MATEC Web of Conferences **44**, 01012 (2016)
DOI: 10.1051/matecconf/20164401012
© Owned by the authors, published by EDP Sciences, 2016

The methodology for constructing the linear and linear-cyclic basic fragments by means of decomposition of PN-model

Dmitry Pashchenko[1,a], Dmitriy Trokoz[1], Nikolas Konnov[1], Andrew Krasnov[1], Paul Chechel[1] and Tania Pashchenko[1]

[1]Penza State University, Department of Computer Science, 440000 Penza Krasnaya 40, Russia

Abstract. This article gives a brief analysis of the subject area and the approaches to the analysis and synthesis of models of complex systems structures were discussed. A number of problems occurred; these problems can be effectively solved with using of the method generally considered in this article. This article describes an approach to the construction of a system of linear and linear-cyclic basic fragments as a part of the method for obtaining the equivalent PN-model with tensor methods by decomposition of initial one. The rules describing the implementation of the various stages of the construction of a system of linear and linear-cyclic base fragments were introduced. An alternative method of vertical division of transitions in the PN-structure was suggested, it allows saving all available links.

1 Introduction

Currently, there is a continuous increase in the complexity of computing systems for information processing in all modern human activities. Each computing system has a number of parameters, and importance of which varies in different fields of application. The tendency to optimize the parameters of operation of a finite computing system requires the developers to do hard work in the modeling stage, where a fundamental working model of functional processes of developed system is constructed [1-6].

There are many methods of analysis and synthesis of models of complex systems' structures, based on a structured approach [7], the methods of the upstream and downstream designing, metamodelling and others. However, as it is noted by Kulagin [8], the contradiction between the complexity made by modern systems and traditional approaches to their designing, today one of the main issues of the theory of systems is defined, it is development of a methodology and the designing of systems of computer-aided design, solving not support, but the main task of synthesis.

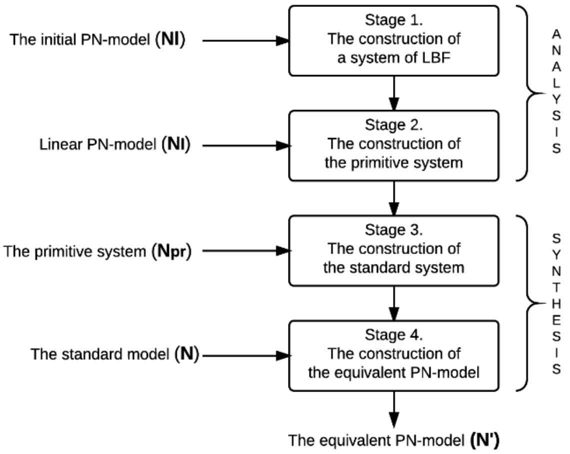
The requirements described above have led to the emergence and development of the system approach to the design of complex systems, the concept of which does not contradict the classical structural approach. In accordance to it, there are the following stages:

1. the development of a number of hypotheses concerning the structure of the subsystems of which the projected system will consist of;
2. the formation of the finished candidate structures from the given sub-systems;
3. analyzing of each structure in order to determine the characteristics that allows reducing the number of candidate structures and ultimately selecting the final structure.

In his works V. P. Kulagin is considering described above method for the analysis of complex systems, based on the synthesis of a variety of candidates, and offers the following way for practical implementation: to use Petri nets as structure models of systems.

2 Way to build alternatives

The approach defining the mechanism for constructing alternative variants (the equivalent PN-models) is presented below.


The initial PN-model (**NI**)
Stage 1. The construction of a system of LBF
Linear PN-model (**Nl**)
Stage 2. The construction of the primitive system
The primitive system (**Npr**)
Stage 3. The construction of the standard system
The standard model (**N**)
Stage 4. The construction of the equivalent PN-model
The equivalent PN-model (**N'**)
ANALYSIS
SYNTHESIS

[a] Corresponding author: dmitry.pashchenko@gmail.com

Figure 1. The scheme of stages of constructing the equivalent PN-model.

The following definitions are introduced:

- *The initial PN-model* ($\mathbf{N_I}$) – is a model, for which it is necessary to do an analysis;
- Linear PN-model ($\mathbf{N_l}$) – is PN, obtained by decomposition of the initial PN-model ($\mathbf{N_I}$) and given by the set of linear and linear-cyclic basic fragments (LBF);
- The primitive system ($\mathbf{N_{pr}}$) – is a set of elementary nets, the power of which is defined by the number of transitions of linear PN-model ($\mathbf{N_l}$);
- *The standard model* (**N**) – is PN, obtained from a primitive system by union of a set of vertices given in advance *(by performing of a program of synthesis);*
- The equivalent PN-model (**N'**) – is PN, obtained from the standard model by means of tensor of transformation;

As part of this article the first stage is considered, the problems and an implementation details - decomposition of the initial PN-model and the construction of a system of linear and linear-cyclic basic fragments (LBF) in the example of the initial structure of Petri net and the vector of initial marking given below.

