

**Conference Paper**

**Web Application for Visual Modeling of Discrete Event Systems**

Stetsenko, I.V., Dyfuchyn, A. and Leshchenko, K. and Davies, J.

This is a paper presented at the 7th IEEE Int. Conference on Internet Technologies and Applications ITA-17, Wrexham, UK, 12-15 September 2017

Copyright of the author(s). Reproduced here with their permission and the permission of the conference organisers.

---

**Recommended citation:**

Stetsenko, I.V., Dyfuchyn, A. and Leshchenko, K. and Davies, J. (2017) 'Web Application for Visual Modeling of Discrete Event Systems'. In: Proc. 7th IEEE Int. Conference on Internet Technologies and Applications ITA-17, Wrexham, UK, 12-15 September 2017, pp. 86-91. doi: 10.1109/ITECHA.2017.8101916

# Web Application for Visual Modeling of Discrete Event Systems

Stetsenko Inna V., Difyuchyn A., Leshchenko K.

Faculty of Informatics and Computer Science  
NTUU “Igor Sikorsky Kyiv Polytechnic Institute”  
Kyiv, Ukraine  
stiv.inna@gmail.com, difuchin@gmail.com,  
katrean@inbox.ru

Davies John

Department of Computing  
Glyndwr University  
Wrexham, U.K.  
j.n.davies@glyndwr.ac.uk

**Abstract—** This research work has resulted in the development of a web application that enables discrete event systems simulation to be created using a Petri-object approach. It provides the development of a model in two stages. In the first stage, the dynamics of the classes of objects are created using Petri net. In the second stage, the model is composed of objects with given dynamics. The simulation algorithm is based on stochastic Petri net with multichannel transitions and is implemented using Ruby. The web application enables the design of the model's dynamics by manipulation with graphics objects and saving it not only as a graphics object but also as a program method. This greatly improves the overall performance of the simulation model development.

**Keywords**—stochastic Petri nets; Simulation Algorithm; Ruby on Rails; Petri net visual software; Petri-object simulation

## I. INTRODUCTION

Models of discrete event systems are at the heart of information control systems and decision support systems. So the quality of the control processes depends greatly on the model quality. The results of the modeling process are the main task of the simulation, however, the development of software component that can be used to simulate the system (and be the component of information control or decision-making systems) are an important factor. Since the systems have a high level of technical complexity then it results in the use of approaches that have certain characteristics. Modelling systems require design models with identical elements, flexible modeling of dynamic elements, a visual representation of the model and opportunities for its adjustment and modification. Modeling flexibility entail researchers to detail the process to the smallest elements, but the demands for convenience representation of models entail their more abstract definition.

## II. BACKGROUND

The theory of Discrete Event Systems and a specification can be found in [1] which defines it as a generalization of queuing systems. This is used in most software simulation including commercially available modeling packages e.g. Arena[13], ExtendSim[14], Plant Simulation[15] etc. The system is represented as a collection of blocks configured to perform certain functions typically waiting, processing, equipment, transportation and others [2]. However the

programming of the control elements is difficult in these models because the algorithms that define the operation blocks are not accessible. For example, the cyber-attack scenario or grid resource broker, or traffic control elements are unable to be represented with blocks of the enterprise model. By changing the parameters of the resource settings in the simulation introduces a new state of resources (as “inoperative”, “damage”), take into account the information about system's state in control elements etc.

An alternative approach involves the use of Petri net graphs which has an advantage over other modeling systems since it is based on a mathematical modeling language. Petri net provides an elegant and mathematically rigorous modelling framework for discrete event dynamic systems [12]. It is described as a directed bipartite graph with state-transitions. Transitions represent the events of system and places represent the conditions that force the events. Directional arcs connect transitions (rectangles) to places (circles) which hold tokens and vice versa. The transition occurs when for each input place the following condition is satisfied: the number of tokens is at least equal to the weight of arc that leads from the input place to the transition. Transition's firing is performed by deleting tokens in input places of transition and adding tokens in output places of transition in accordance the weight of arc.

