



HIERARCHICAL DECOMPOSITION OF PETRI NETS FOR ANALYSIS AND DESIGN OF DIGITAL MICROSYSTEMS

Andrei Karatkevich, Grzegorz Andrzejewski

University of Zielona Góra, ul. Podgórna 50, 65-246 Zielona Góra, POLAND.
A.Karatkevich@iie.uz.zgora.pl, G.Andrzejewski@iie.uz.zgora.pl

Abstract: *The paper is dedicated to hierarchical Petri nets with application to design of control devices. It consists of two parts. The first part introduces a concept of simplifying analysis of flat Petri nets (we discuss analysis of reachability and liveness) by means of hierarchical decomposition. We present a class of subnets which can be replaced by the macroplaces. Further we describe properties of such decomposition and show, that analysis of the mentioned properties of a net can be reduced to the corresponding analysis of the elements of decomposition. A method allowing to obtain the decomposition is presented. The second part presents an approach to specification of a digital control system behavior by means of hierarchical Petri nets. We describe a model, called HPN, which has some of important properties, simplifying description of strongly reactive systems.*

Keywords: *Petri nets, hierarchy, decomposition, analysis, logical controllers.*

1. INTRODUCTION

Petri nets [1] present a mathematical model describing parallel asynchronous discrete systems. Petri nets are used for modeling, simulation, analysis and synthesis of a wide range of systems, such as parallel algorithms, asynchronous digital circuits, networks, multitasking operating systems, communication protocols, distributed software, industrial control systems (especially real-time systems) and so on. Several languages for description of logical control algorithms, such as Grafset, an industrial standard SFC and various forms of interpreted nets, are based on Petri nets or similar models (see, for example, [2,3]).

Analysis of Petri nets is important in most of their applications. Often it is a time- and memory-consuming task. Size of reachability space of a bounded Petri net depends exponentially on the net size (the unbounded nets have infinite reachability space), and a brute force approach does not allow practical analysis of the big nets. So many advanced methods of Petri net analysis have been developed. Liveness and reachability analysis belong to the main tasks of analysis of Petri nets.

A perspective approach to simplifying analysis of Petri nets and, generally, to handling big discrete systems, is based on the idea of decomposition. The task of net analysis is reduced to the task of analysis of the blocks of its decomposition. Also for the synthesis purposes decomposition is useful. More

material about theory and applications of net decomposition can be found in [4-6].

A Petri net model in its original form has no hierarchical structure. But hierarchical structures are required in many applications of Petri nets in system engineering; hierarchical description of a system simplifies greatly understanding of its functioning, its analysis, verification and implementation. So, numerous hierarchical Petri net models and the ways of hierarchical decomposition have been proposed (see [2,7,8]).

A hierarchical structure is easy to process if all the elements at all the levels are of the same kind and need no special ways to describe and deal with. Below we describe a method that allows transforming a flat safe Petri net into a hierarchical one. The proposed method, being a combination of hierarchical and block decomposition, is simple and does not require exhausting of multiple variants; on the other hand, it allows selecting the subnets of complex structure. In the last part of the article we present a model of hierarchical Petri net which allows simple description of strongly reactive systems. The model supports many of important elements of behavioral description of control systems: behavioral hierarchy, similar to statecharts [17], prohibiting and enabling arcs which allow easy modeling of synchronization points, time dependencies which extend range of using the model in areas of real control.

2. DEFINITIONS (PETRI NETS)

An ordinary *Petri net* [1,9] is a triple $\Sigma = (P, T, F)$, where P is a set of *places*, T – a set of *transitions*; $P \cap T = \emptyset$; $F \subseteq (P \times T) \cup (T \times P)$. For $t \in T$ *t denotes $\{p \in P | (p, t) \in F\}$; t^* denotes $\{p \in P | (t, p) \in F\}$; *t and t^* are the *sets of input and output places*, respectively.

A *marking* of a net is defined as a function $M: P \rightarrow \{0, 1, 2, \dots\}$. It can be considered as a number of *tokens* situated in the net places. Number of tokens in a place p for marking M is denoted as $M(p)$. Initial marking is denoted as M_0 .

A transition t is *enabled* and can *fire* if $\forall p \in {}^*t: M(p) > 0$. Transition firing removes one token from each input place and adds one token to each output place of it.

A marking M' that can be reached from M by a sequence of transition firings (a *firing sequence*) $\sigma = t_1 t_2 \dots t_n$ is called *reachable* from M ; we write $M \sigma M'$. The set of all markings reachable from M is denoted as $[M]$.

A transition is *live*, if there is a reachable marking at which it is enabled. A net is *live*, if for every reachable marking every transition of the net is live. A net is *quasi-live*, if for the initial marking every transition of the net is live. A net is *safe*, if $\forall p \in P \forall M \in [M_0] M(p) \leq 1$.