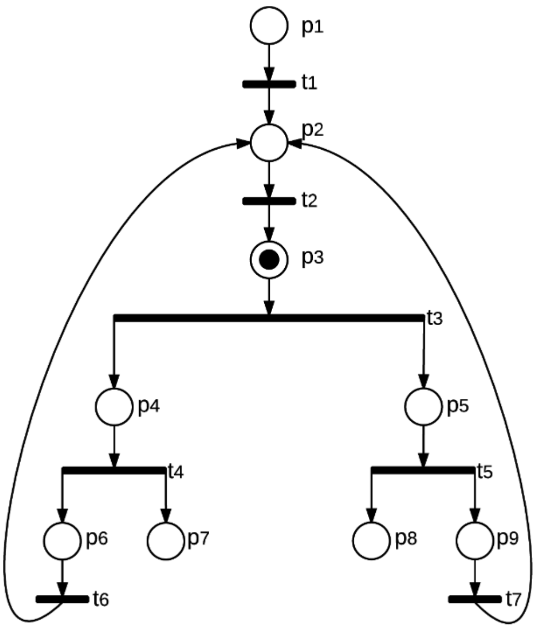


Figure 2. The initial PN-structure.

Table 1. The initial vector of marking.

P1	P2	**P3**	P4	P5	P6	P7	P8	P9
0	0	**1**	0	0	0	0	0	0

For the decomposition of the initial PN-model and the construction of a system of linear and linear-cyclic basic fragments it is required:

1. to identify sequentially and separate the shortest cycles;
2. to select alternately the branches with the maximum length.

As a result, the initial PN-model is transformed into a non-connected graph consisting of a system of linear and linear-cyclic base fragments.

Let's prepare an initial PN-model for convenient selection of cycles, dividing vertically all the transitions that have more than one input or (and) one output position. The method described below is an alternative one - in contrast to one proposed by Kulagin [2], it allows to split the transition without losing any links.

3 The vertical dividing of transition

If for a transition t_i the following conditions are true:

$$\mathrm{pre}(t_i) = \{p_{11,}\ p_{12,\dots,}p_{1m}\}, \mathrm{post}(t_i) = = \{p_{21,}\ p_{22,\dots,}p_{2n}\}; \quad (1)$$

and

a) $$\mathrm{pre}(t_i) = \{p_{11,}\ p_{12,\dots,}p_{1m}\}, \mathrm{post}(t_i) = = \{p_{21,}\ p_{22,\dots,}p_{2n}\};\ \mathrm{m} > n, \quad (2)$$

- when n > 1 the transition is divided into transitions t_i^1, t_i^2, for which it is true that:

$$\mathrm{pre}(t_i^1) = \{p_{11,}\ p_{12,\dots,}p_{1m}\}, \mathrm{post}(t_i^1) = \{p_{21}\} \quad (3)$$
$$\mathrm{pre}(t_i^2) = \{p_{11,}\ p_{12,\dots,}p_{1m}\}, \mathrm{post}(t_i^1) = =\{p_{22,}\ p_{23,\dots,}p_{2n}\}; \quad (4)$$

- when n = 1 the transition is divided into transitions $t_i^1, t_i^2, \dots, t_i^n$, for which it is true that:

$$\mathrm{pre}(t_i^1) = \{p_{11}\}, \mathrm{post}(t_i^1) = \{p_{21}\}$$
$$\mathrm{pre}(t_i^2) = \{p_{12}\}, \mathrm{post}(t_i^2) = \{p_{21}\} \quad (5)$$
$$\dots$$
$$\mathrm{pre}(t_i^n) = \{p_{1n}\}, \mathrm{post}(t_i^n) = \{p_{21}\}$$

b) m < n ,

- when n > 1 the transition is divided into transitions t_i^1, t_i^2, for which it is true that:

$$\mathrm{pre}(t_i^1) = \{p_{11}\}, \mathrm{post}(t_i^1) = \{p_{21,}\ p_{22,\dots,}p_{2n}\}; \quad (6)$$
$$\mathrm{pre}(t_i^2) = \{p_{12,}\ p_{13,\dots,}p_{1m}\}, \mathrm{post}(t_i^1) = =\{p_{21,}\ p_{23,\dots,}p_{2n}\}; \quad (7)$$