The tokens outputs occur with a determined time delay for timed Petri net. If stochastic Petri net is considered the time delay can be given by stochastic value. The functioning of timed Petri net differs largely from the Petri net without time delays. For example, the fragment of timed Petri net in Fig.1 presents the performance of two processes, which conflict for capture resource.

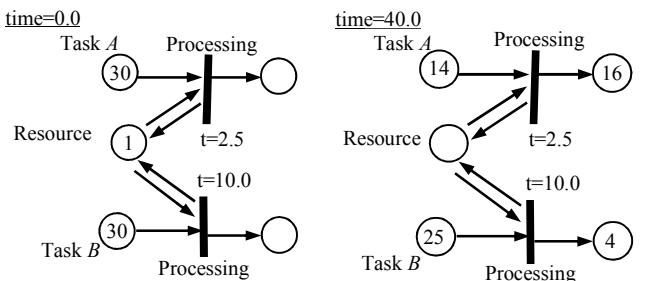


Fig. 1. The functioning of timed Petri net

In the case of Petri net without time delays the equal quantities of tasks which are performed must be obtained. However, in the case of timed Petri net, the one process is four times more efficient.

The tokens inputs in multichannel transition are repeated until the firing condition is satisfied Fig.2. If ordinary transitions are used then one hundred transitions are needed for the same fragment of Petri net. So the use of multichannel transitions reduces the number of elements for the model representation.

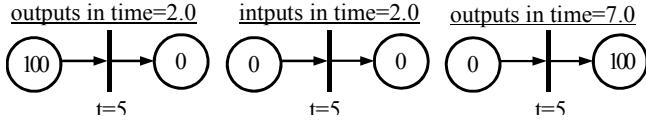


Fig. 2. The functioning of multichannel transition

Input place can be connected with a transition by information arc. This means that by deleting tokens from such place is not performed when the transition fire has occurred. For example, when a car moves over the crossroad the condition of “green light” still persists. Another example, when an attacker uses the vulnerability it doesn’t mean that the vulnerability has vanished, so the appropriate condition should not change.

In Fig.2 the transition has a condition “Permission” that doesn’t change if the transition is fired. So two inputs occur at the one moment and two moments of outputs of tokens are memorized. When the time is over the output of tokens is performed.

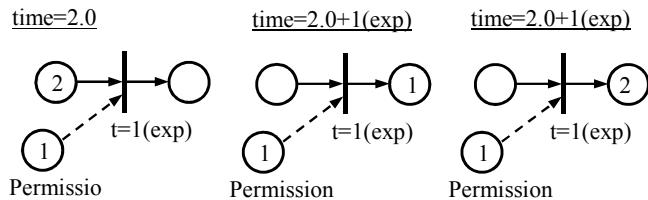


Fig. 3. The functioning of multichannel transition with information arc

When using Petri net for complex systems researchers are compelled to decompose it either into a simpler investigation as in [7] or for a more efficient model with parallelism as in [8].

This research provides a software modeler which allows the Petri-object approach for model designing to be used. The theoretical foundations of Petri-object simulation is outlined in [6]. According to this approach, the discrete event system’s model composes of elements, the dynamics of which are represented by stochastic Petri net. Adopting this approach guarantees that the dynamics of the system also represents a stochastic Petri net. So the simulation algorithm is the same for elements as for the system. Another advantage of Petri-object approach is that it provides a significant reduction of simulation time for systems with a large number of elements.

Visual programming tools are intended to simplify the process of Petri-object model construction, reduce the number of errors caused by the wrong linking of elements, and increase the perception of a simulation model.

