

Please cite this article as:

Henryk Piech, Piotr Borowik, Petri net synthesis based on model behavior, Scientific Research of the Institute of Mathematics and Computer Science, 2012, Volume 11, Issue 2, pages 103-111.

The website: <http://www.amcm.pcz.pl/>

PETRI NET SYNTHESIS BASED ON MODEL BEHAVIOR

Henryk Piech, Piotr Borowik

*Institute of Computer and Information Science
Czestochowa University of Technology, Poland
h.piech@adm.pcz.czest.pl, piotrborowikI@gmail.com*

Abstract: Modeled processes in this kind of networks are performed with help of Petri (PT) structure. In this realization, we are used tokens as well as sequenced and concurrent fired transitions. There are exploited graph systems and mathematic operators in known approaches. In the proposed approach we use behavioral imitation, based on states matrix. It is treated as a synthesis method and is realized in stages referring to states of modeled objects behavior. There aren't important functional dependences among state attributes in time. In the proposed algorithm, we are not limited by tokens number, which need not be constant. Effectiveness of placements exploitation is not on high level but simplicity of proposed algorithm and clarity of its results, which directly refer to states characteristics are the main advantages of the given method.

Introduction

A typical way of modeling multi-state objects is to specify their *placements* and *transitions* between placements. Simultaneously we describe the kind of *states attributes*. The respective models are called *transition systems*. They abstract away from the internal structure of placements and transitions and are formally graphs. Their paths represent runs of the modeled systems [1-3]. However, this way of modeling may be insufficient to reflect the behavior of systems in a world-wide network, meaning systems which are distributed in the sense that their attributes and activities consist of more or less independent components [4-6]. Events with disjoint states of related conditions are *independent* and may occur concurrently. Instead of studying a net's internal behavior as is usual in the Petri net theory, some authors restricts their view of the net to the outer interface and study how the net interacts with its environment by substituting it into an arbitrary Petri context and studying the behavior of the resulting [7, 8] larger system. Such nets are studied in same papers and characterize the canonical strong similarity, that arises by synthesizing a labeled transition system using the technique introduced by Leifer and Milner [9] and expanded by the authors in [10]. Leifer and Milner introduced the notion of reactive systems in [9] as a generalization of several situations that occur in the field of process calculus. Indeed, since the publication of Berry and Boudol's influential work on the chemical abstract machine [11, 12], it has become

commonplace to define the operational semantics via their reduction - often generated from basic rules and closed by substituting for parameters and into arbitrary reactive/evaluation contexts [13, 14]. Leifer and Milner were particularly interested in using their theory on examples where the contexts were not term algebras, but rather were of a graphical nature. This work, developed to some extent in Leifer's thesis [15], has led to the definition of specific graphs by Milner and Jensen [16]. Different approach regarding causal time in context allowing to model systems in an intuitive and modular way, with the potentiality of efficient verification [17-19]. For this purpose, it is presented a comparative case study concerning the railroad crossing problem and gives its specification in terms of timed automata as well as in terms of high-level nets with causal time.

1. Presentation set of states and attributes synthesis process

1. Let us describe states S with help of attributes:

$$S = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,m} \\ a_{2,1} & a_{2,2} & \dots & a_{2,m} \\ \dots & \dots & \dots & \dots \\ a_{v,1} & a_{v,2} & \dots & a_{v,m} \end{pmatrix} \quad (1)$$

where:

$a_{i,j}$ - attribute i in state j ,
 v - number of attributes,
 m - number of states.

2. Creation tables of tokens changing:

$$Tch = \begin{pmatrix} a_{1,2} - a_{1,1} & a_{1,2} - a_{1,2} & \dots & a_{1,m} - a_{1,m-1} \\ a_{2,2} - a_{2,1} & a_{2,3} - a_{2,2} & \dots & a_{2,m} - a_{2,m-1} \\ \dots & \dots & \dots & \dots \\ a_{v,2} - a_{v,1} & a_{v,3} - a_{v,2} & \dots & a_{v,m} - a_{v,m-1} \end{pmatrix} \quad (2)$$

3. Preparation the table with created transit connections between placements (the so called occupation table).

$$O = \begin{pmatrix} at_{1,1} = 0(1) & at_{1,2} = 0(1) & \dots & at_{1,n} = 0(1) \\ at_{2,1} = 0(1) & at_{2,2} = 0(1) & \dots & at_{2,n} = 0(1) \\ \dots & \dots & \dots & \dots \\ at_{n,1} = 0(1) & at_{n,2} = 0(1) & \dots & at_{n,n} = 0(1) \end{pmatrix} \quad (3)$$

where a connection will be created if $at_{i,j} = 1, n$ - number of transitions.