Graphically Petri nets are presented as oriented graphs with nodes corresponding to places and transitions and the arcs joining places to transitions or transitions to places. Tokens are shown as dots inside the places.

Definitions of *subnets* and the related notions see in [1]. Definition of *block* see in [3].

3. A CONCEPT OF HIERARCHICAL DECOMPOSITION OF PETRI NETS

Consider the properties of a subnet that can be replaced with a place (see for example Fig. 1).

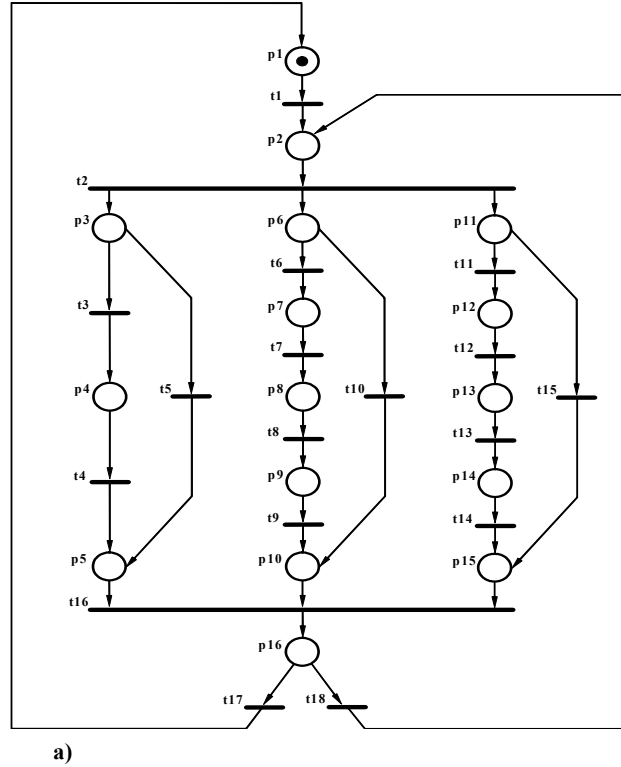
Let $\Sigma' = (P', T', F')$ be a subnet of $\Sigma = (P, T, F)$. First of all, Σ' must have an interface – a set of places $P'_{i/o}$ incident to the external transitions. $P'_{i/o}$ has two subsets - P'_{in} and P'_{out} , being output and input for the external transitions, correspondingly (those sets may intersect). The next condition must be satisfied:

$$\forall t \in (T \setminus T') : \left(({}^*t \cap P'_{in} \neq \emptyset \Rightarrow (t^* \supseteq P'_{in})) \wedge \left(({}^*t \cap P'_{out} \neq \emptyset \Rightarrow ({}^*t \supseteq P'_{out})) \right) \right) \quad (1)$$

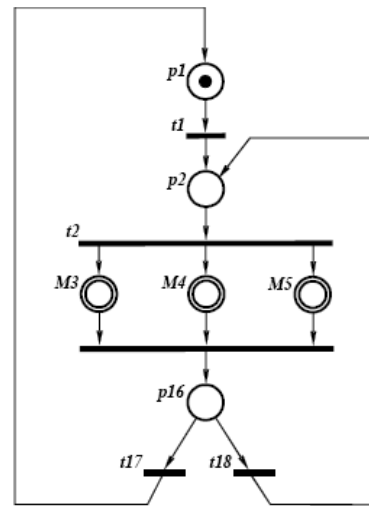
That means that a subnet can get the tokens from the outside only into all of its input places together,

and it can loose tokens from all the output places also only together. The initial marking also should be taken into account, so the next condition is needed:

$$\left((\exists p \in P' : M_0(p) > 0 \Rightarrow (p \in P'_{in} \wedge (\forall q \in P'_{in} : M_0(q) = M_0(q))) \vee (p \in P'_{out} \wedge (\forall q \in P'_{out} : M_0(q) = M_0(q))) \right) \quad (2)$$



a)



b)

Fig.1 - A Petri net and the top level of corresponding hierarchical Petri net (one of possible variants)

Denote by M'_{in} the marking such that $M'_{in}(p) = 1$ if and only if $p \in P'_{in}$, and by M'_{out} the marking such that $M'_{out}(p) = 1$ if and only if $p \in P'_{out}$. It is reasonable to consider M'_{in} as an initial marking for

the subnet ($M'_{in}=M'_0$). So one more condition is important:

$$\forall M((M \in [M'_{in}]) \Rightarrow (M'_{out} \in [M])) \quad (3)$$

And finally, if all the places in P'_{out} obtain tokens, no tokens should remain "inside" the subnet. This condition is described by the next formula:

$$\forall M((M \in [M'_{in}]) \Rightarrow \overline{(M > M'_{out})}) \quad (4)$$

Definition. A P-block is a subnet Σ' of Petri net Σ , which satisfies conditions (1-4).

