PN Matlab Toolbox 2.0

Martina Svadova[1], Zdenek Hanzalek[1]

[1] Center for Applied Cybernetics, Czech Technical University

Abstact. This paper introduces PN Matlab Toobox 2.0 further called PN Toolbox. PN Toolbox is software package containing a new version of Matlab toolbox for Petri nets and newly developed graphical editor called PN Editor. PN Editor allows to draw a Petri net and its export to Petri Net Markup Language (PNML). The matlab toolbox for Petri nets supports the behavior analysis of discrete Petri nets, continuous Petri nets (CPN) and hybrid Petri nets (HPN).

1 INTRODUCTION

Petri nets are one of possible graphical tools, which can be used for modeling, analysis of various systems, since they offer profound mathematical background especially from graph theory and linear algebra. There are many various Petri nets tools, which offer convenient graphical environment and allow simulation and analysis of various Petri Net classes, but on the other hand mostly tools have very limited possibility of extensions to problems specifically needed for given application. Some of tools are accessible in source code, but these software projects are relatively large, difficult to modify and platform dependent.

Matlab has the following advantages:

- Matlab runs on many platforms (Windows, Unix, Linux).
- Even for beginners it is quite easy to implement and run simple algorithm. Matlab manipulations are usually taught in basic subjects from control engineering, so students are already familiar with this environment when they come to the subjects dealing with Petri Nets.

There is a wide range of toolboxes helping to implement sophisticated algorithms. Since these toolboxes are mainly designed for continuous space systems it is very attractive to use them for hybrid Petri Nets.

Approach adopted in this article is based on the three steps:

- Petri net modeling in graphical editor (PN Editor)
- export from PN Editor to Matlab compatible format matrixes
- Matlab based Petri net analysis and visualization of simulation results (in Matlab or in PN Editor)

This approach helps to organize a work in a modular way, to use standard libraries and to build own tools. In other words one is no more using a 'universal' tool but he/she is programming his/her own tool with support in a modeling and visualization stage. This is quite more convenient because no tool is universal enough.

The paper is organized in three sections. The first section describes graphical interface PN Editor. The second one explains newly developed functions of the Matlab toolbox or Petri nets. The last section shows examples with using functions of PN Matlab Toolbox.

2 PN EDITOR

The first version of matlab tolbox for Petri nets had not own graphical editor and PM Editeur was used as graphical interface. Since this editor is not suited for continuous and hybrid Petri nets, a new editor, called PN Editor, was developed.

Key properties of PN Editor are:

- possibility to draw more types of Petri nets
- platform independent
- support of PNML
- possibility to extend the editor by additional functions

PN Editor (**Fig.1**) was developed as user-friendly graphical interface for Matlab toolbox for Petri nets, which allows to draw discrete PNs, continuous Petri nets, hybrid Petri nets and extended hybrid Petri nets [5], [6], [7]. PN Editor is JAVA based application, which was designed to be platform independent.

