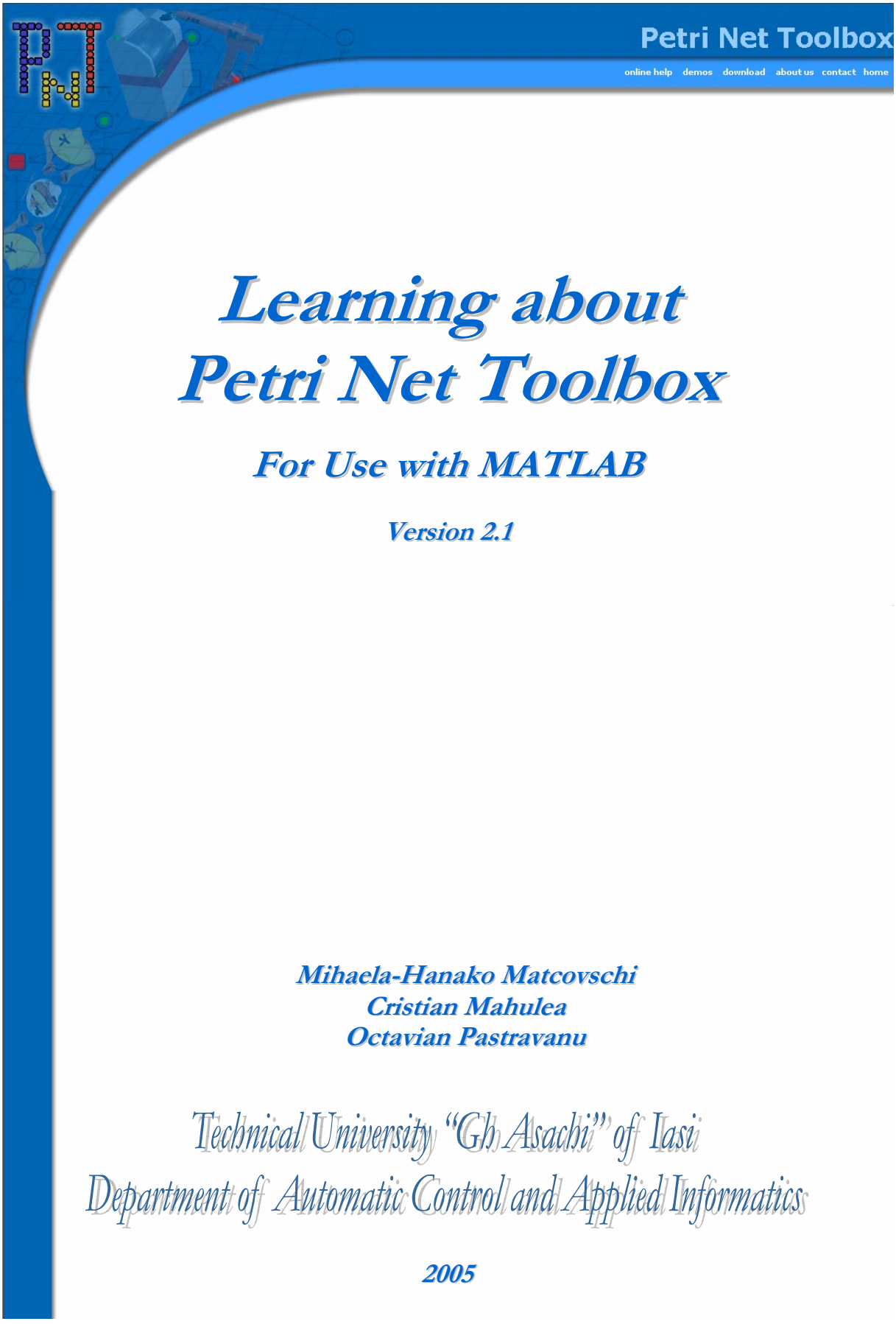


Learning about Petri Net Toolbox

For Use with MATLAB

Version 2.1

Mihaela-Hanako Matcovschi
Cristian Mahulea
Octavian Pastravanu

Technical University "Gh Asachi" of Iasi
Department of Automatic Control and Applied Informatics

2005

Contents

Introduction.