So, if a subnet Σ' of Petri net Σ is a P-block, we can construct the net $\Sigma_H = (P_H, T_H, F_H)$ (meaning by H the higher level) in the next way (here $q \in P_H$ is the macroplace):

$$\begin{aligned} P_H &= (P \setminus P') \cup \{q\}; \\ T_H &= (T \setminus T'); \\ F_H &= \{(x, y) \mid (((x \in P_H \wedge y \in T_H) \\ &\vee (x \in T_H \wedge y \in P_H)) \wedge (x, y) \in F) \\ &\vee ((x = q) \wedge (y \in T_H)) \\ &\wedge \exists p \in P': (p, y) \in F) \vee (y = q) \\ &\wedge (x \in T_H) \wedge (\exists p \in P': (x, p) \in F)\} \end{aligned} \quad (5)$$

Correspondence between markings of Σ_H and Σ , when Σ is safe, is described by the next formula:

$$M_H(p) = \begin{cases} M(p), & p \in P \\ \max(M(s \in P')), & p = q \end{cases} \quad (6)$$

The next lemmas demonstrate that the tasks of reachability and liveness analysis of a Petri net can be reduced to the same tasks for the elements of the composition. Proofs are omitted because of limited space.

Lemma 1. Let Σ be a safe Petri net, Σ' its P-block, Σ_H is constructed according to (5), M' is a projection on Σ' of marking M of Σ . Then for Σ $M_1 \in [M_2]$, if and only if:

1. for Σ_H : $M_{1H} \in [M_{2H}]$, $M_{1H}(q) = M_{2H}(q) = 0$, or
2. for Σ_H : $M_{1H} \in [M_{2H}]$, $M_{1H}(q) = 0$, $M_{2H}(q) = 1$; for Σ' : $M'_{out} \in [M'_2]$; or
3. for Σ_H : $M_{1H} \in [M_{2H}]$, $M_{1H}(q) = 1$, $M_{2H}(q) = 0$; for Σ' : $M'_1 \in [M'_{in}]$; or
4. for Σ_H : $M_{1H} \in [M_{2H}]$, $M_{1H}(q) = M_{2H}(q) = 1$, for Σ' : $M'_1 \in [M'_2]$, if the firing sequence in Σ_H leading from M_{2H} to M_{1H} does not

contain transition t such that $q \in \bullet t$, or $(M'_{out} \in [M'_2], M'_1 \in [M'_{in}])$, otherwise;

- or
5. $M_{1H} = M_{2H}$, $M'_1 \in [M'_2]$.

Lemma 2. Let Σ be a safe Petri net, Σ' its P-block. Σ is live, if and only if Σ_H is live and Σ' is quasi-live.

How much P-decomposition simplifies the analysis? It is evaluated by the next lemma, describing dependency between cardinality of reachability sets of decomposed net and the elements of decomposition.

Lemma 3 Let Σ be a safe Petri net, Σ' its P-block., Σ_H is constructed according to (5). Then

$$\begin{aligned} |[M]| = & |\{M_H \in [M_{0H}] \mid M_H(q) = 0\}| + \\ & + |\{M_H \in [M_{0H}] \mid M_H(q) = 1\}| \cdot |[M'_{in}]| \end{aligned} \quad (7)$$

where $[M_0]$ concerns Σ , $[M_{0H}]$ - Σ_H , $[M'_{in}]$ - Σ' , correspondingly.

Of course, a net can be decomposed into many subnets in hierarchical way; the affirmations analogous to the lemmas presented above but describing multi-level decomposition can be easily proved by using induction. Let us call a net decomposition *P-decomposition*, if all the subnets the net is decomposed into are the P-blocks (except of the net at the highest level).

4. PROPERTIES OF P-DECOMPOSITION

Lemma 4. If the nets Σ' and Σ'' are the P-blocks of net Σ , and Σ'' is a subnet of Σ , then Σ'' is a P-block of Σ' .

This ensures the possibility of hierarchical decomposition.

Lemma 5. In the nets Σ' and Σ'' are the P-blocks of net Σ , and neither of those blocks is a subnet of the other, but $T' \cap T'' \neq \emptyset$, then $\Sigma_1 = (\Sigma' \cap \Sigma'')$, $\Sigma_2 = (\Sigma' \setminus \Sigma'')$, $\Sigma_3 = (\Sigma'' \setminus \Sigma')$ and $\Sigma_4 = (\Sigma' \cup \Sigma'')$ are the P-blocks of Σ (see Fig. 2; it can be shown, that only those two variants of intersection of P-blocks are possible).

It does not mean that for a Petri net one canonical P-decomposition with maximal number of blocks exists, because the P-blocks may intersect by places, having no common transitions. It is easy to see, that all the blocks Σ_1 , Σ_2 and Σ_3 in Fig. 2b cannot be replaced all together by the macroplaces in a higher-level net.