- n = 1; then the transition is divided into transitions $t_i^1, t_i^2, \dots, t_i^n$, for which it is true that:

$$\mathrm{pre}(t_i^1) = \{p_{11}\}, \mathrm{post}(t_i^1) = \{p_{21}\}$$
$$\mathrm{pre}(t_i^2) = \{p_{11}\}, \mathrm{post}(t_i^2) = \{p_{22}\} \quad (8)$$
$$\dots$$
$$\mathrm{pre}(t_i^n) = \{p_{11}\}, \mathrm{post}(t_i^n) = \{p_{2n}\}$$

After vertical dividing of all the transitions, which have more than one input or (and) one output position the initial PN-model was transformed to the following form.

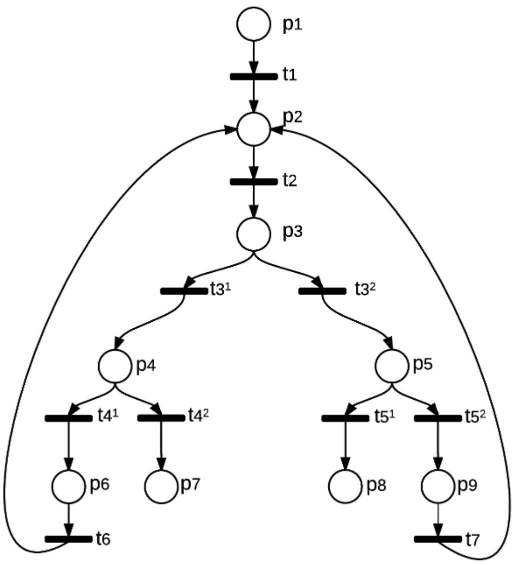


Figure 3. PN-structure after the vertical dividing of transitions.

In next step we will select all the cycles in the transformed PN-structure. We construct the adjacency matrix of transitions from the row-vertex to the column-vertex:

Table 2. The adjacency matrix.

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9
P_1	0	**1**	0	0	0	0	0	0	0
P_2	0	0	**1**	0	0	0	0	0	0
P_3	0	0	0	**1**	**1**	0	0	0	0
P_4	0	0	0	0	0	**1**	**1**	0	0
P_5	0	0	0	0	0	0	0	**1**	**1**
P_6	0	**1**	0	0	0	0	0	0	0
P_7	0	0	0	0	0	0	0	0	0
P_8	0	0	0	0	0	0	0	0	0
P_9	0	**1**	0	0	0	0	0	0	0

Let us use the following algorithm:

If a vertex has only input or only output arcs, then it obviously does not occur in any cycles. You can remove all such vertices from the graph with all the related arcs.

As a result, new vertices will occur, having only input or only output arcs. They are also to be removed. The iterations are repeated as long as the graph does not stop changing. The absence of any changes indicates the absence of cycles, if all the vertices have been removed. Otherwise, all the remaining vertices necessarily belong to cycles.

Table 3. The adjacency matrix after the applying of algorithm for cycles searching.

	P_2	P_3	P_4	P_5	P_6	P_9
P_2	0	**1**	0	0	0	0
P_3	0	0	**1**	**1**	0	0
P_4	0	0	0	0	**1**	0
P_5	0	0	0	0	0	**1**
P_6	**1**	0	0	0	0	0
P_9	**1**	0	0	0	0	0

As a result of the applying of algorithm, two cycles have been selected in the converted PN-structure:

- $\{P_2, P_3, P_4, P_6, P_2\}$
- $\{P_2, P_3, P_5, P_9, P_2\}$

In the next step we alternately separate the shortest cycles of the PN-structure. Due to the fact that the resulting cycles have equal length, we separate the first obtained cycle from the PN-structure.

During the separating of vertices cycle, transitions that intersect with the rest of the PN-structure (with more than one input or output position or transition) also will be divided taking into account copying of token if it presents there.

In this case, after separating of the first cycle, the second one exists no longer - as the transition **t2** remained in the first cycle.