Petri Net Toolbox at a First Glance

Preface. The ***Petri Net Toolbox*** (***PN Toolbox***) is a software tool for simulation, analysis and design of discrete event systems, based on *Petri net* (PN) models. This software is embedded in the MATLAB environment and its usage requires the **MATLAB** version 6.1 or higher.

The integration with the MATLAB philosophy presents the considerable advantage of creating powerful algebraic, statistical and graphical instruments, which exploit the high quality routines available in MATLAB. Moreover, this MATLAB orientation of the ***PN Toolbox*** was intended to permit further development, including tools devoted to hybrid systems, because MATLAB incorporates comprehensive software for studying continuous and discontinuous dynamics.

Unlike other PN software, where places are meant as having finite capacity (because of the arithmetic representation used by the computational environment), our toolbox is able to operate with infinite-capacity places, since MATLAB includes the built-in function **Inf**, which returns the IEEE arithmetic representation for positive infinity.

The ***PN Toolbox*** was designed and implemented at the **Department of Automatic Control and Applied Informatics** of the **Technical University „Gh. Asachi" of Iaşi**, **Romania**.

Expected Background. To derive a full benefit from reading this guide, you should be familiar with:

- the formal and graphical exploitation of PNs as a highly effective tool for modeling discrete event dynamical systems;
- the concepts and use of MATLAB programming language.

Types of Petri Nets. In the current version of the ***PN Toolbox***, five types of classic PN models are accepted, namely: untimed, transition timed, place-timed, stochastic and generalized stochastic. The timed nets can be deterministic or stochastic, and the stochastic case allows using the appropriate distribution function.

Main Features. The ***PN Toolbox*** has an easy to exploit ***Graphical User Interface*** (GUI), whose purpose is twofold. First, it gives the user the possibility to draw PNs in a natural fashion, to store, retrieve and resize (by Zoom-In and Zoom-Out features) such drawings.

Second, it permits the simulation, analysis and design of the PNs, by exploiting all the computational resources of the environment, via the global variables stored in the MATLAB's Workspace. All the net nodes and arcs are handled as MATLAB objects so as changes in their parameters can be approached through the standard functions **set** and **get**.

When starting to build a new PN model, the user must define its type by appropriate checking in a dialogue box. The properties of the net-objects (places, transitions, arcs) depend on the selected type of model.

After drawing a PN model, the user can:

- visualize the ***Incidence Matrix***, which is automatically built from the net topology;
- explore the ***Behavioral Properties*** (such as liveness, boundedness, reversibility etc.) by consulting the ***Coverability Tree***, which is automatically built from the net topology and initial marking;
- explore the ***Structural Properties*** (such as structural boundedness, repetitiveness, conservativeness and consistency):
- calculate ***P-Invariants*** and ***T-Invariants***;
- run a ***Simulation*** experiment;
- display current results of the simulation using the ***Scope*** and ***Diary*** facilities;
- evaluate the global ***Performance Indices*** (such as average marking of places, average firing delay of transitions, etc.);
- perform a ***Max-Plus Analysis*** (restricted to event-graphs);
- ***Design*** a configuration with suitable dynamics (via automated iterative simulations).

The ***PN Toolbox*** uses the *XML* format for saving a model. A short description of the file format and the specific commands is given in Appendix 1 of this documentation.

To ensure flexibility, the set of all default values manipulated by the ***PN Toolbox*** are accessible to the user by means of a configuration file presented in Appendix 2.

Product Support. The users are kindly asked to notify the authors of this toolbox of any possible deficiency and to suggest possible improvements.

Contact Adress.