We have no polynomial algorithm allowing finding all the minimal P-blocks in a net; but as it was shown above (Lemma 3), even finding some of P-blocks can simplify the net analysis.

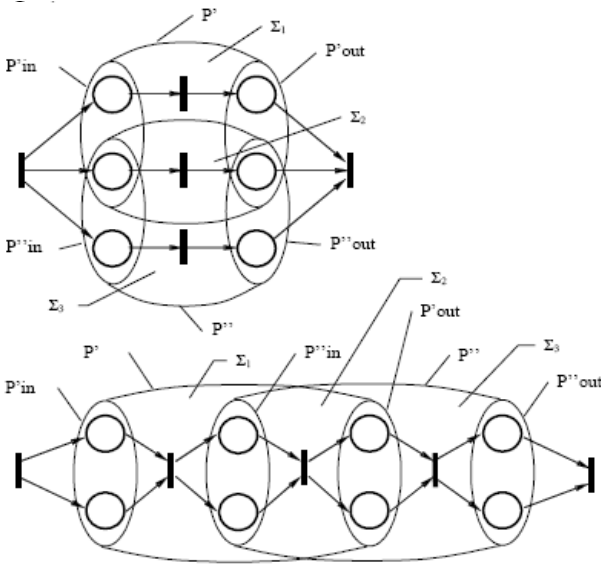


Fig.2 - Variants of intersection of P-blocks

5. FINDING P-BLOCKS

To check whether a subnet is a P-block, it is enough to check conditions (1-4). Checking of conditions (3,4) in general case requires constructing of the reachability graph; but if it is known that the net is live and safe, then such check can be performed easier.

Lemma 6. If the net Σ is live and safe and a subnet Σ' of it satisfies (1, 2), then Σ' is a P-block.

But even in a case when it is easy to decide whether a subnet is a P-block, decomposition is not a simple task because the number of subnets exponentially depends on the net size.

It is easy to see, that any P-block Σ' such that $P'_{in} \cap P'_{out} = \emptyset$ is a *block* in the sense of [3], but not any block is a P-block. Affirmation 5.14 from [3] provides a simple way to find for a given Petri net its partition into the minimal blocks. This partition can be a base for finding a P-decomposition (without a guarantee that the P-blocks are minimal). Some blocks have to be united to obtain P-blocks.

We propose the next algorithm for finding P-blocks for a given safe Petri net Σ .

Algorithm 1

1. Obtain the partition of Σ into minimal blocks (according to Affirmation 5.14 in [3]). Do not consider below the blocks for which their internal places are marked at M_0 .
2. For each block Σ' such that

$$\forall t \in T((t^* \cap P'_{in} \neq \emptyset) \Rightarrow (t^* \supseteq P'_{in})) \quad (8)$$

attach the blocks Σ'' such that $P''_{in} \subseteq P'_{out}$, expanding Σ' while it is possible and Σ' does not satisfy (1,2). If Σ' satisfies (1,2), then it is a P-block.

3. For the blocks that are not included in P-blocks at the previous step consider combinations of their parallel composition satisfying (8) and process them as in item 2.

The algorithm allows to find some (not all) of P-blocks of a net. After applying it to a flat net and composing the P-blocks into the macronets, the same algorithm can be applied to the highest-level net Σ_H , and so on, while the highest-level net is decomposable. In such a way a multi-level hierarchical net can be constructed in bottom-up order. Some notes should be taken into account here: first, if some of the P-blocks detected by Algorithm 1 intersect by the places, but have no common transitions, then, unlike for the *blocks*, partial composition is impossible for them; they may be only in a *complete composition* [3]. In such cases cardinality of the reachability set of those P-blocks equals the sum of cardinalities of reachability sets of the P-blocks, and for analysis purposes those P-blocks should be united. In particular, there is no sense to build a macroplace from an SM-component [1], if it can be built from a "bigger" SM-component (covering the first one). For example, a subnet with transitions t_7 and t_8 in Fig. 1a satisfies definition of P-block, but there is no sense to select it, because the subnet consisting of transitions t_6-t_{10} (and the incident places) can be selected (as it is done in Fig. 1b).

6. EXAMPLE

Consider the net shown in Fig.1a. The subnets with the next sets of transitions are the P-blocks: $\{t_3-t_{15}\}$, $\{t_3-t_5\}$, $\{t_6-t_{10}\}$, $\{t_{11}-t_{15}\}$, $\{t_7, t_8\}$, $\{t_{12}, t_{13}\}$. According to the notes above, there is sense to convert to macroplaces only the subnets corresponding to the sets $\{t_3-t_5\}$, $\{t_6-t_{10}\}$ and $\{t_{11}-t_{15}\}$. So we obtain the hierarchical net, high level of which is shown in Fig. 1b.