### III. RELATED WORK

#### A. Applications of Petri-nets

Applications in many different areas have been modeled using Petri net graphs, examples of how the range has expanded recently are highlighted here. Classically Petri nets were used in the area of manufacture and business applications. The use of timed Petri nets for flexible manufacturing systems is considered in [3]. The theory and practice of using Petri nets for modeling business processes is outlined in [4]. Petri nets seminal role for formalization of business processes is unveiled in [5]. More recently applications in the computer systems area have appeared including: Communications systems modelled [17] [18], electronic Hardware Design [19], Formal Methods in PLC Programming [20], Concurrent Object-Oriented Programming [21] and Verification of protocols and performance evaluation of networks [22]. A more exhaustive list can be found at [16].

#### B. Petri net Simulators

There are many Petri net Simulators available a more exhaustive list can be found at [23]. However the most widely used versions are summarized in table 1 showing comparisons of their characteristics, uses and disadvantages with the web application DESS (Discrete Event Systems Simulation) that is developed.

TABLE 1. COMPARISON OF PETRI NETS SIMULATORS

	Object-oriented techniques	Stochastic Petri Nets	Web graphic editor	Token Game Animation	Fast Simulation	Reachability Graph Analysis
Coopn (Concurrent Object-Oriented Petri net builder)	+	-	-	+	+	-
JSARP (Simulator and Analyzer Petri net in Java)	+	-	-	+	+	+
PNTalk	+	-	-	+	+	
Renew	+	-	-	+	+	-
CPN (Coloured Petri nets) Tools	-	+	-	+	+	-
Petri.NET Simulator	-	-	-	+	+	-
WoPeD (Workflow Petri Net Designer)	-	-	-	+	-	-
PIPE2 (Platform-Independent Petri net Editor)	-	+	-	+	+	-
DESS (Discrete Event Systems Simulation)	+	+	+	+	+	-

The greatest advantage of DESS is the unique object-oriented approach to building model with Petri nets. This new innovative concept allows the representing of a Petri net as a parameter of object's constructor. Then many objects can be simply created with the same Petri net. That allows user to quickly create the list of Petri-objects and then set links between them.

Also there is another unique feature, a web-based graphic editor. Petri nets can be created or edited on any computer or even on a smartphone or tablet that are connected to the internet.

Looking at the most popular object-oriented Petri nets simulators it can be seen that only some of them supports stochastic Petri Nets. The support of stochastic Petri Nets is very important for realistic simulation of complex systems.

So it can be seen that DESS has some major advantages that make the process of modeling systems more comfortable and easier.

#### IV. PETRI-OBJECT APPROACH

##### A. Petri-object Definition

Petri-simulator is defined as a class that realize the simulation of some real object in accordance the stochastic timed Petri net which describes the object's dynamics. The Petri net is saved in the field 'net' and can be specified for every object. Then any subclass of this class implements the simulation the same way. The name given to the object of class Petri-simulator or its subclass is the *Petri-object*.

Stochastic Petri net with multichannel transitions is used, the mathematical description of which and state equations can be found in [9].

Petri net, that represents the object's dynamics, is created using the static function of class NetLibrary and then transferred to Petri-object's constructor as an argument. This approach provides the ability of using the same method of the class NetLibrary for creating Petri nets of plurality of similar objects.

Petri-objects have all properties of the object (in object-oriented terminology), simulate the dynamics of object by the stochastic Petri net, and they are constructive elements which make up the Petri net of complex system.

##### B. Petri-object Model Definition

The Petri-object model is the model that aggregates (in object-oriented terminology) the Petri-objects. The dynamics of Petri-objects is connected in two ways: 1) the *common places*; 2) the *event initializing* (Fig.4).

In the first case the connection is given by assigning memory addresses to appropriate places. In the second case the relationship between the transition of one object and the place of another object is set by token passing along the connection when the transition is fired. By connecting it in this way it has been proved that the dynamics of model is described by the stochastic Petri net composed of its Petri-objects nets is guaranteed [6]. So this provides a computable model.