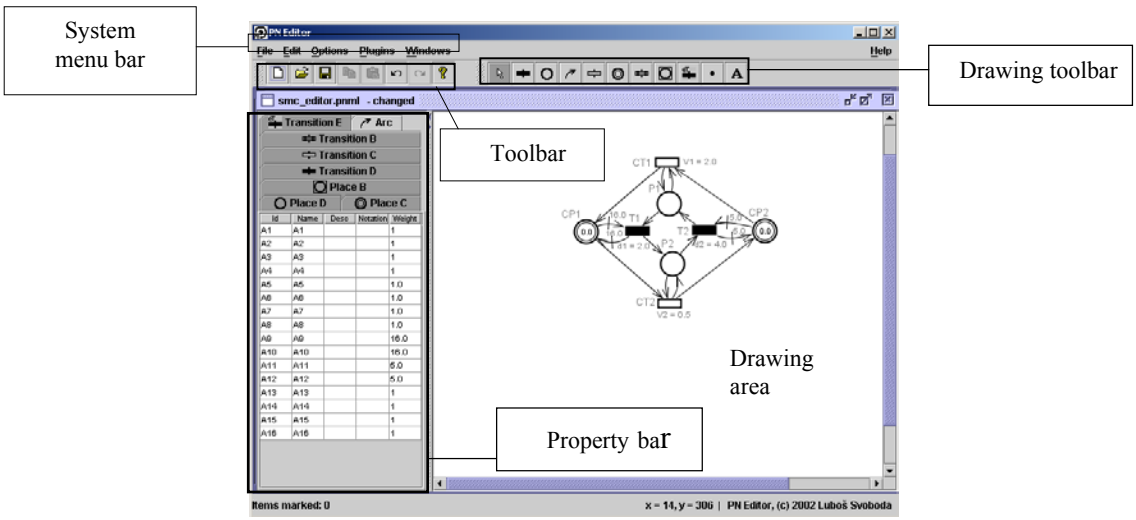

Fig.1. Graphical editor PN Editor

PN Editor supports Petri Net Markup Language (PNML). PNML is XML-based interchange format for Petri nets (it determines Petri nets saving format) and it is described in [3]. Due to possibility of import/export of files in PNML format, PN Editor allows maximal preservation of compatibility with other tools supporting PNML.

Unfortunately PNML standard is not fully defined for some specific properties of Petri nets or internal information. This is solved by the element “toolspecific”, which determines start of section of additional information designed only for a given editor. An attribute of element toolspecific expresses in which editor additional information was designed. The elements, which editor doesn’t know, are ignored. Default values are set for the elements that are missing in an import file.

The editor was originally designed as environment for graphical interpretation of Petri nets, which is transformed to matrix form suitable for processing in Matlab. Since using of some functions in Matlab was limited (e.g. displaying possibilities of Matlab), PN Editor can be extended by plugins up to specific user requirements. PN Editor contains plugins to display a graph of reachable markings and evolution graph for continuous or hybrid Petri nets.

3 MATLAB TOOLBOX FOR PETRI NETS FUNCTIONS

PN matlab toolbox contains functions for analysis of Petri net properties. There are functions for PN import (matrices *Post, Pre, M0*), calculation of incidence matrix *W*, minimal standardized P–invariants and deadlock detection. The PN Toolbox also contains token player for T-timed and stochastic Petri nets and the function for generating of graph of reachable markings. In comparison with previous version of PN matlab toolbox [4], a new version was extended by functions described in following chapters.

3.1 Import / Export functions

The following functions import a structure and parameters of Petri nets created in graphical editor PN Editor. A function *pnml2stpn.m* is used to import stochastic and timed Petri nets, a function *pnml.m* is used in the case of autonomous Petri nets and a function *pnml2hpn.m* is used in the case of continuous and hybrid Petri nets.

Syntax:

```
[Pre,Post,M0]=pnml(filename)
[Pre,Post,M0,d,TypeT]=pnml2STPN(filename)
[Pre,Post,M0,V,d,TypeP,nofpd] = pnml2HPN(filename)
```

Input parameter:
filename - name of file, into which Petri net, drawn in graphical editor PN Editor, is saved.

Output parameters:
Pre - matrix of pre-conditions
Post - matrix of post-conditions
M0 - column vector of initial marking of Petri Net
d - column vector of time delay associated to the transitions
V - vector of maximal speeds
nofpd- the number of discrete places
TypeT - column vector of transitions types:
0 - zero timed transition
1 - timed transition
2 - stochastic time transition with uniform distribution
TypeP - column vector of places types
1 - discrete place
2 - continuous place

Export functions are used to export the results calculated in matlab for its further visualization in PN Editor. A function *graph2hpn.m* is used to export parameters for visualization of an evolution graph of continuous and hybrid Petri nets. The functions have no output parameters.