Suppose that we want to know, whether the marking is reachable with places p_4 , p_9 and p_{15} marked. At the highest level we have to check whether the marking is reachable at which macroplaces M_3 , M_4 and M_5 are marked. To check it, it is enough to make two steps of simulation of the higher-level net. Then at the lower level we have to check whether in the corresponding subnets the markings are reachable in which only places p_4 , p_5 , and p_{15} are marked, correspondingly. Such check is simple, because all those blocks are sequential. So we have answered the question without constructing of complete reachability graph of the net.

7. HIERARCHICAL PETRI NETS

There are many kinds of hierarchical nets described in the literature: HCPN (*Hierarchical Coloured PN*) [7], OOPN (*Object Oriented PN*)

[11], HOONets (Hierarchical Object-Oriented PN) [12], THORN (Timed Hierarchical Object-Related Nets) [13], MacroNets [14], PetriCharts [15], GrafChart (hierarchical extension of Grafset [16]). Most of them are only formulated as mathematical models. Their fitness for the purpose is limited as they lack some effective elements of description which are required in engineering practice, e.g. time dependencies, the history of macrostates, preemption. An alternative model of hierarchical Petri was proposed in [2]. This model is called HPN (*Hierarchical Petri Net*) and it integrates most important elements of a formal description which are available in other kinds of nets, e.g. interpretation, time dependencies, enabling and inhibiting arcs, history and preemption mechanism. It has allowed the elaboration of an efficient model of specification of control systems with high flexibility of description methods and a clear graphic interface comparable to Statecharts (Fig. 4).

There is a simple example showing a simplified control system of initial washing in an automatic washer (Fig. 3).

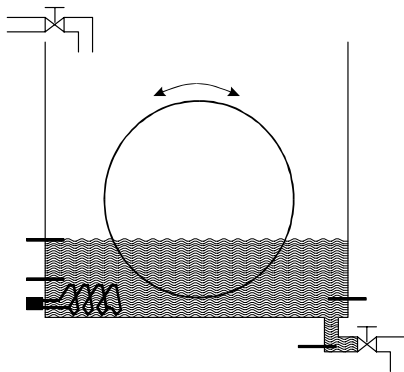


Fig.3 - An example of automatic washer control system

After turning on the washing program valve *VI* is opened and water is infused. The infusing process lasts to the moment of achieving *L1* level. At the same time after exceeding *L2* level (total sinking of heater *H*) if the temperature is below the required (*TL1*), the heater system is turned on. After valve *VI* closing the washing process is started, in which the washing cylinder is turned alternate left and right for 10 sec. with a 5 sec. break between. The process of keeping the temperature constant is active for the whole washing cycle. The cycle is turned off after 280 sec. and cylinder is stopped, the heater is turned-off and valve *V2* is opened for water removal.

Interpreted hierarchical Petri net is presented graphically as an oriented graph, similar to the ordinary Petri nets. Places distinguished by 'x' determine condition of control left out from subnet. Macroplace is a place distinguished by double circle.

With each macroplace a subnet is associated, which activity is dependent on token possession by this macro. An expansion of macroplace (subnet) is defined inside rounded rectangle.

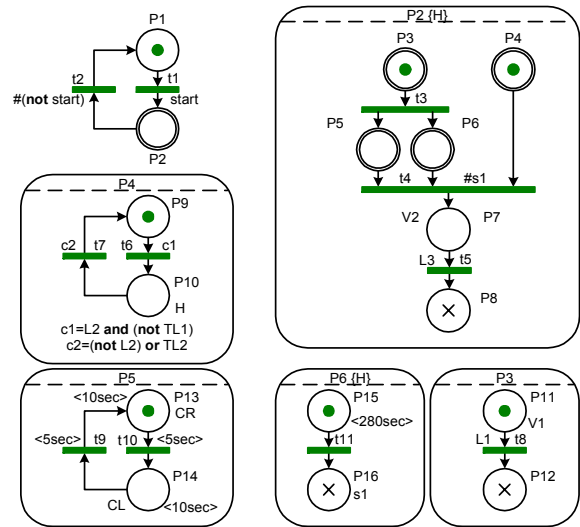


Fig.4 - An example of HPN (automatic washer controller form [2])

Three sets of signals are assigned with the net: *I*, *O* and *L*, meaning sets of inputs, outputs and internal signals, respectively. The sets *I* and *O* are accessible on every level of hierarchy. Signals from set *L* can be of local or global range depending on point of their declarations.

With the nodes the labels are associated. They are created on set *S*. Label assigned to place *p* is compound from names of signals separated by comma and it means subset active signals when place *p* is active.

Transition's label is compound from three elements: *cond*, *abort* and *action*. Two initial of them are given as logical expressions and they mean conditions of transition *t* enabling, weak and strong respectively. Strong condition (*abort*) is preceded by # and it is used for all input macroplaces of transition *t* preemption. The action component is created as the place's label, but subset associated with this element is active only one clock cycle.

