https://doi.org/10.24846/v26i2y201703

Conception of Hierarchical Fuzzy Interpreted PETRI Net

Michał MARKIEWICZ*, Lesław GNIEWEK

[1] Rzeszow University of Technology, Departament of Computer and Control Engineering,
al. Powstancow Warszawy 12, Rzeszow, 35-959, Poland

mmarkiewicz@prz.edu.pl (*Corresponding author*); lgniewek@prz.edu.pl

Abstract: This article presents a conception and a formal definition of hierarchical fuzzy interpreted Petri net as a representation of a flat net. This solution allows to create control systems using techniques of a composition, a decomposition and their combination. The proposed formal description has been implemented in the computer simulator called FIPN-SML, earlier used to model flat nets.

Keywords: Petri nets, fuzzy nets, interpreted nets, computer simulator, control system.

1. Introduction

Petri nets (PNs) for more than 50 years have become useful tools that can be used to model control, decision making and diagnosis systems, that are sequential or concurrent. The main advantages of PNs are mathematical formalism, concurrency and the possibility of using a graphic form of their presentation. Due to the numerous applications of PNs to create informatics systems and to enhance these systems complexity, designers have applied a new approach to the description of PNs operation, analysis and presentation. They have implemented a hierarchical structure into formal descriptions of PNs (similar to object-oriented programming languages), modularization and mechanisms based on a hierarchy, that allow to evaluate the nets properties on the basis of their reduced versions. Within PNs there is no unequivocal approach to the use of a hierarchy. Based on the literature several basic trends could be distinguished [1-7], [10-15], [17-23]: refinement of places and transitions; the definition of a hierarchical net; the definition of an object-oriented net; a reduction technique; application of a hierarchical approach as the methodology for dealing with large systems; a hierarchical net as a standard.

The first trend - refinement of places and transitions - consists of replacing a part of the net using a transition or a place, and then moves this part into the newly created node [1, 17, 19], so called macroplace or macrotransition. To create new nodes it is required to have certain properties of separated parts of the net. This allows to present the graphical diagram of the net in different levels of abstraction and to examine the system on the basis of its smaller fragment.The second mentioned direction is to create the formal description and the definition of hierarchical PN [3, 11, 13]. Besides the imposition of certain restrictions preliminary such as the number of inputs or outputs of a subnet, it is possible to create a subnet of any hierarchical structure and any properties.

The definitions of the object-oriented PNs can be considered as a separate approach to the hierarchy. This class of PNs has features of object-oriented programming languages, for example inheritance or morphism [12, 14, 20]. Like the previous trend, this can be given in the form of a hierarchical definition. However, the part or the supplement of this definition is given in pseudo code or programming language code, in which they are implemented in order to show their object-oriented features.

Another way of the hierarchical approach is the reduction [4, 6, 7, 15]. This technique allows to determine dynamic and static properties of the net on the basis of its counterpart to which it has been transformed. The transformation is done according to specific rules that are applied in the specific order.

A separate direction of a hierarchical approach is to develop a methodology for dealing with large systems e.g. in production environments [21, 22]. This approach is usually a general description given in the verbal form supplemented with examples and formal descriptions.

The last mentioned trend related with the hierarchical structure in PNs is their presentation as a standard. Such an example is Grafcet [18], that is used in industrial systems. It uses several hierarchical mechanisms, including macrosteps. The macrostep allows to divide the whole net into separate subnets with a single input and output step (place).

The boundary between all trends is not clear and they can complement each other, e.g. an important part of the methodology for dealing with large systems can be the reduction technique [21]. Additionally, within these systems development can be distinguished a decomposition technique, a composition technique, a combination thereof, the methods of bottom-up and top-down [5, 7, 23]. The decomposition and the composition will be discussed later in Section 4.

Analyzing the above examples, it can be seen that the hierarchical approach greatly facilitates the practical use of PNs. They can be applied in many ways, e.g. to create control systems [2, 10]. One of the few fuzzy nets that can be used to create such systems is the fuzzy interpreted Petri net (FIPN) described in [8, 9]. The main advantage of the FIPN in comparison to discrete PNs is that it allows to get a better visualization of the system dynamics. The application of analog sensors and the graphical representation of the net allows in a better way to observe the progress in transferring the markers between the places. However the lack of hierarchical structure limits the practical use of the FIPN.

The aim of this article is to present the conception and the formal definition of the hierarchical FIPN (HFIPN) as the representation of the flat net. The second section presents the formal basis of the FIPN that is used to define the HFIPN in the third section. The section 4- section 6 describe the example of the HFIPN creation through the decomposition (section 4), its algebraicrepresentation (section 5), and the practical use of this net (section 6). The proposed solution facilitates dividing the flat net into some number of subnets using the concept of the macroplace with one input and one output place. All the pictures in this article come from the simulator FIPN-SML [16] in which model HFIPN has been implemented.

2. The formal basis of the FIPN

Section 2 is entirely based on [8] that shows the FIPN formalism and its conception. All definitions presented here are used later to define the operation and the structure of HFIPN in section 3. Three definitions show the FIPN structure and the principle of its operation. The first definition describes the net construction in the following manner.