3. Calculation of the total sum of tokens changing for all columns in table

$$Tch: \{s_1, s_2, \dots, s_{m1}\}$$

4. Analysis for transitions creation:

a) when $s_j = 0$ for not occupied transitions and $\exists k: (a_{k,j+1} - a_{k,j}) \neq 0$ then without adding new placements are created inner connections to compensate tokens changing:

$$s_j = \sum_{i=1}^v (a_{i,j+1} - a_{i,j}) * (1 - at_{i,j}) = 0 \text{ and } \exists k: (a_{k,j+1} - a_{k,j}) \neq 0 \Rightarrow$$

$$\exists \{T(in(r)), at_{s,r} = 0, r = 1, 2, \dots, v \text{ or}$$

$$T(out(s)), at_{s,r} = 0, s = 1, 2, \dots, v\}: \sum_r w_r^+ = \sum_s w_s^-, \quad (4)$$

where:

$T(in(r))$ - set of transitions - predecessor placement r ,

$T(out(s))$ - set of transitions - successor placement s ,

b) when $s_j > 0$ for not occupied transitions and $\exists k: (a_{k,j+1} - a_{k,j}) \neq 0$ then we add new placement and new transition: $p_{v+i}, t_{v+i,k}, i = 1, 2, \dots, nad(in)$, where $nad(in)$ - number of added placements (transitions) with weights:

$$\sum_{i=1}^{nad(in)} w_{v+i,k} = s_j \text{ and } M(p_{v+i}) \geq w_{v+i,k}.$$

There are created inner and outer connections to compensate tokens changing:

$$s_j = \sum_{i=1}^{nad} (a_{i,j+1} - a_{i,j}) * (1 - at_{i,j}) = 0 \text{ and } k: (a_{k,j+1} - a_{k,j}) \neq 0 \Rightarrow$$

$$\exists \{T(in(r)), at_{s,r} = 0, r = 1, 2, \dots, v + nad(in) \text{ or}$$

$$T(out(s)), at_{s,r} = 0, s = 1, 2, \dots, v + nad(in)\}: \sum_r w_r^+ = \sum_s w_s^- + \sum_{i=1}^{nad} w_{v+i,k} w_{v+i,k}$$

$$\quad (5)$$

c) when $s_j < 0$ for non occupied transitions and $\exists k: (a_{k,j+1} - a_{k,j}) \neq 0$ then we add new placements and new transitions: $p_{v+i}, t_{k,v+i}, i = 1, 2, \dots, nad(out)$, where $nad(out)$ - number of added placements (transitions) with weights

$$\sum_{i=1}^{nad(out)} w_{k,v+i} = s_j.$$

Inner and outer connections are created to compensate tokens changing:

$$s_j = \sum_{i=1}^{v+1} (a_{i,j+1} - a_{i,j}) * (1 - at_{i,j}) = 0 \text{ and } \exists k: (a_{k,j+1} - a_{k,j}) \neq 0 \Rightarrow$$

$$\exists \{T(in(r)), at_{s,r} = 0, r = 1, 2, \dots, v + nad(out) \text{ or}$$

$$T(out(s)), at_{s,r} = 0, s = 1, 2, \dots, v + nad(out)\}:$$

$$\sum_r w_r^+ + \sum_{i=1}^{nad} w_{k_{v+i}} = \sum_s w_s^- . \quad (6)$$

Example 1. Let's try to trace synthesis process, according to states of modeled object presented with help of table S (Table 1). In this case the sum of attribute changes is equal zero $s_1 = 0$.

Table 1

Initial data of attributes of states

S	state1	state2	state3	state4	state5	state6/1
atrib1	0	1	2	1	3	0
atrib2	2	1	3	0	1	2
atrib3	0	1	2	1	3	0
atrib4	2	1	3	0	1	2

Then we create table of tokens changes (Table 2).

Table 2

Attributes changes, which should be adequate tokens changes

Tch	$a_{i,2} - a_{i,1}$	$a_{i,3} - a_{i,2}$	$a_{i,4} - a_{i,3}$	$a_{i,5} - a_{i,4}$	$a_{i,6} - a_{i,5}$
atrib1	1	1	-1	2	-3
atrib2	-1	2	-3	1	1
atrib3	1	1	-1	2	-3
atrib4	-1	2	-3	1	1

Comparing stage 1 and 2 and providing analysis we fire transition $t_{2,1}$ and $t_{4,1}$ with weight $w_{2,1}$ and $w_{4,1}$ and initial tokens $M(p_2) = 2$ $M(p_4) = 2$. Table of activated transitions is presented in Table 3.