Syntax:

```
graph2hpn(Mar,VV,event_sum,time_event,mtimings,loopback,nofpd
,filename)
```

Input parameter:
filename - name of file, into which modified input parameters of function for export to PN Editor, are saved
Mar - matrix of markings; each column vector of the matrix consists of a discrete and a continuous part. The discrete part represents marking of discrete places in the IB-

state, the continuous part represents marking of continuous places at the start of each IB-state;

VV - matrix of instantaneous firing speeds

Time_XC - row vector of relative time (j-th entry of Time_XC corresponds to the j-th IB state)

event_sum - column vector of events; each entry of vector is list of events that evoked passage from one IBstate to following one

time_event - column vector of times, when particular events occurred (time elapsed in previous IB-state)

mtimings - matrix of timings; each entry of column *timing vector* expresses remaining firing delay of discrete transition (*noftd* represents the number of discrete transitions).

Loopback - index of IB-state following *IBq* (the evolution graph has terminal node when *q* = *loopback*)

3.2 Evolution graph of continuous Petri nets function

The function generates the evolution graph for continuous Petri nets.

```
[Mar,VV,Time_XC] = evolgr(Pre,Post,M0,V,Prio)
```

The input and output parameters have been described above.

3.3 Evolution graph of hybrid Petri nets function

The function generates evolution graph of hybrid Petri nets. The results of function can be visualized in PN Editor. The algorithm of the function was designed to be applicable for simulation of behavior of systems modeled by discrete PN, continuous PN and hybrid Petri nets. The algorithm can be used only for an analysis the nets with bounded discrete part. The algorithm works with single server semantics and it solves conflicts. Discrete transitions do not reserve the tokens.

Syntax:

```
[Mar,VV,event_sum,time_event,mtimings,loopback] =
HPN(Pre,Post,M0,V,TypeP,d,Prio)
```

Input parameters:

TypeP - column vector that contains type of places of a net (discrete, continuous)

The output and the other input parameters have been described in previous paragraphs.

3.4 Displaying of markings evolution of chosen places

Function *drawnpl.m* displays marking evolution of chosen places during all IB-states of the evolution graph. This function is usually used in junction with the function generating the evolution graph, but it can be also used separately.

```
drawnpl(Mar,time_event,loopback,nofpd,drawnDP,drawnCP,sizeF)
```

input parameters:

drawnDP - vector of discrete places, whose marking evolution will be displayed

drawnCP - vector of continuous places, whose marking evolution will be displayed

sizeF - font size of the text in the graph

4 EXAMPLE

The example shows a sequence of steps during finding of an evolution graph of hybrid Petri net showed in **Fig.2**

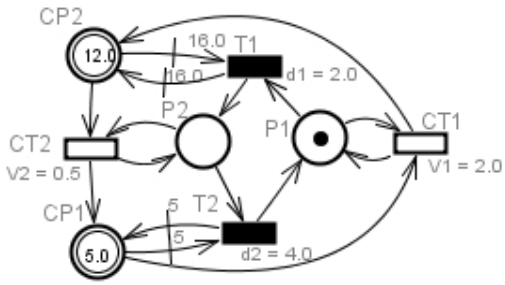

Fig.2. Hybrid Petri net example

1. Save the net in **Fig.2** as file *net.pnml* in PN Editor environment

2. Import the net structure using function *pnml2hpn.m*

```
[Pre,Post,M0,C] = pnml2hpn(net.pnml)
```

3. Calculate the matrices needed for the evolution graph construction (**Fig.3**)

```
[Mar,VV,event_sum,time_event,mtimings,loopback] =
HPN(Pre,Post,M0,V,TypeP,d,Prio)
```

$$Mar = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 5 & 1 & 0 & 0 & 5 & 7 \\ 12 & 16 & 17 & 17 & 12 & 10 \end{pmatrix}$$

$$VV = \begin{pmatrix} 2 & 2 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0.5 & 0.5 & 0 \end{pmatrix}$$

$$event\_sum = \begin{pmatrix} [\,] \\ m(P4) = 16 \\ m(CP1) = 0 \\ T1 \\ m(P3) = 5 \\ T2 \\ m(P4) = 16 \end{pmatrix}$$

$$mTimings = \begin{pmatrix} 2 & 2 & 1.5 & 2 & 2 & 2 & 2 \\ 4 & 4 & 4 & 4 & 4 & 4 & 4 \end{pmatrix}$$