Definition 1. The fuzzy interpreted Petri net is the system [8] $FIPN = (P, T, \mathit{\Omega}, \mathit{\Psi}, R, \mathit{\Delta}, K, W, \mathit{\Gamma}, \mathit{\Theta}, M_0, e)$, where $P = P' \cup P''$ – a nonempty finite set of places; $P' = \{p'_1, p'_2, \ldots, p'_{a'}\}$ – a set of places for modeling processes; $P'' = \{p''_1, p''_2, \ldots, p''_{a''}\}$ – a set of places for modeling resources; $T = \{t_1, t_2, \ldots, t_b\}$ – a nonempty finite set of transitions; $\mathit{\Omega} = \{\omega_1, \omega_2, \ldots, \omega_{a'+a''}\}$ – a nonempty finite set of statements; $\mathit{\Psi} = \{\psi_1, \psi_2, \ldots, \psi_b\}$ – a nonempty finite set of conditions; P, T, $\mathit{\Omega}$, $\mathit{\Psi}$ – where none of these sets have common elements; $R \subseteq (P \times T) \cup (T \times P)$ – the incidence relation that assigns to each transition $t_i \in T (1 \le i \le b)$ a place, where there is a place $p' \in P'$ such that $(p', t_i) \in R$ or $(t_i, p') \in R$; $\mathit{\Delta}: P \to \mathit{\Omega}$ – the function that assigns to each place a statement; $K: P' \to 1$ and $P'' \to \aleph \backslash \{1\}$ – the function that assigns to each place its capacity, where $\aleph = \{1, 2, \ldots\}$; $\mathit{\Gamma}: T \to \mathit{\Psi}$ – the function that assigns to each transition a condition; $\mathit{\Theta}: T \to [0, 1]$ – the function that defines the degree to which the conditions corresponding to the transitions t are satisfied; $W: R \to \aleph$ – the weight function that meets two conditions: $W(p,t) \le K(p)$ and $W(t,p) \le K(p)$ (p means p' or p'');

$M_0 : P' \to \{0,1\}$ i $P'' \to W_+$ – the initial marking function, where $M_0(p''_j) = z_j / K(p''_j)$, $z_j \in \aleph \cup \{0\}$, $z_j \le K(p''_j)$, for $1 \le j \le a''$; W_+ – the set of non-negative rational numbers; e –the event that synchronizes work of all transitions.

Two sorts of places can be distinguished in the FIPN (Figure 1). Their markings are numbers from the range [0,1]. The marking of the places that can hold a greater than one number of tokens is presented as a fraction. For these places the capacity meets condition $K(p) > 1$ and appears in the denominator. It plays the role of the factor that normalizes the value of the marking into the range [0,1]. Moreover, additional conditions are assigned to synchronized transitions and the weights of arcs are integers greater than or equal to one.

The transfer of a marker through a transition can be initiated when the conditions (1) by Definition 2 is satisfied and it is finished when condition (2) becomes true.

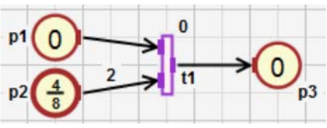

Figure 1. The net with the different types of places connected to the transition t_1.

Definition 2. The transition $t \in T$ with marking M: $P \to [0, 1]$ is enabled from the moment in which the degree of fulfill the condition $\Theta(t) = \vartheta$, which it is assigned to the transition, is greater than zero and the following conditions are satisfied [8]:

$\forall p \in {}^{\bullet}t, M(p) \geq W(p,t)/K(p)$ and

$$\forall p \in t^{\bullet}, M(p) \leq 1 - W(t,p)/K(p), \qquad (1)$$

to the moment in which:

$$\exists\, p' \in {}^{\bullet}t, M(p') = 0 \text{ or } \exists\, p' \in t^{\bullet}, M(p') = 1, \qquad (2)$$

where ${}^{\bullet}t = \{p \in P \,|\, (p,t) \in R\}$ – it is a set of input places of transition t, and $t^{\bullet} = \{p \in P \,|\, (t,p) \in R\}$ – it is a set of its output places.

The new marking for the input and output places connected to the active transition is depended on the increment of a degree to which the condition corresponding to the transition is satisfied. The method for the calculation of the new marking shows Definition 3. The transition is enabled until the transfer of the markers from the input places to the output places is completed.

Definition 3. Let M be the marking for which transition $t \in T$ is enabled, the degree $\Theta(t) = \vartheta \in [0,1]$ to which the condition corresponding to the enabled transition is satisfied will be changed by $\Delta\vartheta \geq 0$ and there will be an event e which synchronizes the work of all transitions. New marking of the net M' is computed according to the following rule [8]:

$$M'(p) = \begin{cases} M(p) - \dfrac{\Delta\vartheta \cdot W(p,t)}{K(p)} \text{ for } p \in {}^{\bullet}t \setminus t^{\bullet}, \\ M(p) + \dfrac{\Delta\vartheta \cdot W(t,p)}{K(p)} \text{ for } p \in t^{\bullet} \setminus {}^{\bullet}t, \\ M(p) - \dfrac{\Delta\vartheta \cdot [W(p,t) - W(t,p)]}{K(p)} \text{ for } p \in {}^{\bullet}t \cap t^{\bullet}, \\ M(p) \text{ for } p \notin {}^{\bullet}t \cup t^{\bullet}. \end{cases} \qquad (3)$$

where increment $\Delta\vartheta < 0$ does not introduce any changes in marking of the net.

In Figure 2 the fired transition t_1 and the marking change in its input and output places is shown. Assuming that at the time of the arrival of the synchronization signal the increment of the degree to which the condition corresponding to the transition t_1 is satisfied has the value $\Delta\vartheta = 0.1$. Such a change of marking gives a better visualization (compared to discrete PNs) and it allows to monitor the progress in the transfer of tokens between input and output places connected to transitions.

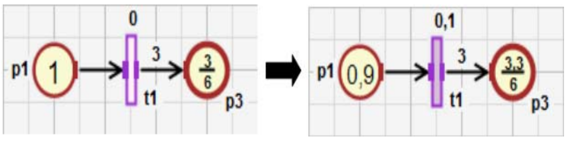

Figure 2. The new marking of the places connected to the transition t_1 for $\Delta\vartheta = 0.1$.