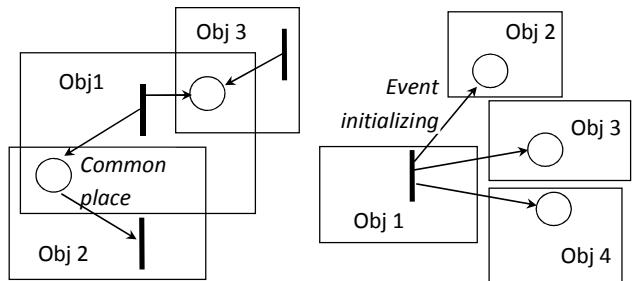


Fig. 4. Petri-objects dynamics connection

##### C. Petri-object Simulation Algorithm

The Petri-object model simulation algorithm is built in line with the stochastic timed Petri net function. Current time is promoted from one moment of event to the nearest next slot. In every time slot the tokens outputs and tokens inputs must be calculated. Tokens outputs are performed for transitions which have moments of output equaling to current moment. This transformation of Petri net is called  $D^+$ . In the same moment tokens inputs are performed in transitions where the fire conditions are true. Because of multichannel transitions tokens inputs are repeated while any transitions fire condition is true. This transformation of Petri net is called  $(D^-)^m$  (Fig.5)

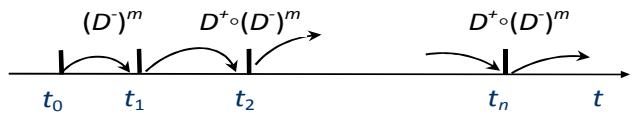


Fig. 5. The changing of state of the timed Petri net

It has been proven that transformation  $D^+$  of the model equals the transformations  $D^+$  of Petri nets of all its Petri-objects. Similarly for transformation  $(D^-)^m$ . The partition on Petri-objects allows the number of elementary operations needed to implement transformation of model's Petri net to be reduced. The nearest moment of event from the set of nearest moments of event which are saved in Petri-objects. Transformation  $D^+$  is performed only for Petri-objects which are the nearest moment of event equal to current moment. Transformation  $(D^-)^m$  is performed for every Petri-object but the fact that tokens inputs are repeated in every Petri-objects  $m_j \leq m$ , so there is also a significant reduction of elementary operations for this transformation.

If more than one Petri-object has the nearest moment of token output then there is a conflict of Petri-objects. To select one Petri-object from the set of conflicting objects the set is sorted by priority and then a random selection from all Petri-objects with the highest priority is performed.

Computing complexity of Petri-object simulation is investigated in [10]. The mathematical evaluation of complexity and the results of experiments are close. Polynomial evaluation of algorithm complexity has been obtained. So this simulation method can be implemented for complex systems with a large number of elements.

## V. CLIENT-SERVER SOFTWARE APPLICATION FOR PETRI-OBJECT SIMULATION

The software for the discrete event systems simulation is developed based on client-server architecture. Web-oriented implementation ensures cross-platform operation, and enables the use of models from open warehouse, and supports the advanced operation for modelers' communication and collaboration. The web application is implemented using Ruby on Rails because it is open-source, provides compact design and simpler implementation and reduces routing in web programming [11].

The main purpose of the software is to provide fast and safe development of the simulation model. Visual means of representation of models simplify their perception and reduce the number of errors in its construction. The process of model construction is as follows: firstly, the dynamics of Petri-objects is designing with the use of graphics elements of stochastic Petri nets: places, transitions, arcs. All graphics elements are defined with base manipulation operations: create, drag and drop, delete and edit parameters (Fig.6).

Any transition should be determined with the following parameters: the value of time delay, the value of priority and the value of probability. The value of time delay can be given by stochastic or determine value including zero value but, it must be a nonnegative value. The value of priority is defined as a positive integer value. The value of probability is defined by double value in the interval  $[0; 1]$ . By default the parameters of transition have the values: zero time delay, 1 priority, and 1.0 probability.

