_2$,  
(iii) $A = A_1 \cup A_2 \cup \{(t_1, IP_2)\} \cup \{(t_2, IP_2)\} \cup \cdots \cup \{(t_n, IP_2)\}$,  
(iv) $AF = AF_1 \cup AF_2 \cup \{(t_1, IP_2), k\} \cup \{(t_2, IP_2), m\} \cup \cdots \cup \{(t_n, IP_2), n\}$, where $k, m, n, \ldots \in IP_2$,  
(v) $TP = TP_1 \cup TP_2$,  
(vi) $RP = RP_1 \cup RP_2$,  
(vii) $IP = IP_1$,  
(viii) $OP = OP_1$,  
(ix) $RM = RM_1 \otimes RM_2$  
(x) $OT = OT_1$.

**Definition 12** is similar to **Definition 8** in the sense of the overloading of the operator $\cdot:\ PNET \times ANET \rightarrow PNET$ with the operator $\cdot:\ ANET \times ANET \rightarrow ANET$ of the AN $AN2$ asynchronous-processing from the AN $AN1$. It is easy
to formulate similar lemmas like Lemmas 10 and 11 for the case of this overloaded operator \( \cdot \) - .

**Lemma 13.** Let \( PN_1 = (P_1, T_1, A_1, AF_1, TP_1, RP_1, IP_1, OP_1, RM_{1j}) \) be a sound PN and \( AN_2 = (P_2, T_2, A_2, AF_2, TP_2, RP_2, IP_2, OP_2, RM_{2i}, OT_2) \) a sound AN, and let \( t_1, t_2, \ldots, t_n \in T_1, n \in \mathbb{N} \), be the transitions of the PN \( PN_1 \). Then the PN \( PN_1.t_1.t_2.\cdots.t_n \cdot AN_2 \) is also sound.

**Proof.** Clear. Because the PN \( PN_1 \) is sound, it holds that \( \forall M_i \exists k \in \mathbb{N} \forall q \in P_i \forall M \in [M_i] : M(q) \leq k \). It also holds that \( \forall M_i \exists t_1 t_2 \cdots t_m : M_i[t_1] M_i[t_2] M_2 \cdots [t_m] M_x, m \in \mathbb{N}, t_1, t_2, \ldots, t_m \in T_i \). From these facts follows that every of the transition \( t_1, t_2, \ldots, t_n \in T_1, n \in \mathbb{N} \), will be fired finite times in every transition sequence \( M_i[t_1] M_i[t_2] M_2 \cdots [t_m] M_x \).

From these immediately follows that in the resulting PN \( PN_1.t_1.t_2.\cdots.t_n \cdot AN_2 \) is satisfied the condition \( \forall M_i \exists k \in \mathbb{N} \forall M \in [M_i] : M(IP_2) \leq k \). Because of the soundness of the AN \( AN_2 \) the whole PN \( PN_1.t_1.t_2.\cdots.t_n \cdot AN_2 \) must be bounded. Because the PN \( PN_1 \) terminates properly for any of its input markings \( M_i \) and also the AN \( AN_2 \) terminates properly and their firing of the transitions are independent it is clear that also the resulting PN \( PN_1.t_1.t_2.\cdots.t_n \cdot AN_2 \) 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.
with the object lens El8-200 mm to eliminate the problem of insufficient lighting in lecture rooms during recording (see Figure II), 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 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.
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 nonscanned area of the lecture room),
2. exit of the speaker outside the **general-view** area (into the nonscanned 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 nonscanned 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 **IPI** of the PN **ROOM1** in its input marking \( M_i \). The input place **IPI** can contain 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 nonscanned 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 nonscanned 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...
of the infinite transitions sequence $T_2 \cdot T_2 \cdots$ 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 $ROOM_1$ is sound for every one of its input markings $M_i$.

The PN $ROOM_2$ 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 (3). The entry into the detail-view area by the individual speaker and sending of the message represented by the token (5) 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 (6)) and returning again into the detail-view area (firing of the transition T13 and sending of the message (5)). Every such speaker is then represented by the token (30) after the firing of the transition T14 and sending of the message represented by the token (4) 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 Mj.