3. The formal definition of the HFIPN

The concept of the macroplace with one input and output place has been used to define a hierarchical net and it is similar to [17, 18, 19].However, the proposed solution is more general and does not change the operation of the entire net as in the case of [18, 19] and is not only limited to create macronets with certain properties [17, 19]. In the proposed net input place cannot be output place, what allows to get a better movement's visualization of the marker by macroplace. In addition, it does not reduce the possibility of creation the macroplace, that has the input and output place of different capacity. Thanks to the use of hierarchical structure, it is possible to divide the flat net into subnets and to facilitate an overall analysis ofthe net through a separate testing of its fragments.

Figure 3 illustrates an exemplary hierarchical net that consists of two subnets N_0 and N_1. The subnet N_0 contains the macroplace mp_1 that includes the subnet N_1. The macroplace mp_1 is internal macroplace of the subnet N_0 and is overriding macroplace for the subnet N_1. The input place (p'_2) and the output place (p''_1) of the macroplace (mp_1) are also the input and the output place of the subnet (N_1) for which this macroplace (mp_1) is overriding.

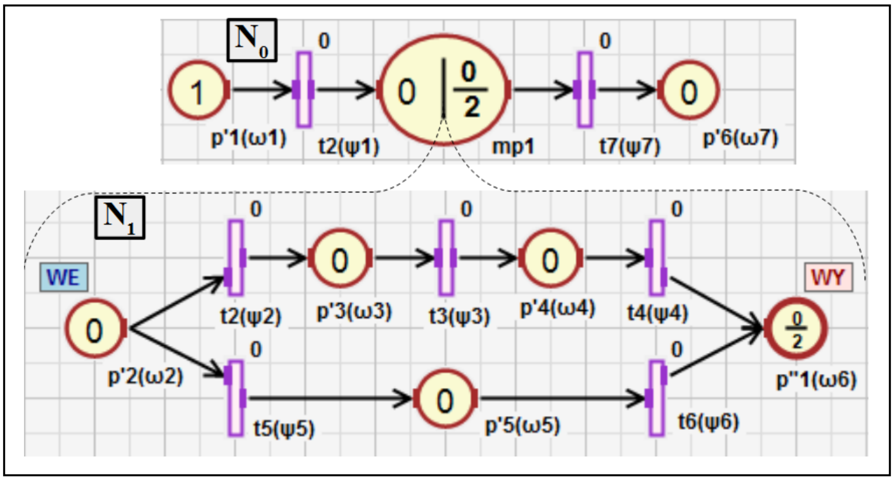

Figure 3. Example of net with macroplace.

In the assumption the HFIPN is a hierarchical representation of the flat FIPN, therefore it is defined with the use of the presented model in section 2. This definition consists of 3 parts: the general definition, the definition of subnets in the HFIPN and the definition of their relationships. They are closely interlinked and complement one another. Furthermore, they should be analyzed indissociably. Symbols used in those definitions: P, T, Ω, Ψ, R, Δ, K, W, Γ, Θ, M_0, e , $\aleph$, W_+ , have the same meanings as in the Definition 1. The general definition is as follows.

Definition 4. Hierarchical FIPN is the system $HFIPN = (N, MP)$, where

$N = \{N_0, N_1, \ldots, N_d\}$ - a nonempty finite set of subnets described in the Definitions 5 and 6;

$MP = \{mp_1, mp_2, \ldots, mp_d\}$ - a finite set of macroplaces, such as $\forall mp_k \in MP$: $mp_k = \{N_k, p_{in(k)}, p_{out(k)}\}$ where

$N_k \in N$ ($1 \le k \le d$) - a subnet for which the macroplace mp_k is overriding,

$p_{in(k)}$ - an input place of the macroplace and the subnet N_k ,

$p_{out(k)}$ - an output place of the macroplace and the subnet N_k ,

$p_{in(k)} \neq p_{out(k)}$ - the input place cannot be simultaneously the output place.

It is considered to be the new type of places in a net because of the incidence relationship. The input place (p'_2) of the macroplace (mp_1) cannot have outgoing arcs from it to a transition in the subnet (N_0) for which this macroplace is internal. On the other hand, the output place (p''_1) of the macroplace (mp_1) may have only outgoing arcs in subnet (N_0) for which this macroplace is internal.

The consequence of the Definition 4 is that each subnet with the exception of N_0, has exactly one overriding macroplace. Furthermore, HFIPN is divided into d + 1 subnets. If $d = 0$ then the net is flat. However, if $d > 0$ then the net is hierarchical. The subnet N_0 is called the main subnet or the subnet of the highest level. The remaining subnets are called the subnets of lower level or macronets. All subnets are defined in the following manner.

Definition 5. Each subnet $N_i \in N, 0 \le i \le d$ is defined as follows: N_i = (MP_i, P_i, T_i, Ω_i, Ψ_i , R_i , Δ_i , K_i , W_i , Γ_i , Θ_i , $M_{0(i)}$, e) where

$MP_i \subseteq MP$ – a finite set of internal macroplaces in the subnet N_i ;

$P_i \subseteq P$ and $P_i = P'_i \cup P''_i \cup P_{in(i)} \cup P_{out(i)}$ - a nonempty finite set of places (in N_i):

$P'_i \subseteq P'$ - a set of p'-type places,

$P''_i \subseteq P''$ - a set of p''-type places,

$P_{in(i)} = P'_{in(i)} \cup P''_{in(i)}$ - a set of input places (p'- type or p''-type) of internal macroplaces from the set MP_i, $P_{out(i)} = P'_{out(i)} \cup P''_{out(i)}$ - a set of output places (p'- type or p''-type) of internal macroplaces from the set MP_i, $P'_i, P'_{in(i)}, P'_{out(i)}, P''_i, P''_{in(i)}, P''_{out(i)}$ - none of these sets have common elements;

$T_i \subseteq T$ - a nonempty finite set of transitions;

$\Omega_i \subseteq \Omega$ - a nonempty finite set of statements;

$\Psi_i \subseteq \Psi$ - a nonempty finite set of conditions;