Often a situation occurs in which internal states at selected hierarchy levels must be remembered. In HPN it is realized throughout ascribing history attribute *{H}* to selected macroplace. With token leaving a macro the tokens location in the adequate subnet is remembered. And after renewed macro activation tokens are inserted to the lately active places. For user convenience there is included a possibility of history attribute ascribing to all subordinated macros (operator *{H*}*).

With places the time parameters are associated. Ascribing of time *t* from discrete scale of time to

place p determines minimal activity time of this place. This means that output transition of place p will be enabled only after time t , after the moment of activation of place p moment. This solution provides in practice important possibilities in describing of strongly time dependent systems.

A mathematical model of HPN can be presented as follows.

Definition. Hierarchical Petri Net is shown as a tuple:

$$HPN = (P, T, F, S, \delta, \chi, \psi, \lambda, \alpha, \varepsilon, \tau) \quad (9)$$

where:

1. P is a finite non-empty set of places. In "flat" nets with places the capacity function was assigned $\kappa: P \rightarrow N \cup (\infty)$, which described max number of tokens in place p . For reactive systems descriptions the function equals 1 for each place $p \in P$ $\kappa(p)=1$.
2. T is a finite non-empty set of transitions. $N=P \cup T$ is the set of nodes.
3. F is a finite non-empty set of arcs, such that: $F=F_o \cup F_e \cup F_i$, where $F_o \subset (P \times T) \cup (T \times P)$ is a set of *ordinary* arcs, $F_e \subset (P \times T)$ is a set of *enabling* arcs, $F_i \subset (P \times T)$ is a set of *inhibitor* arcs. Moreover $(P \times P) \cup (T \times T) = \emptyset$.
4. S is a finite non-empty set of signals, such that: $S=I \cup O \cup L$, where I , O and L mean sets of input, output and internal signals respectively.
5. $\chi: P \rightarrow 2^N$, is a hierarchy function, describing set of immediate sub-nodes of place p .

Definition. A place $p \in P$ is called basic place if $\chi(p) = \emptyset$.

Definition. A place $p \in P$ is called macroplace if it isn't a basic place: $\chi(p) \neq \emptyset$.

Set of nodes N , being in hierarchy relation to place p creates a subnet. Condition of disjointing of all subnets assigned to macroplaces is required (no common nodes and arcs).

Definition. A place p is the lowest ancestor of node n' , such that: $n' \in \chi(p)$, what is described by: $la(n')=p$.

6. $\psi: P \rightarrow \{true, false\}$, is a boolean history function, assigning history attribute to each of place p , such that $\chi(p) \neq \emptyset$. For basic places the function is not defined.
7. $\lambda: N \rightarrow 2^S$, is a labeling function, assigning expression created from elements of set S to nodes from N . Places may be labeled only by the elements from set $O \cup L$ (labels mean actions assigned to places), label of a transition may be composed by following elements:
 - *cond* – created on the set $I \cup L$, being a Boolean expression imposed as a condition to transition t and generated by operators: $!$, $+$

and $*$, corresponding to logical operators: **not**, **or** and **and**, respectively;

- *abort* – created as a *cond*, but interpreted in another way (see the previous page), represented graphically by # at the beginning of the expression;
 - *action* – created on the set $O \cup L$, meaning action assigned to transition t , represented graphically by / at the beginning of the expression.
8. $\alpha: P \rightarrow \{true, false\}$, is an initial marking function, assigning attribute of initial place to the places $p \in P$.
 9. $\varepsilon: P \rightarrow \{true, false\}$, is a final marking function, assigning attribute of final place to the places $p \in P$.
 10. τ is a function, assigning numbers from discrete scale of time to each element from set of nodes N .

The operation of a net is determined by the movement of tokens. The rules of their movement are defined by conditions of transition enabling and action assigned to transition firing.

Let t_0 mean an activation moment of node $n \in N$. The function $\tau(n)$ assigns a number to node n at the moment t_0 : $\tau(n, t_0)=t$, where $t \in F$. In further instants the number is decremented and after time t it accomplishes value 0: $\tau(n, t_0+t)=0$.

Definition. P_p^{end} is a set of final places of a subnet assigned to macroplace p , such that:

$$\forall_{p' \in \chi(p)} \varepsilon(p') = true \Rightarrow p' \in P_p^{end} \quad (10)$$

Definition. $\xi: P \rightarrow P$ is a function of the set of final places, such that for macroplace p it returns its set of final places:

$$\xi(p) = P_p^{end} \quad (11)$$

The expression ξ^* denotes a transitive-reflexive closure of ξ function, such that for each $p \in P$ and $\chi(p) \neq \emptyset$ the following predicates hold:

- $p \in \xi^*(p)$,
- $\xi(p) \in \xi^*(p)$,
- $p' \in \xi^*(p) \Rightarrow \xi(p') \subseteq \xi^*(p)$.