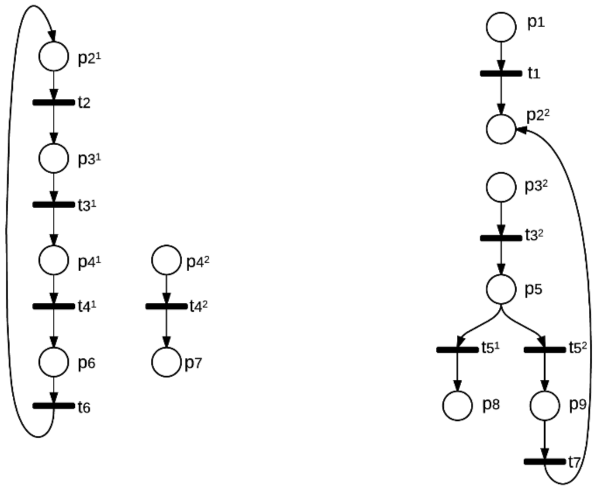


Figure 4. PN-structure after the separating of a cycle.

Because of the fact that all the cycles were separated from the PN-structure, we go to the final stage of the transformation of the initial PN-structure to the system of linear and linear-cyclic basic fragments – we select linear fragments, taking into account the requirement to minimize redundancy and condition for the separation of only those vertices and transitions that have more than one input and output position or transition, as well as rule for copying of token, taking into account its presence.

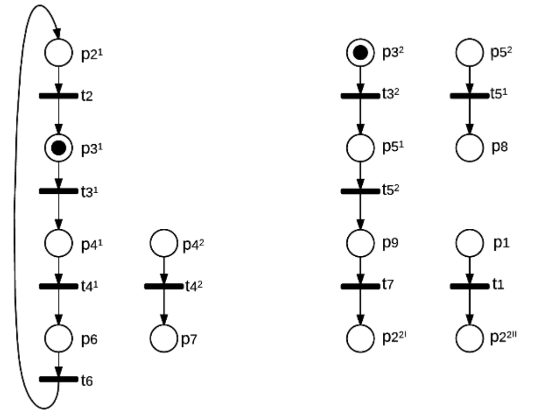


Figure 5. The system of LBF of the initial PN-structure.

The system of LBF obtained taking into account the introduced rules, is the minimal possible one, and it is equivalent to the initial PN-structure due to the using of an alternative method of vertical dividing of transitions with saving of all the initial links. This LBF system can be used in the following stages of the algorithm of *constructing an equivalent model* [9-10].

The example illustrating the algorithm for *constructing the LBF system* was based on a relatively simple initial PN-structure. To work with PN-structures organized more complexly, software for automation of constructing of LBF system is required [11-12]. Further it is supposed to integrate software for the automation of constructing of LBF-systems with the software products designed to automate the following stages of the algorithm for constructing the equivalent PN-models in a common comprehensive software solution for the user.

Acknowledgment

The development of software for automating of the construction of LBF system is performed within RSF grant to conduct fundamental scientific research and exploratory scientific research involving young researchers on the theme "Analysis and synthesis of complex net systems based on tensor and transformational methods" (convention № 15-11-10010).

References

1. D. A. Zaitsev. IEEE Transactions on Systems, **44**, 1–12 (2013)
2. M. Beccutti, E. Bibbona, A. Horvath, R. Sirovich, A. Angius, G. Balbo. App. and Theory of Petri Nets and Con., 273–293 (2014)
3. T. Chatain, S. Haar. Transact. on Petri Nets and Other Models of Con. IX, 83–98 (2014)
4. K. S. Cheung. *Augmented Marked Graphs* (2014)
5. S. Y. Li, A. M. An, Y. Wang. Journal of Intelligent Manufact., **24**, 1157–1173 (2013)
6. S. Li, Y. Che, W. Liu, H. Wu, Y. Cai. Inform. Science and Manag. Engineer. II, **3**, 19–28 (2015)
7. D. Pashchenko, D. Trokoz, N. Konnov, M. Sinev. Procedia Computer Science, **49**, 99–103 (2015)
8. V. P. Kulagin. *Razrabotka teoreticheskih osnov i instrumental'nyh sredstv avtomatizirovannogo proektirovanija struktur parallel'nyh vychislitel'nyh sistem na osnove tenzornogo ischislenija setevyh modelej* (1995)
9. W. Reisig. *Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies* (2013)
10. M. Diaz. *Petri Nets: Fundamental Models, Verification and Applications* 656 (2013)
11. E. Kizilov, D. Pashenko, D. Trokoz, N. Konnov 5th International Workshop on Computer Science and Engineering Information Processing and Control Engineering, WCSE 2015-IPCE, 185-190 (2015)
12. A. V. Dubravin, S.A. Zinkin, D. V. Paschenko. Int. Siberian Conf. on Control and Communic., (2015)