$R_i \subseteq R$ and $R_i \subseteq ((P'_i \cup P''_i \cup P_{out(i)}) \times T_i) \cup (T_i \times (P'_i \cup P''_i \cup P_{in(i)}))$ - the incidence relation of the subnet;

$\Delta_i : P_i \to \Omega_i$ - the function that assigns to each place a statement;

$K_i : (P'_i \cup P'_{in(i)} \cup P'_{out(i)}) \to 1$ and $(P''_i \cup P''_{in(i)} \cup P''_{out(i)}) \to \aleph \setminus \{1\}$ - the function that assigns to each place its capacity;

$\Gamma_i : T_i \to \Psi_i$ - the function that assigns to each transition a condition;

$\Theta_i : T_i \to [0,1]$ - the function that defines the degree to which the condition corresponding to the transitions is satisfied;

$W_i : R_i \to \aleph$ - the weight function that meets the two conditions: $W_i(p,t) \le K_i(p)$ and $W_i(t,p) \le K_i(p)$ for $\forall p \in P_i$, $\forall t \in T_i$;

$M_{0(i)} : (P'_i \cup P'_{in(i)} \cup P'_{out(i)}) \to \{0,1\}$ and $(P''_i \cup P''_{in(i)} \cup P''_{out(i)}) \to W_+$ - the initial marking functions, where $\forall p'' \in (P''_i \cup P''_{in(i)} \cup P''_{out(i)})$ there exists $z \in \aleph \cup \{0\}$ such that $z \le K_i(p'')$ and $M_{0(i)}(p'') = z / K_i(p'')$;

e - the global event that synchronizes the work of all transitions.

By Definition 5, the each set that belongs to the subnet (places, transitions, conditions, statements and arches) is respectively subset of the set (*P, T, Ψ, Ω, R*) which is contained in the definition of the flat net. This definition together with the Definition 4 induct recursion. Accordingly, the each subnet can have internal macroplaces, that are overriding macroplaces for some subnets**.** These subnets can contain another internal macroplaces etc. The relationships between subnets/macroplaces is described in the following definition.

Definition 6. For any two subnets $N_i, N_k \in N$ for $0 \le i \le d, 0 \le k \le d$ and $i \ne k$, the relationship between their sets is defined as follows:

$$P_i \cap P_k = \begin{cases} \{p_{in(i)}, p_{out(i)}\} \text{ for } i \ne 0 \text{ and } mp_i \in MP_k \\ \{p_{in(k)}, p_{out(k)}\} \text{ for } k \ne 0 \text{ and } mp_k \in MP_i \text{ ;} \\ \varnothing \text{ in other cases} \end{cases}$$

$T_i \cap T_k = \varnothing$; $\Omega_i \cap \Omega_k = \varnothing$; $\Psi_i \cap \Psi_k = \varnothing$;

$R_i \cap R_k = \varnothing$; $MP_i \cap MP_k = \varnothing$;

whereas the following conditions are additionally fulfilled for $d > 1$ and $i,k \ne 0$:

$\{p_{in(i)}, p_{out(i)}\} \cap \{p_{in(k)}, p_{out(k)}\} = \varnothing$.

By Definition 6, it is shown that all the sets of subnets are disjoint with the exception of the places set. Taking into consideration two subnets that an overriding macroplace for the first subnet is an internal macroplace for the second, the common part of the places sets of the both subnets is the input and output place of the macroplace. However, the input and output place of the first subnet cannot be the input and output place of the second subnet. This restriction prevents the possibility of the creation of an infinite number of macroplaces for a finite number of places.

4. The creation of the macroplace using the decomposition

Systems based on the HFIPN could be constructed through various techniques. The first approach is called the decomposition technique. It involves a creation of separate subnets by selecting the areas of a "ready" flat net. Then the properties of individual subnets are investigated and if needed, subnets are changed. The second way is the composition, that involves combining previously created subnets (macroplaces) into one hierarchical net. The authors predict that in a practical approach to control systems based on the HFIPN these two methods will be used complementary on different levels of the hierarchy.

Figure 4 - Figure 10 show the example of the HFIPN creation based on the flat net. Figure 4 presents the flat FIPN and the two selected fragments to form macroplaces. At the beginning of the decomposition two subnets N_1

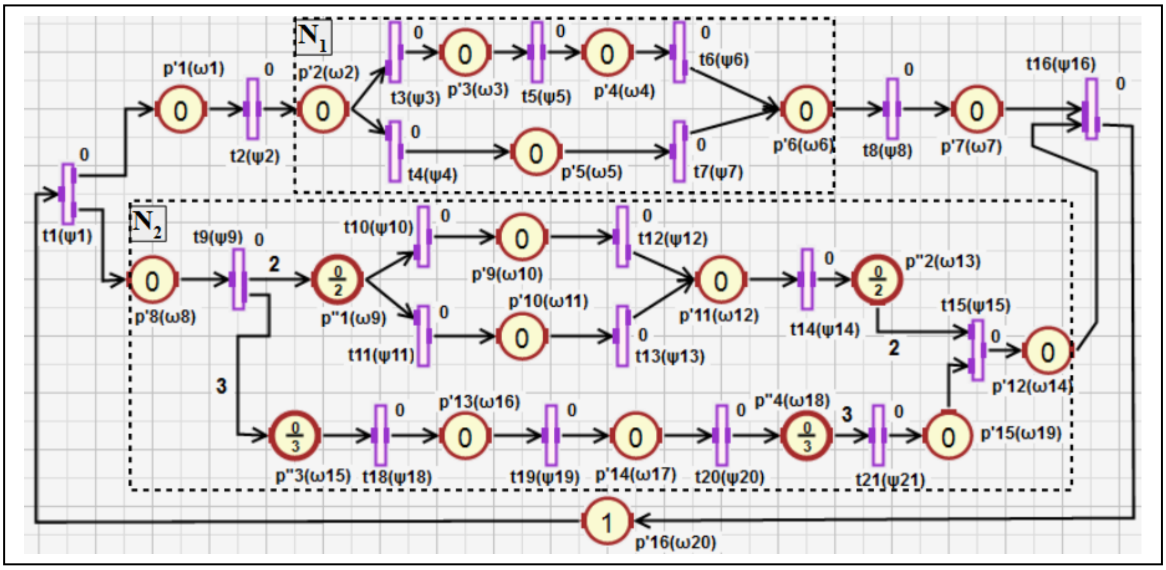

Figure 4. The view of the flat net and the sampled subnets to create its Hierarchical representation

and N_2 have been separated and replaced by their overriding macroplaces mp_1, mp_2. In this way, the subnet N_0 was created (Figure 5) and: the macroplaces mp_1, mp_2 are internal for this net. The subnet N_0 after the decompositions consists of the following sets:

$P'_0 = \{p'_1, p'_7, p'_{16}\}$, $P''_0 = \emptyset$, $P'_{in(0)} = \{p'_2, p'_8\}$, $P''_{in(0)} = \emptyset$, $P'_{out(0)} = \{p'_6, p'_{12}\}$, $P''_{out(0)} = \emptyset$, $T_0 = \{t_1, t_2, t_8, t_{16}\}$, $\Omega_0 = \{\omega_1, \omega_7, \omega_{20}\}$, $\Psi_0 = \{\psi_1, \psi_2, \psi_8, \psi_{16}\}$, $R_0 = \{(t_1,p'_1), (p'_1,t_2), (t_2,p'_2), (p'_6,t_8), (t_8,p'_7), (p'_7,t_{16}), (t_1,p'_8), (p'_{12},t_{16}), (t_{16},p'_{16}), (p'_{16},t_1)\}$, $MP_0 = \{mp_1,mp_2\}$.

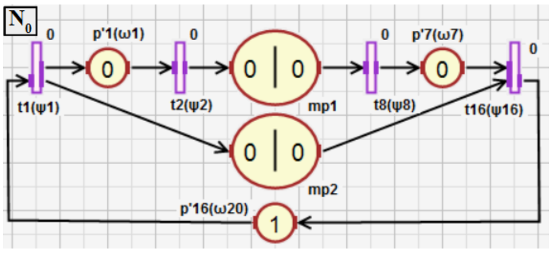

Figure 5. The view of N_0 after the creation of the macronets N_1 and N_2.

Macronet N_1 consists of the following elements (Figure 6): $P'_1 = \{p'_2, p'_3, ..., p'_6\}$, $P''_1 = \emptyset$, $P'_{in(1)} = \emptyset$, $P''_{in(1)} = \emptyset$, $P'_{out(1)} = \emptyset$, $P''_{out(1)} = \emptyset$, $T_1 = \{t_3, t_4, ..., t_7\}$, $\Omega_1 = \{\omega_2, \omega_3, ..., \omega_6\}$, $\Psi_1 = \{\psi_3, \psi_4, ...,\psi_7\}$, $R_1 = \{(p'_2,t_3), (t_3,p'_3), (p'_3,t_5), (t_5,p'_4), (p'_4, t_6), (t_6, p'_6), (p'_2,t_4), (t_4, p'_5), (p'_5,t_7), (t_7,p'_6)\}$, $MP_1 = \emptyset$. In the subnet N_2 (Figure 7) the ubnets N_3 and N_4 were separated sequentially.

Finally N_2 consists of the following sets (Figure 8): $P'_2 = \{p'_8, p'_{12}, p'_{15}\}$, $P''_2 = \{p''_2\}$, $P'_{in(2)} = \emptyset$, $P''_{in(2)} = \{p''_1, p''_3\}$, $P'_{out(2)}= \{p'_{11}\}$, $P''_{out(2)}= \{p''_4\}$, $T_2 = \{t_9, t_{14}, t_{15}, t_{21}\}$, $\Omega_2 = \{\omega_8, \omega_{13}, \omega_{14}, \omega_{19}\}$, $\Psi_2 = \{\psi_9, \psi_{14}, \psi_{15}, \psi_{21}\}$, $R_2= \{(p'_8,t_9), (t_9,p''_1), (p'_{11},t_{14}), (t_{14},p''_2), (p''_2,t_{15}), (t_9,p''_3), (p''_4,t_{21}), (t_{21},p'_{15}), (p'_{15},t_{15}), (t_{15},p'_{12})\}$, $MP_2 = \{mp_3, mp_4\}$.

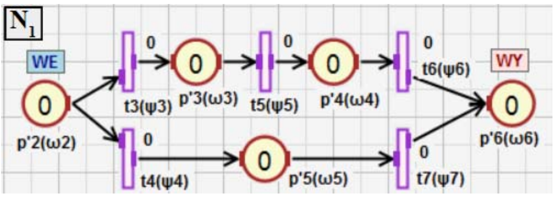

Figure 6. The view of N_1 after the decomposition

In this way, there was created macronet N_3 (Figure 9), that consists of the sets: $P'_3 = \{p'_9p'_{10}, p'_{11}\}$, $P''_3 = \{p''_1\}$, $P'_{in(3)} = \emptyset$, $P''_{in(3)} = \emptyset$, $P'_{out(3)} = \emptyset$, $P''_{out(3)} = \emptyset$, $T_3 = \{t_{10}, t_{11}, t_{12}, t_{13}\}$, $\Omega_3 = \{\omega_9, \omega_{10}, \omega_{11}, \omega_{12}\}$, $\Psi_3 = \{\psi_{10}, \psi_{11}, \psi_{12}, \psi_{13}\}$, $R_3 = \{(p''_1,t_{10}), (t_{10},p'_9), (p'_9,t_{12}), (t_{12},p'_{11}), (p''_1,t_{11}), (t_{11},p'_{10}), (p'_{10},t_{13}), (t_{13},p'_{11})\}$, $MP_3 = \emptyset$ and macronet N_4 (Figure 10) described by sets: $P'_4 = \{p'_{13}, p'_{14} \}$, $P''_3 = \{p''_3, p''_4\}$, $P'_{in(4)} = \emptyset$, $P''_{in(4)} = \emptyset$, $P'_{out(4)}= \emptyset$, $P''_{out(4)} = \emptyset$, $T_4 = \{t_{18}, t_{19}, t_{20}\}$, $\Omega_4 = \{\omega_{15}, \omega_{16}, \omega_{17}, \omega_{18}\}$, $\Psi_4 = \{\psi_{18}, \psi_{19}, \psi_{20}\}$, $R_4 = \{(p''_3,t_{18}), (t_{18},p'_{13}), (p'_{13},t_{19}), (t_{19},p'_{14}), (p'_{14},t_{20}), (t_{20},p''_4)\}$, $MP_4 = \emptyset$.