Definition. A final marking of a subnet is such a marking, which contains all final places of that subnet.

Let be given a hierarchical HPN net and place p from the set of places of this net.

Definition. A set of input places of transition t is called $P_t^{in(o)}$, such that: $P_t^{in(o)} = \{p \in P: (p, t) \in F_o\}$.

Definition. A set of enabling places of transition t is called $P_t^{in(e)}$, such that: $P_t^{in(e)} = \{p \in P: (p, t) \in F_e\}$.

Definition. A set of inhibitor places of transition t is called $P_t^{in(i)}$, such that: $P_t^{in(i)} = \{p \in P: (p, t) \in F_{ij}\}$.

Definition. A set of output places of transition t is called P_t^{out} , such that: $P_t^{out} = \{p \in P: (t, p) \in F_o\}$.

Definition. The function $ac: P \rightarrow \{true, false\}$ is called a place activity function and it assigns *true* to each place that has a token, or *false* otherwise.

Definition. Let N^i mean a set of nodes assigned to a macroplace p by hierarchy function: $N^i = \chi(p)$, and let F^i mean set of all arcs joining nodes belonging to N^i . Then Z^i is a subnet assigned to macroplace p , such that: $Z^i = N^i + F^i$. All subnets assigned to macroplaces are required to be disjoint (no common nodes and arcs).

The conditions of transition t enabling:

$$\exists p \in P \quad la(t) = p \Rightarrow ac(p) = true \quad (12-a)$$

$$\forall p \in P_t^{in(o)} \cup P_t^{in(e)} \quad ac(p) = true \quad (12-b)$$

$$\forall p \in P_t^{in(i)} \quad ac(p) = false \quad (12-c)$$

$$cond(t) = true \quad (12-d)$$

$$\forall p \in P_t^{in(o)} \quad \chi(p) \neq \emptyset \Rightarrow \forall p' \in \xi^*(p) \quad (ac(p') = true) \quad (12-e)$$

$$and \quad \tau(p') = 0$$

$$\forall p \in P_t^{in(o)} \quad \tau(p) = 0 \quad (12-f)$$

$$abort(t) = true \quad (12-g)$$

Note: From all conditions the following logical expression can be composed (the general condition): $a*b*c*(d*e*f+g)$, that means a possibility of enabling transition t without the need to satisfy conditions d , e , and f if g is true. This situation is known as preemption.

The actions assigned to transition t firing:

$$\forall p \in P_t^{in(o)} \quad ac(p) := false \quad (13a)$$

$$\forall p \in P_t^{in(o)} \quad \chi(p) \neq \emptyset \Rightarrow \forall p' \in \chi^*(p) \quad ac(p') := false \quad (13-b)$$

$$\forall p \in P_t^{in(o)} \quad \forall s \in action(p) \quad s := false \quad (13-c)$$

$$\forall p \in P_t^{in(o)} \quad \chi(p) \neq \emptyset \Rightarrow \forall p' \in \chi^*(p) \quad \forall s \in action(p') \quad s := false \quad (13-d)$$

$$\tau(t) = 0 \Rightarrow \forall p \in P_t^{out} \quad ac(p) := true \quad (13-e)$$

$$\forall p \in P_t^{out} \quad \chi(p) \neq \emptyset \Rightarrow (\forall p' \in \chi^*(p) \quad ac(la(p')) = true) \quad (13-f)$$

$$\wedge \psi(la(p')) = false \Rightarrow ac(p') := \alpha(p')$$

$$\forall p \in P_t^{out} \quad \chi(p) \neq \emptyset \Rightarrow (\forall p' \in \chi^*(p) \quad ac(la(p')) = true) \wedge \psi(la(p')) = true \wedge \exists p'' \in \xi^*(la(p')) \quad ac(p'') = false \Rightarrow ac(p') := ac(p', t_e) \quad (13-g)$$

$$\forall p \in P_t^{out} \quad \chi(p) \neq \emptyset \Rightarrow (\forall p' \in \chi^*(p) \quad ac(la(p')) = true) \wedge \psi(la(p')) = true \wedge \forall p'' \in \xi^*(la(p')) \quad ac(p'') = true \Rightarrow ac(p') := \alpha(p') \quad (13-h)$$

$$\forall p \in P_t^{out} \quad \forall s \in action(p) \quad s := true \quad (13-i)$$

$$\forall p \in P_t^{out} \quad \chi(p) \neq \emptyset \Rightarrow (\forall p' \in \chi^*(p) \quad ac(p') = true) \Rightarrow \forall s \in action(p') \quad s := true \quad (13-j)$$

$$\tau(t, t_0) = \eta \Rightarrow \forall s \in action(t) \quad s := true \quad (13-k)$$

$$in < t_0, t_0 + \eta + 1 >$$

where $ac(p, t_e)$ means the state of possession (or not) of a token by place p at an instant, at which the token leaves place $la(p)$.