$$Time\_event = \begin{pmatrix} 0 & 2 & 0.5 & 1.5 & 10 & 4 & 3 \end{pmatrix}$$

$$loopback = 2$$

Fig.3. Output matrices of function HPN.m

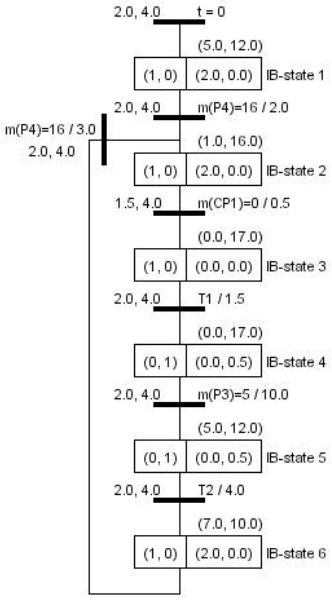

Fig.4. The evolution graph of HPN in **Fig.2**

4. Visualize of the evolution graph in PN Editor

```
Graph2hpn(Mar,VV,event_sum,time_event,mtimings,loopback,nofpd
,net.hpn)
```

Launch PM Editor -> in the system menu chose item *"Plugins"* -> choose item *"evolution graph for HPN*" -> choose file *"net.hpn "*

Fig.4 shows the evolution graph of hybrid Petri net in **Fig.2** displayed in PN Editor

5. Marking evolution of discrete place P1 and continuous places CP1 and CP2

```
drawnpl(Mar,time_event,loopback,nofpc,drawnDP,drawnCP,sizeF)
```

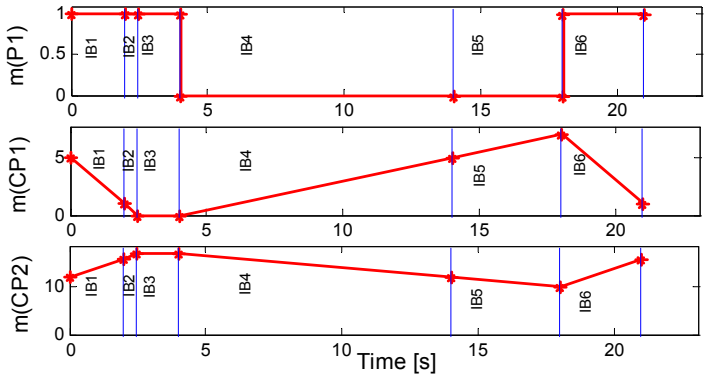

Fig.5. Evolution marking of chosen places of HPN in **Fig.2** during IB-states

PN Toolbox is available at web site: http://dce.felk.cvut.cz/cak/Research/PN/index_PN.htm.

This work was supported by the Ministry of Education of the Czech Republic under Project LN00B096.

5 REFERENCE

[1] Tadao Murata: "Petri Nets: Properties, Analysis and Applications" Proceedings of the IEEE, vol. 77, No. 4, April 1989.

[2] in: C. Girault, W. Reisig (eds): Application and Theory of Petri Nets, Informatik Fachberichte No.52, Springer (1982), 301-310.

[3] Jungel, Matthias. Kindler, Ekkart. Weber, Michael. The Petri Net Markup Language. http://www.informatik.hu-berlin.de/top/pnml/download/JKW_PNML.ps.

[4] Matlab Toolbox for Petri Nets.
In: Tool Demonstrations. Newcastle upon Tyne: University of Newcastle, 2001, p. 35-39.

[5] David,R., H. Alla. 1993. "Autonomous and timed continuous petri nets". In: *Advances in Petri nets*, Springer-Verlag.

[6] David,R., H. Alla. 1998. "Continuous and Hybrid Nets". In: *Journal of Circuits, Systems and Computers*, Volume 8, No.1, p. 159-188.

[7] David,R., H. Alla. 2001. "On Hybrid Petri Nets". In: *Discrete event dynamic systems: Theory and applications*, Volume 11,p. 9-41.

6 CONTACT

Center for Applied Cybernetics
Czech Technical University
Karlovo nam. 13
121 35, Prague 2, Czech Republic
{xsvadova, hanzalek}@control.felk.cvut.cz