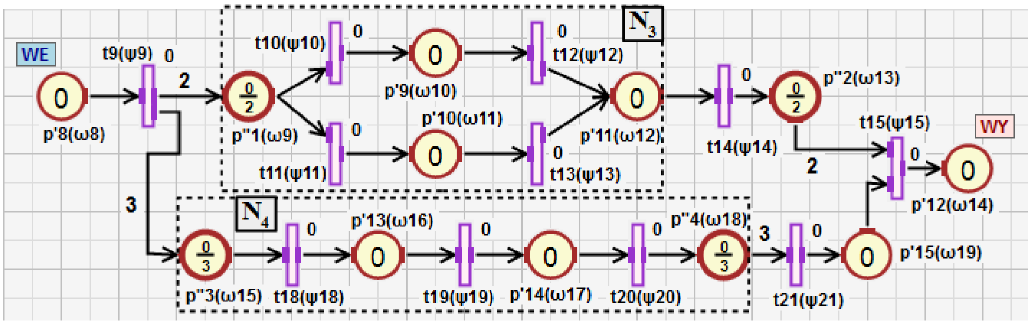

Figure 7. The view of N_2 after first decomposition before next

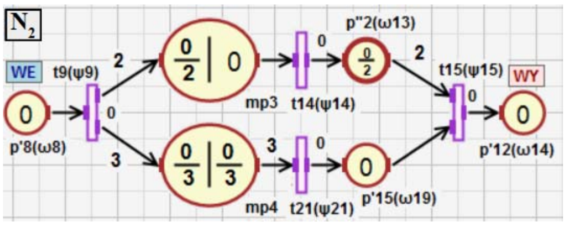

Figure 8. The view of N_2 after creating the macroplaces mp_3 and mp_4.

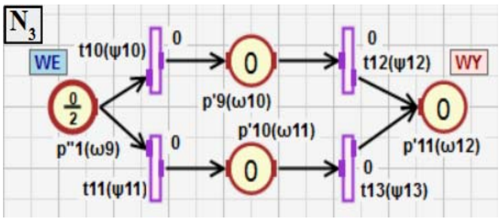

Figure 9. The view of N_3.

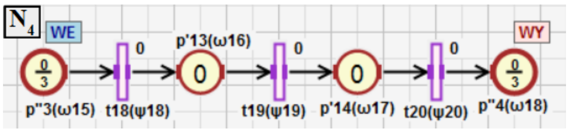

Figure 10. The view of N_4.

In order to avoid conflicts between the transitions t_{12} and t_{13} in the subnet N_3 (Figure 9), the disjoint conditions should be attributed to them.

5. The algebraic representation of the HFIPN

The HFIPN may be modeled by algebraic representation described in [8]. For each subnet $N_i \in N$ the formula becomes:

$$M'_i = M_i + \frac{(U_i \wedge \Delta\Theta_i)\cdot C_i}{K_i} \qquad (4)$$

where:

$\boldsymbol{M_i}$ - the vector of the length 1 x a_i (a_i - number of places) holds the current marking of places;

$\boldsymbol{M'_i}$ - the vector holds the next marking with the same length;

$\boldsymbol{U_i}$ - the vector of the length 1 x b_i (b_i - number of transitions), in which the given coefficient is equal one, if it corresponds to the enabled transition by the marking $\boldsymbol{M_i}$ in the subnet;

$\boldsymbol{\Delta\Theta_i}$ - the vector of the length 1 x b_i, in which the coefficient $\Delta\vartheta_j, 1 \le j \le b_i$ describes the increment in a degree to which the condition corresponding to the transition t_j is satisfied;

$\boldsymbol{K_i}$ – the vector of the length 1 x a_i which assigns to each place its capacity;

$\boldsymbol{C_i}$ – the incidence matrix of dimension b_i x a_i, analogous to the classical Petri nets.

According to (4) each subnet N_i in the HFIPN has its own algebraic representation. The algebraic representation of the subnet N_0 is for modeling of the entire net. For this subnet the incidence matrix $\boldsymbol{C_0}$ and the vectors $\boldsymbol{M_0}$, $\boldsymbol{M'_0}$, $\boldsymbol{U_0}$, $\boldsymbol{\Delta\Theta_0}$, $\boldsymbol{K_0}$ have the same form as for the corresponding flat FIPN. All elements of subnet N_0 and subnets associated with it at a lower level must be used in order to create mentioned matrix and vectors. For the HFIPN generated by the decomposition shown in Figure 4 - Figure 10 the algebraic representation is the sum obtained from subnets: $N_0 \cup N_1 \cup N_2 \cup N_3 \cup N_4$.

For other subnets the algebraic representation is used to study the static and dynamic properties. For the subnet N_2 in Figure 8 these elements are obtained from the sum of the subnets: $N_2 \cup N_3 \cup N_4$. The incidence matrix $\boldsymbol{C_2}$ for this macronet looks as in Figure 11.