Note: Actions $e-j$ are performed when $\tau(t) = 0$. Action k is performed during all activity time of transition t .

9. CONCLUSION

A method of hierarchical decomposition of Petri nets is discussed in this paper; theoretical results on properties of the decomposition are presented. The proposed approach can be useful for verification of devices and systems, which behavior can be described by Petri nets. Certain analysis tasks, as we have shown, can be reduced to the analysis tasks for the elements of decomposition, which can remarkably reduce their complexity (in many cases depending exponentially on the net size). Further research work in this direction is going to be related with studying of application of the approach to some special verification tasks. Described model of hierarchical Petri net (HPN) is a useful tool for description and implementation of digital control systems. There are methods for formal analysis of selected properties of the model (e.g. liveness), but only for model without time dependencies. Further research will be steered also to extension of mentioned methods which can be applied to full HPN model.

9. REFERENCES

- [1] T. Murata. Petri nets: properties, analysis and applications, *Proceedings of IEEE 77* (1989). p. 541-580.
- [2] G. Andrzejewski. *Programowy model interpretowanej sieci Petriego dla potrzeb projektowania mikrosystemów cyfrowych*. Uniwersytet Zielonogórski, Zielona Góra, 2003.
- [3] А. Д. Закревский, *Параллельные алгоритмы логического управления*, Институт технической кибернетики АНБ, 1999.
- [4] G. Bruno et al. Scheduling hard real time systems using high-level Petri nets, *LNCS: Application and Theory of Petri nets* 616 (1992). p. 93-112. Springer-Verlag, Berlin, 1992.
- [5] A. Wegrzyn, M. Wegrzyn. Symbolic verification of concurrent logic controllers by means Petri nets. *Proceedings of International Conference "Computer-Aided Design of Discrete Devices (CAD DD'99)"*, Minsk, Belarus, 1999, Vol. 1, p. 45-50.
- [6] A. Zakrevskij, A. Karatkevich, M. Adamski. A method of analysis of operational Petri nets. *Proceedings of International Conference "Advanced Computer Systems (ACS'2001)"*, Poland, 2001, Kluwer Academic Publishers, Boston, 2002, p. 449-460.
- [7] K. Jensen. *Coloured Petri nets. Basic concept, analysis methods and practical use*. Springer-Verlag, Berlin, 1997.
- [8] G. Andrzejewski, A. Karatkevich. Interpreted hierarchical Petri nets in digital controller design, *Radioelektronika i Informatika* (1) (2005). p. 74-79.
- [9] C. A. Petri, *Kommunikation mit Automaten*, Institut für Instrumentelle Matematik, Bonn, 1962.
- [10] M. Adamski, M. Wegrzyn, P. Wolanski. A VHDL based approach to logic controller design. *Proceedings of International Conference "Programmable Device and Systems (PDS'98)"*, Gliwice, Poland, 1998, p. 9-16.
- [11] R.Esser: *An object oriented Petri net approach to embedded systems design*, PhD thesis, Swiss Federal Institute of Technology, Zurich, 1996
- [12] J.E.Hong, D.H.Bae: *HOONets: Hierarchical Object-Oriented Petri Nets for System Modeling and Analysis*, KAIST Technical Report CS/TR-98-132, November, 1998
- [13] S.Schof, M.Sonnenschein, R.Wieting.: *High-level Modeling with THORNs*, Proceedings of the 14th International Congress on Cybernetics, Namur, Belgium, 1995
- [14] J.M.Fernandes, M.Adamski, A.J.Proenca: *VHDL Generation from Hierarchical Petri Net Specifications of Parallel Controllers*. IEE Proc.: Computers and Digital Techniques, no. 144, vol. 2, Mar., 1997, pp. 127-137
- [15] T.Holvoet, P.Verbaeten: *Petri Charts: an Alternative Technique For Hierarchical Net Construction*. In IEEE Conference on Systems, Man and Cybernetics, 1995
- [16] R. David, H. Alla: *Petri Nets & Grafcet. Tools for modeling discrete event systems*. Prentice Hall, New York, 1992
- [17] D. Harrel. Statecharts: A Visual Formalism for Complex Systems. *Science of Computer Programming*, 8:231-274, 1987

Andrei Karatkevich, Received the Ph.D. degree from Belarusian State University of Informatics and Radioelectronics in 1998. Since 2000 he is an assistant professor at Technical University of Zielona Góra (since 2001 - University of Zielona Góra). His current research interests include Petri net theory and its applications, especially to problems related to analysis and verification of parallel or distributed control systems.



Grzegorz Andrzejewski, Received the Ph.D. degree from Technical University of Szczecin in 2002. Since 2002 he is an assistant professor at University of Zielona Góra. His current research interests include system level description of control systems and algorithms of their software and hardware synthesis.