Department of Automatic Control and Applied Informatics
Technical University "Gh. Asachi" of Iasi
Blvd. Mangeron 53A, Iasi, 700050, Romania
Tel./Fax: +40-232-230.751
E-Mail: pntool@delta.ac.tuiasi.ro

Part I.

Describing the Graphical User Interface (GUI)

I.1. Overview

The ***PN Toolbox*** is equipped with an easy to exploit GUI (fig. I.1), which allows drawing PNs and permits simulation, analysis and synthesis. This GUI becomes operational once the command
>> pntool
is typed at the MATLAB's prompt.

There are two modes in which the ***PN Toolbox*** may be exploited, namely the ***Draw Mode*** that allows the user to build a new PN model or modify the properties of an existing one, and the ***Explore Mode*** that enables the user's access to simulation, analysis and design tools.

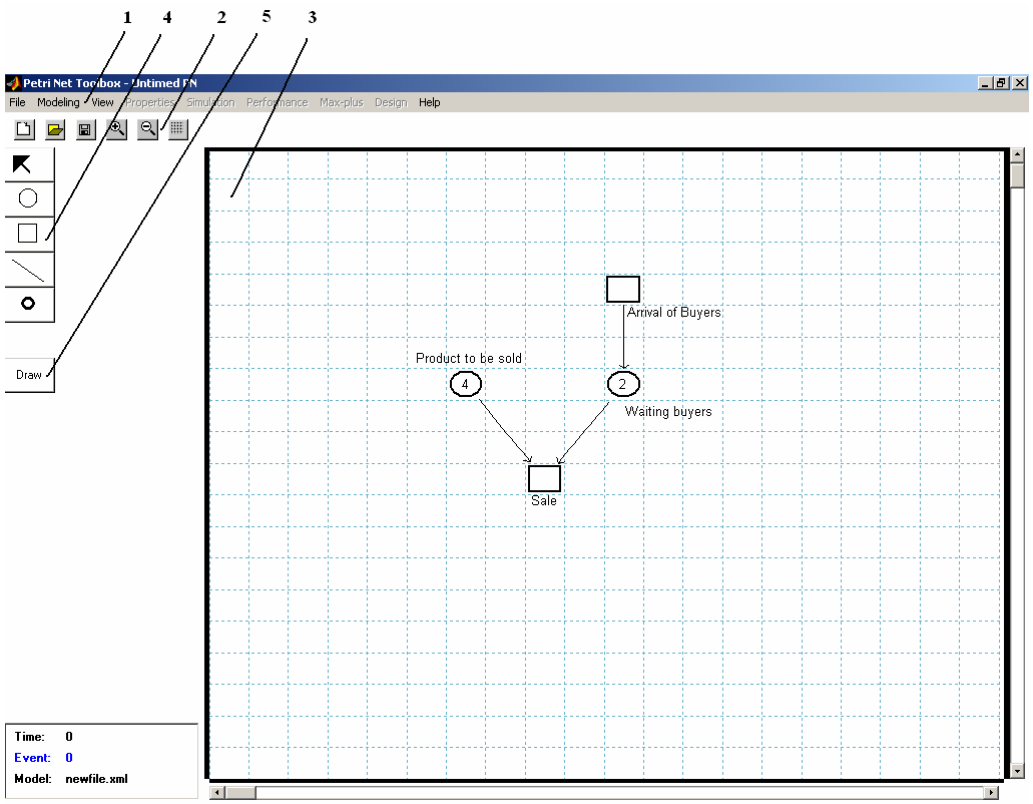
The GUI exhibits eight control panels (see fig. I.1): ***Menu Bar*** (1), ***Quick Access Toolbar*** (2), ***Drawing Area*** (3), ***Drawing Panel*** (4), ***Draw/Explore Switch*** (5), ***Simulation Panel*** (6), ***Status Panel*** (7) and a ***Message Box*** (8). Further on, all these panels are described.