The incidence matrix C_2 can be used to study the properties of subnets N_2 e.g. to find P-invariants.

$$
\begin{array}{r|cccccccccccc|}
 & p'_8 & p'_9 & p'_{10} & p'_{11} & p'_{12} & p'_{13} & p'_{14} & p'_{15} & p''_1 & p''_2 & p''_3 & p''_4 \\
t_9 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 3 & 0 \\
t_{10} & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
t_{11} & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
t_{12} & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
t_{13} & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\boldsymbol{C}_2 = t_{14} & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
t_{15} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & -2 & 0 & 0 \\
t_{18} & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\
t_{19} & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\
t_{20} & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 \\
t_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -3
\end{array}
$$

Figure 11. The view of the matrix C_2 for the subnet N_2.

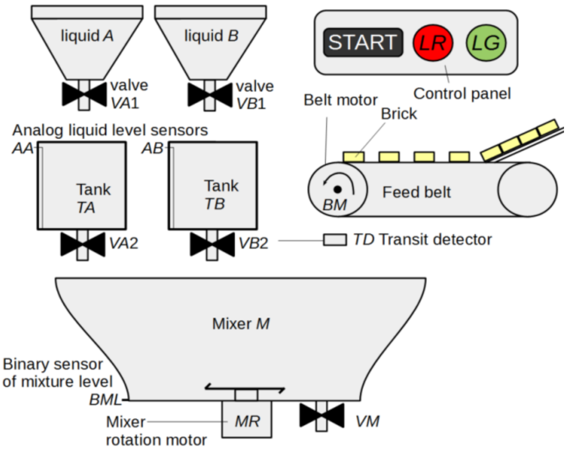

Figure 12. View of the system to mix three components: two liquids and soluble bricks

.

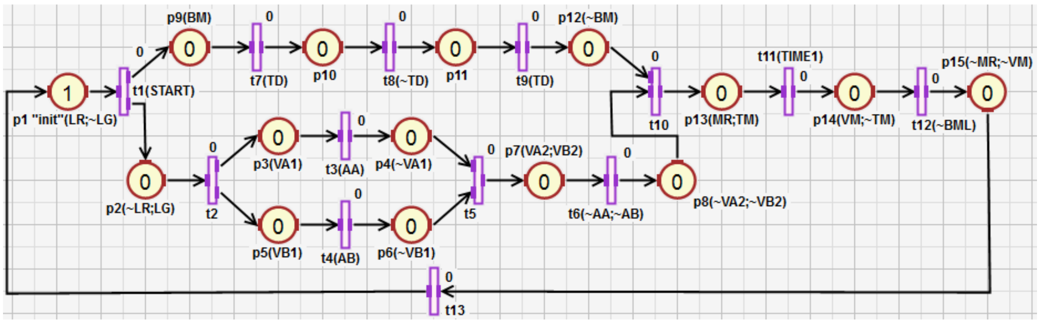

Figure 13. View of the control system to mix three components with the application of flat FIPN.

6. Exemplification of the approach

In this section, a practical example is described based on [18], and it is modified to highlight the advantages of HFIPN in comparison to discrete PNs. In Figure 12, a system to mix three components is presented. In Figure 13 and Figure 14, a FIPN graphs to control the system from Figure 12 are shown. The first has a flat structure (Figure 13), while the second is a hierarchical net (Figure 14). The hierarchical net can be created based on Figure 13 using the decomposition technique as described in section 4, or each of its subnets can be constructed separately and combined by the application of the composition technique.

In the presented example, the depth level of the supported hierarchical structure is 1. However, the simulator FIPN-SML does not limit this level through the use of recursion. This tool enables the creation of macroplaces in two ways: by selecting a part of a subnet and by creating an "empty" macroplace that has only two places (input and output). When creating a macroplace by the selection, the selected part of a subnet may only have one input and one output place connected to the unselected part.

The proposed hierarchical control system works in the following manner. First, when the button *START* on a control panel is pressed, a red light *LR* is off, and a green light *LG* is on, which means the process is running. Then, liquids *A* and *B* are transferred into tanks *TA* and *TB*. Dosing of the liquid *A* (*B*) using a valve *VA*1(*VB*1) is monitored by an analog sensor *AA* (*AB*), and it is finished when the tank is full (the sensor value is equal to 1). Next, the both liquids stored in the tanks *TA* and *TB* are transferred to the mixer *M* through the valves *VA*2 and *VB*2. The valve *VA*2 (*VB*2) is closed when the value of liquid level sensor *AA* (*AB*) is equal to 0. In the meantime, two soluble bricks are fed into the mixer *M*, which is monitored by a transit detector *TD*. The bricks are brought using a belt as shown in Figure 12. After that, when all of the components are in the mixer *M,* they are mixed for a period of time measured by a timer *TIME1*. The timer *TIME1* is initiated by a variable *TM*. In the mixing

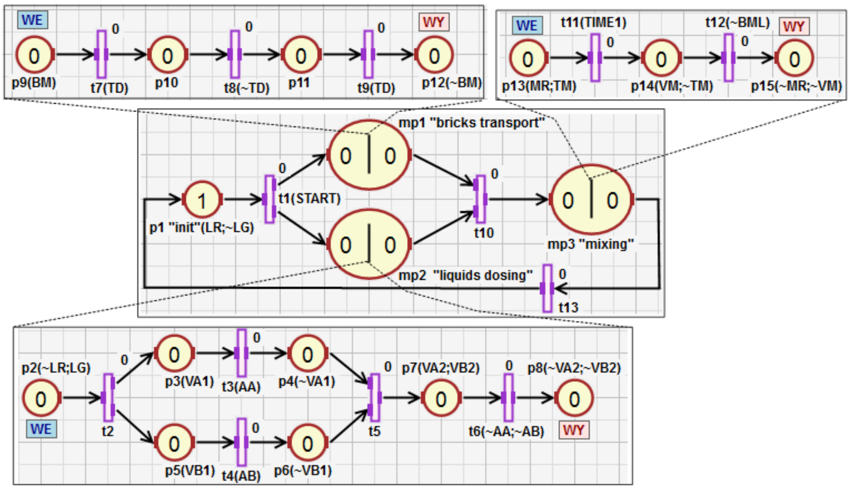

Figure 14. View of the control system to mix three components using a HFIPN graph

process, a rotation motor *MR* is used. Finally, the resulted liquid is emptied from the mixer *M* through a valve *VM*, and the whole process is repeated. The mixture level in the mixer *M* is monitored by a binary liquid level sensor *BML*.