PN ROOM that models all the main functionalities of the audiovisual lecture room is then the PN \( \text{ROOM} = \text{ROOM}.P1 \) (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 (1)–(6) 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 CAMI in Figure 15. The input place IP3 of the AN CAMI accepts all the messages represented by the tokens (1)–(6) and generated by the area sensors of the audiovisual lecture room. Resource place R20 with the token (0) of the only one allowed static marking Mj of this AN (this static marking Mj is such the only-one element of the set \( M_s \)) of the AN CAMI 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 CAMI is sound for every one of its input markings Mj.

Figure 16 shows the resulting PN RNET1 that is composed of the PN \( \text{ROOM} = \text{ROOM}.P1 \) (ROOM2 and the AN CAMI using the ASYNC-PROC operator such that \( \text{RNET1} = \text{ROOM}.T11.T10.T11.T12.T13.T14 - | \text{CAMI} \). 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 (1)–(6) 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, detail-view, or area{s} at the given time. Every (anonymous) speaker that is presenting in the general-view, board-view, detail-view areas, respectively, at the given time is then modeled by the tokens (1), (2), and (3), 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 \( \text{RM}_s = \{(i' \cdot 1 + j' \cdot 2 + k' \cdot 3) | i \in N_0, j \in N_0, k \in N_0\} \). The functionalities of the PN CAM2 are as in the following:

(i) Acceptance of the message (1) (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 (1) (entry of the speaker into the general-view area) into the resource place R30.

(ii) Acceptance of the message (3) (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 (1) (exit of the speaker outside the general-view area) from the resource place R30 and adding the token (2) (entry of the speaker into the board-view area) into the resource place R30.

(iii) Acceptance of the message (5) (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 the token (2) (exit of the speaker outside the board-view area) from the resource place R30, and adding the token (3) (entry of the speaker into the detail-view area) into the resource place R30.

(iv) Acceptance of the message (2) (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 (1) (exit of the speaker outside the general-view area into the nonscanned area) from the resource place R30. If

![Figure 15: SNT agent net CAMI.](image-url)
the resource place R30 still contains at least one token (1) (i.e., at least one speaker is still presented in the general-view area) the transition T35 then fires and the token (2) is changed into the token (0) (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 (4) (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 (2) (exit of the speaker outside the board-view area) from the resource place R30 and adding of the token (1) (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 (2) (i.e., at least one speaker is still presented in the board-view area) the transition T35 then fires and the token (4) is changed into the token (0) (i.e., no camera switching will be performed). The transition T34 will fire otherwise and the camera switching can be performed.
(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 \text{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 \text{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 \text{T34} will fire otherwise and the camera switching can be performed.

(vii) The transition \text{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 \( M \) 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 \( \text{RM}_{\text{PN}} = \{(k) \mid k \in \{0, 1, 2, 3\}\} \). 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 nonscanned area of the lecture room) results in the firing of the transition \text{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 \text{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 nonscanned area of the lecture room) results in the firing of the transition \text{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 \text{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 \( \text{RNET}' \) with the support of the PLACE-SUBST operator applications such that

\[
\text{RNET} = (\text{RNET1}.P20 \mid \text{CAM2}).OP4 \mid \text{CAM3}, \quad (6)
\]

where \( \text{RNET1} = \text{ROOM}.T1.T3.T10.T11.T12.T13.T14 \mid \text{CAM1} \) and \( \text{ROOM} = \text{ROOM1}.P1 \mid \text{ROOM2} \); that is,

\[
\text{RNET} = (((\text{ROOM1}.P1 \mid \text{ROOM2}).T1).T3.T10.\text{T11}.\text{T12}.\text{T13}.\text{T14} \mid \text{CAM1}).P20 \mid \text{CAM3}). \quad (7)
\]

OP4 \mid \text{CAM3}.
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 SNTPN 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 SNTPNs [18]).

There is an opportunity to define additional operators for the class of SNTPNs and to generalize it (for instance, the SYNC-PROC composition operator that models synchronous nonrecursive calling of chosen method represented by a PN $PN_2$ of some programming system from the body of another method of the same programming system represented by a PN $PN_1$). Selected PN or AN then represents typically a method in the model of the programming system realized with the support of the SNTPN 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


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