The number of tokens should be determined for every place. The default value of this parameter is zero.

Any arc is determined by the number of links and the Boolean value specifying if the arc is the information. By default, the arc creates as ordinary (non-information) with one link.

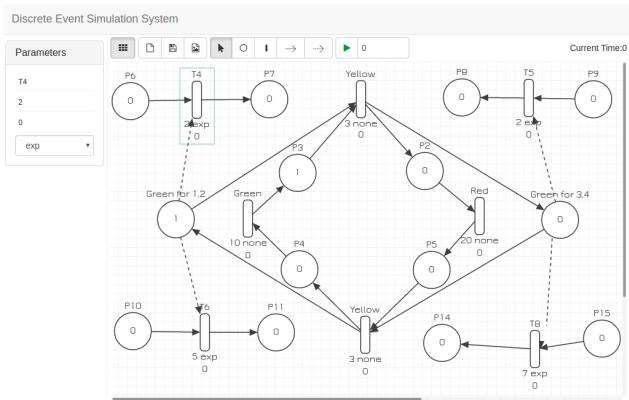


Fig. 6. Creating a Petri net

The dynamics can be seen by running the Petri net, which is built, using either the animation mode or simulation mode. The simulation results include protocol events, the values of average, maximum and minimum marking places and the values of average, maximum and minimum loading of transitions. A reports panel allows the user to view the

information about all events that occurred during the simulation and the statistics for each element of Petri nets. At the end of creating the Petri net it is saved as a program method with dynamic parameters as its arguments.

Secondly, the Petri-objects are created and saved in the model list. They are created with the given Petri net in the form of appropriate program method with given parameters.

Thirdly, the connections between Petri-objects are determined visually. Users can choose the object from the model's list and determine the connection with other objects of the model. This then allows the model to be saved and run. It is important that the model is saved so that the program components can be run or transformed in graphical images and be modified.

The transformation of the graphical images to a program method is performed by analyzing the image and coding automation. Users can modify the list of arguments of the method and its name. If a Petri-object is opened the reverse transformation from program method to graphic image is also provided automatically. The reflection is used for analyzing the program method and a graphical image is recovered based only on the information that the method contains. Users can modify it and save as a new method.

User authorization is carried out through an internal account or using one of the social networks: Github, Google +, Facebook. User activities include: saving and opening models in Open Warehouse and it also allows other developers to share models and modify their models.

Experimental research of time performance has been carried out with the Petri-object model which consisted of  $n$  Petri-objects with  $k = 5$  sequential events for each one. Common places are used for objects connections in series. This model construction allows flexible control of the number of events  $nk$ . Taking one Petri-object  $nk$  with sequential events the simulation of stochastic Petri net was obtained. This allows the comparison of the implementations of the Petri-object model and the stochastic Petri. The results are shown in Fig.7, which confirms the theoretical polynomial evaluation of model complexity given in [10]. The simulation algorithm of the Petri-object model provides twice the reduction of time performance in comparison with the simulation of stochastic Petri net.