The advantage of (H)FIPN in comparison to Grafcet and other discrete nets is the possibility of using analogue sensors. In the proposed system (Figure 14), the analogue sensors *AA* and *AB* allow observing the progress of filling and emptying of the tanks *TA* and *TB*. Moreover, these sensors offer the possibility of designing monitor system. Such system wouldallow to check the uniformity of filling the mixer *M* by the liquids *A*, *B* and to execute additional control actions. If one of the liquids is transferred too fast or too slow compared to the other, the mixing time can be increased or the all process can be stopped.

7. Conclusions

The article presents a conception, formal definition and the algebraic representation of the HFIPN. The HFIPN model based on the FIPN model, allows in a better way to visualize the progress of tokens' flow between the places (in comparison to binary PNs). The proposed solution simplifies the design and validation of complex net by the use of a hierarchical structure, because it allows to divide and analyze its smaller fragments. The HFIPN model has been implemented in the simulator FIPN-SML, which permits to build a net with the use of the decomposition, the composition, and their combinations. The FIPN-SML could significantly reduce the time of the creation of control systems. The proposed model facilitates the creation of hierarchical nets based on a macroplace with one input and output place.

REFERENCES

1. van der Aalst W. (2000). Workflow verification: Finding control-flow errors using petri-net-based techniques. In *Business Process Management* (pp. 161-183). Springer Berlin Heidelberg.
2. Amer-Yahia C. & Zerhouni N. (2001). State Equation and Stability for a Class of Continuous Petri nets Application to the Control of a Production System. *Studies in Informatics and Control*, *10*(4), 301-318.
3. Andrzejewski G. (2005). Hierarchical Petri nets for digital controller design. In *Design of embedded control systems* (pp. 27-36). Springer US.
4. Berthelot G. (1985, June). Checking properties of nets using transformations. In *European Workshop on Applications and Theory in Petri Nets* (pp. 19-40). Springer Berlin Heidelberg.
5. David R. & Alla H. (2010). *Discrete, continuous, and hybrid Petri nets*. Springer Science & Business Media.
6. Desel J. & Esparza J. (2005). *Free choice Petri nets*. Cambridge university press.
7. Girault C. & Valk R. (2003). *Petri Nets for Systems Engineering: A Guide to Modeling, Verification, and Applications* (pp. 18–40, 108–117). Springer Science & Business Media.

8. Gniewek L. (2013). Sequential control algorithm in the form of fuzzy interpreted Petri net. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *43*(2), 451-459.

9. Gniewek L. (2014). Coverability graph of fuzzy interpreted Petri net. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *44*(9), 1272-1277.

10. Hernandez-Martinez E. G., Puga-Velazquez E. S., Foyo-Valdes S. A. & Meda-Campana J. A. (2013). Modeling Framework for Automated Manufacturing Systems Based on Petri Nets and ISA Standards. *Studies in Informatics and Control*, *22*(2), 163-174.

11. Holvoet T. & Verbaeten P. (1995, October). Petri charts: an alternative technique for hierarchical net construction. In *Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference on* (Vol. 3, pp. 2688-2693). IEEE.

12. Hong J. E. & Bae D. H. (1998). HOONets: Hierarchical object-oriented Petri nets for system modeling and analysis, *KAIST Technical Report CS/TR*, 98-132.

13. Jensen K. (2013). *Coloured Petri nets: basic concepts, analysis methods and practical use* (Vol. 1). Springer Science & Business Media.

14. Lakos C. (1995, June). From coloured Petri nets to object Petri nets. In *International Conference on Application and Theory of Petri Nets* (pp. 278-297). Springer-Berlin Heidelberg.

15. Lee K. H. & Favrel J. (1985). Hierarchical reduction method for analysis and decomposition of Petri nets, *IEEE Transactions on Systems, Man, and Cybernetics*, (2), 272-280.

16. Markiewicz M., Surdej Ł. & Gniewek L. (2016, August). Transformation of a fuzzy interpreted Petri net diagram into structured text code. In *Methods and Models in Automation and Robotics (MMAR), 2016 21st International Conference on* (pp.94-99). IEEE.

17. Suzuki I. & Murata T. (1983). A method for stepwise refinement and abstraction of Petri nets, *Journal of computer and system sciences*, *27*(1), 51-76.

18. The International Electrotechnical Commission (2013). GRAFCET specification language for sequential function charts, *International standard IEC 60848:2013*.

19. Valette R. (1979). Analysis of Petri nets by stepwise refinements, *Journal of computer and system sciences*, *18*(1), 35-46.

20. Valk R. (1998, June). Petri nets as token objects. In *International Conference on Application and Theory of Petri Nets* (pp. 1-24). Springer Berlin Heidelberg.

21. Zhou M. (1998). Modeling, analysis, simulation, scheduling, and control of semiconductor manufacturing systems: A Petri net approach, *IEEE Transactions on Semiconductor Manufacturing*, *11*(3), 333-357.

22. Zhou M., DiCesare F. & Desrochers A. A. (1992). A hybrid methodology for synthesis of Petri net models for manufacturing systems, *IEEE transactions on robotics and automation*, *8*(3), 350-361.

23. Zhou M. & Venkatesh K. (1999). *Modeling, simulation, and control of flexible manufacturing systems: a Petri net approach*. World Scientific.