Table 3

Table of transition firing for obtained state 2

	p1	p2	p3	p4
p1				
p2	1			
p3				
p4			1	

As result of the realized process, we obtain stage 2: $M(p_1) = attrib1 = 1, M(p_2) = attrib2 = 1, M(p_3) = attrib3 = 1, M(p_4) = attrib4 = 1$.

In the second stage we have to obtain set of attributes: $attrib1 = 2, attrib2 = 3, attrib3 = 2, attrib4 = 3$. Total changes are equal $s = 6 > 0$. Adding 4 placements p_5, p_6, p_7, p_8 ($M(p_5) = 1, M(p_6) = 2, M(p_7) = 1, M(p_8) = 2$) and firing transitions $t_{5,1}, t_{6,2}, t_{7,3}$ and $t_{8,4}$ (Table 4) with weights: $w_{5,1} = 1, w_{6,2} = 2, w_{7,3} = 1$ and $w_{8,4} = 2$ we obtain given levels of attributes (state 3).

Table 4

Transitions firings for obtained state 3

	p1	p2	p3	p4
p1				
p2	1			
p3				
p4			1	
p5	1			
p6		1		
p7			1	
p8				1

In the third stage we have to obtain set of attributes: $attrib1 = 1, attrib2 = 0, attrib3 = 1, attrib4 = 0$. Total changes are equal $s = -8 < 0$. Exploiting placements p_5, p_6, p_7, p_8 ($M(p_5) = 2, M(p_6) = 3, M(p_7) = 2, M(p_8) = 3$) and firing transitions $t_{1,5}, t_{2,6}, t_{3,7}$ and $t_{4,8}$ (Table 5) with weights: $w_{1,5} = 1, w_{2,6} = 3, w_{3,7} = 1$ and $w_{4,8} = 3$ we obtain given levels of attributes (state 4).

Table 5

Transitions firings for obtained state 4

	p1	p2	p3	p4	p5	p6	p7	p8
p1					1			
p2	1					1		
p3							1	
p4			1					1
p5	1							
p6		1						
p7			1					
p8				1				

In the fourth stage we have to obtain a set of attributes: $attrib1 = 3, attrib2 = 1, attrib3 = 3, attrib4 = 1$. Total changes are equal $s = 6 > 0$. Adding placements $p_9, p_{10}, p_{11}, p_{12}$ ($M(p_9) = 2, M(p_{10}) = 1, M(p_{11}) = 2, M(p_{12}) = 1$) and firing transitions

We can join the last two stages in one, treating them as sequenced or concurrent (because of their independent character: $\{t_{1,2}, t_{3,4}, t_{1,9} \text{ and } t_{3,11}\} \cup \{t_{6,10}, t_{8,12}\}$). Results of synthesis are exploited in creation of the structure algorithm of PT-net (Fig. 1).