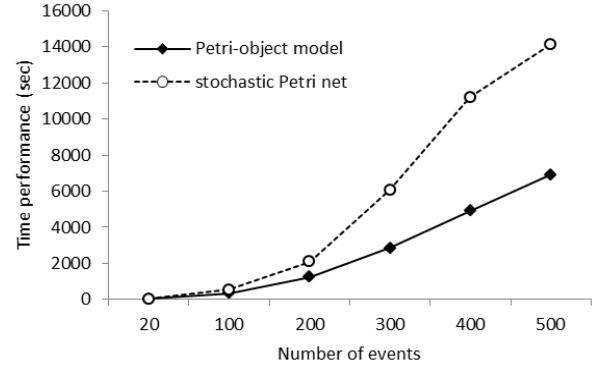


Fig. 7. Time performance dependency of model complexity

TABLE 2. COMPARISON OF RUBY AND JAVA TIMINGS

Parameters		Ruby			Java			tRuby/tJava	
No of objects	No of transitions, n	Petri-object model time, tPO	Petri-net time, t	t/tPO	Petri-object model time, tPO	Petri-net time, t	t/tPO	for Petri-object model time	for Petri-net time, t
20	100	343.73	548.95	1.6	4.54	8.37	1.8	76	66
40	200	1238.15	2094.15	1.7	17.11	32.22	1.9	72	65
60	300	2862.23	6080.37	2.1	38.88	69.61	1.8	74	87
80	400	4920.18	11206.46	2.3	70.90	118.73	1.7	69	94
100	500	6917.14	14131.52	2.0	122.64	177.73	1.4	56	80

Table 2 shows the results obtained using a simulation on a Java desktop compared to the Ruby implementation. A factor of approximately two times reduction is achieved when using the Petri-object model simulation algorithm. However the Ruby implementation is slower in 70 times compared to Java.

## VI. RESULTS

The architecture and software of the web application implementation enables the Petri-object approach to design simulation model of discrete event system to be developed. It is intended to provide the cross-platform, open source, fast and safe development of a simulation model.

The Petri-object simulation technology provides a convenient model construction and fast simulation algorithm, so it is easy to implement for a systems with a large number of elements. By setting multichannel transition, allows the representation of almost any elementary event with either deterministic or stochastic time delay, with different ways of resolving the conflict with other events. Settings of arc include the parameter that allows taking into account, but not changing the state of input places. This provides powerful elementary tools of the model's representation.

Saving Petri nets as a program method gives significant reduction in the memory required to store models. Transformation Petri net's program method to graphic image and vice versa provides the desirable flexibility of modeling. Using one Petri net for creating many elements with similar dynamics reduces the time the effort spent on the modeling. Classes of Petri-objects can be inherited, and Petri net of a superclass can be modified in its subclass.

Saving models in open warehouse allows users to share their models and so accumulate models of systems for different application areas.

## VII. CONCLUSION

The discrete event system's implementation as a Petri-object model enables the system's dynamics of the dynamics of its elements to be created using the unified representation of stochastic Petri nets.

A Petri net is created from a Petri-object which has many elements with similar dynamics by using an object-oriented approach. A significant reduction of computable complexity

of the simulation algorithm is achieved by dividing the model into Petri-objects.

Petri-object model's construction techniques enables the user to concentrate, firstly, on creating the dynamics of base elements of the model, secondly, on creating the elements with given parameters and thirdly, on connecting elements to create the model's dynamics.

The developed web application not only simplifies the model's design but also provides the creation of program components for use in the model's simulation. Instead of saving graphics images of all nets of Petri-objects a method is proposed for creating a Petri net of objects with given parameters and provide the transformation of the Petri net from its graphics image to a method and vice versa.

Additionally some services of the web application are intended to organize the modelers' communication and collaboration, and to form the open models warehouse.

### A. Future Work

Future development will include the improvement of visual programming tools by the transformation of the graphic image of the model to program code and vice versa.

Additionally Open warehouse of Petri-objects needs improving.

The implementation of a parallel simulation algorithm is being considered for future development.