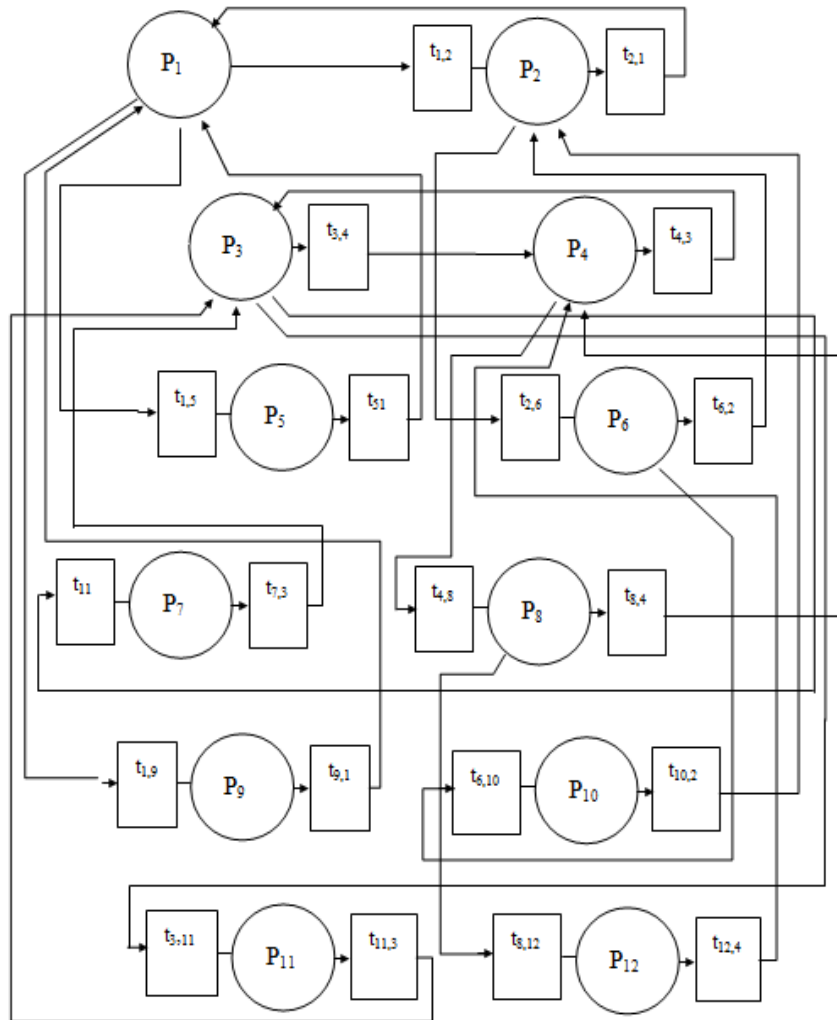


Fig. 1. The Petri net structure

Conclusions

The proposed synthesis method has a universal character, because refers to constant or various tokens number. Obtained form of PT net structure isn't optimal or

even reduced but the behavior algorithm is simple, with complexity on level $O(n^2)$. The number of placements adequating model attributes is usually several times less (in our example three times) than the number of all placements.

References

- [1] Cerone A., Maggiolo-Schettini A., Time based expressivity of time Petri nets for system specification, *Theoretical Computer Science* 1999, 216, 1-53.
- [2] Chiola G., On the structural and behavioral characterization of P/T nets, *International Workshop in Petri nets and Performance Model*, Toulouse 1993, 66-75.
- [3] Petri C.A., *Advanced Course on General Net Theory of Processes and Systems*, Springer Verlag, London 1979.
- [4] Berthelot G., Transformations and Decompositions of Nets. *Advanced in Petri nets*, Springer Verlag, London 1987, 250, 359-377.
- [5] Commoner F., *Deadlocks in Petri Nets*. Applied Data Research Inc., Wakefield, 1972.
- [6] Yakovlev A., Gomes L., *Hardware Designed Petri Nets*, Kluwer Academic Publishers, Norwell 2000.
- [7] Cortadella J., Jakovlev A., Rosenberg G., *Concurrency and hardware design: Advances in Petri nets*, Springer Verlag, 2549, New York 2002.
- [8] Murata T., *Petri Nets: properties, analysis, and applications*, IEEE 1989, 77, 4, 541-580.
- [9] Szpyrka M., *Fast and flexible modeling of real-time systems with RTCP- nets*, *Computer Science* 2004, 81-94.
- [10] Jensen K., Rozenberg G., *High-level Petri Nets - Theory and Application*, Springer Verlag, Berlin 1991.
- [11] Hollyday M.A., Vernon M.K., *A generalized timed Petri model for performance analysis*, *IEEE, Transaction of Software Engineering* 1987, SE-13, 12, 1297-1310.
- [12] Samolej S., Szmuc T., *Time extensions of Petri nets for modeling and verification of hard real-time systems*, *Computer Science* 2002, 55-76.
- [13] Marsan M.A., Balbo G., Conte G., Donatelli S., Franceschinis G., *Modeling with generalized stochastic Petri Nets*, John Wiley and Sons, New York 1995.
- [14] Valmari A., *Petri Net Newsletter*, 46, 1994, 6-14.
- [15] Peterson J.L., *Petri Net Theory and the Modeling of Systems*, Prentice Hall, New York 1981.
- [16] Memmi G., Vautherin J., *Analyzing Nets in Invariant Method*, *Advanced in Petri nets*, Springer Verlag, London 1987, 300-336.
- [17] Bowden F.D.J., *Modeling time in Petri nets*, *Workshop on Stochastic Models in Engineering*, ACM Press, New York 1994, 228-239.
- [18] Dadda L., *The synthesis of Petri nets for controlling purposes and the reduction of their complexity*, *Euromicro*, 2002.
- [19] Zuberek W.M., *Timed Petri nets, definitions, properties, and applications*, *Microelectronics and Reliability* 1991, 31, 4, 627-644.