## REFERENCES

- [1] B. Zeigler, H. Praehofer, T. Gon Kim "Theory of Modeling and Simulation," New York: Academic Press, 2000.
- [2] A.M. Law, "Simulation Modeling and Analysis," New York: McGraw-Hill International, 2014.
- [3] M. Zhou, K. Venkatesh "Modeling, simulation, and control of flexible manufacturing systems: a Petri net approach," World Scientific Publishing, 1999.
- [4] W.M.P. van der Aalst and C. Stahl "Modeling Business Processes: a Petri Net-Oriented Approach", MIT Press, 2011.
- [5] W.M.P. van der Aalst "Business Process Management as the "Killer App" for Petri Nets," Software & Systems Modeling, vol. 14, issue 2, Springer, 2015, pp. 685 – 691.
- [6] I.V. Stetsenko, "Theoretical basis of Petri-object simulation," Mathematical Machines and Systems, no. 4, 2011, pp.135-150 (in Russian)
- [7] W. M. P. van der Aalst "Decomposing Petri Nets for Process Mining: A Generic Approach," Distributed and Parallel Databases, Berlin: Springer-Verlag, , vol. 31, no. 4, 2013, pp. 471-507.,
- [8] D.A. Zaitsev "Decomposition of Petri Nets," Cybernetics and Systems Analysis, vol. 40, no. 5, 2004, pp. 739-746.

- [9] I.V. Stetsenko, "State equations of stochastic timed petri nets with informational relations," *Cybernetics and systems analysis*, vol. 48, no 5, 2012, pp.784-797.
- [10] I.V. Stetsenko, V. Dorosh, A. Dvychyn "Petri-Object Simulation: Software Package and Complexity," Proceedings of the 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2015), Warsaw (Poland), 2015, pp. 381-385.
- [11] M. Hartl "Ruby on Rails Tutorial: Learn Web Development with Rails," Third edition, Addison-Wesley Professional Ruby Series, Addison-Wesley 2015.
- [12] N. Chapman Petri Net Models ISE-2 Surprise 97 Project [http://www.doc.ic.ac.uk/~nd/surprise\\_97/journal/vol2/njc1/](http://www.doc.ic.ac.uk/~nd/surprise_97/journal/vol2/njc1/) accessed 01/02/2017
- [13] Arena, <https://www.arenasimulation.com/> accessed 05/02/2017
- [14] ExtendSim, <http://www.extendsim.com/> accessed 06/02/2017
- [15] Plant Simulation, [https://www.plm.automation.siemens.com/en\\_gb/products/tecnomatix/manufacturing-simulation/material-flow/plant-simulation.shtml](https://www.plm.automation.siemens.com/en_gb/products/tecnomatix/manufacturing-simulation/material-flow/plant-simulation.shtml)
- [16] M.Haustermann(2015), "Applications of Petri Nets", University of Hamburg,<https://www.informatik.uni-hamburg.de/TGI/PetriNets/applications/> accessed 02/02/2017
- [17] J. Billington, M. Diaz, G. Rozenberg (Eds.), "Application of Petri Nets to Communication Networks" Lecture Notes in Computer Science, vol. 1605, Springer-Verlag, 1999, ISBN: 3-540-65870-X.
- [18] H. Ehrig, W. Reisig, G. Rozenberg, H. Weber (Eds.), "Petri Net Technology for Communication-Based Systems" Lecture Notes in Computer Science, vol. 2472, Springer-Verlag, 2003, ISBN: 3-540-20538-1.
- [19] A. Yakovlev, L. Gomes, L. Lavagno (Eds.), "Hardware Design and Petri Nets" Kluwer Academic Publishers, March 2000, ISBN: 0-7923-7791-5.
- [20] G. Frey, L. Litz (Eds.), "Formal Methods in PLC Programming", Proceedings of the IEEE SMC 2000, Nashville, TN, 8-11 October 2000.
- [21] G.A. Agha, F. De Cindio, G. Rozenberg, (Eds.), "Concurrent Object-Oriented Programming and Petri Nets", Springer-Verlag, Berlin, 2001, ISBN: 3-540-41942-X.
- [22] D.A. Zaitsev, "Clans of Petri Nets: Verification of protocols and performance evaluation of networks", LAP LAMBERT Academic Publishing, 2013, 292 p. ISBN: 978-3-659-42228-7
- [23] Petri Nets Tools Database Quick Overview <https://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/quick.html/> accessed 11/03/2017