I.2. Menu Bar

The ***Menu Bar*** (placed horizontally, on top of the main window of the ***PN Toolbox***) displays a set of nine drop-down menus, from which the user can access all the facilities available in the ***PN Toolbox***. These menus are enabled in accordance with the exploitation mode of the ***PN Toolbox***, namely the ***Draw Mode*** or the ***Explore Mode***.

The menus available under the ***Menu Bar*** are: ***File***, ***Modeling***, ***View***, ***Properties***, ***Simulation***, ***Performance***, ***Max-Plus***, ***Design*** and ***Help***.

(a)



(b)

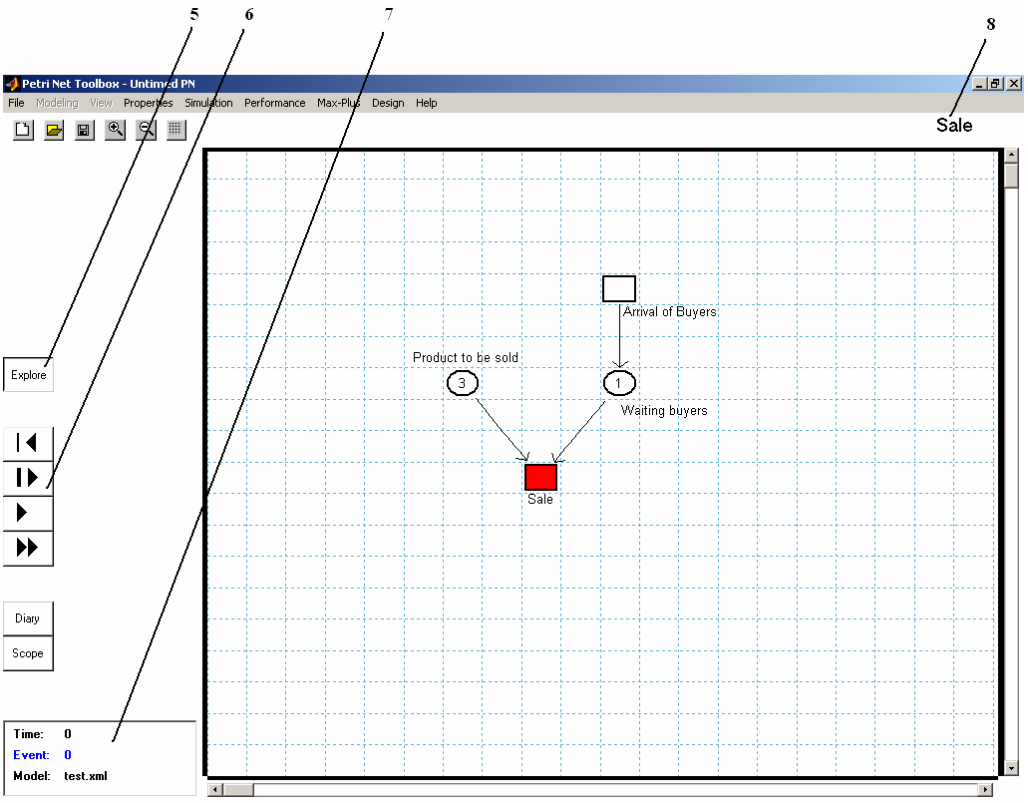


Fig. I.1. *The main window of the **PN Toolbox**:*
(a) *Draw Mode*, (b) *Explore Mode.*

I.2.1. File menu

The ***File*** menu (fig. I.2) offers facilities for file-handling operations. This is the only menu available when the ***PN Toolbox*** GUI is started. This menu contains the commands:

- ***New Model***: opens a new board in the ***Drawing Area*** for the user to build a new PN model. This command first opens a dialogue box for selecting the type of the new model;
- ***Open Model***: loads a previously saved model;
- ***Close Model***: clears the ***Drawing Area*** asking for saving the current model if changes appeared in it;
- ***Save Model***: saves the PN model drawn in the ***Drawing Area***;
- ***Save Model As …***: saves the current model with a name given by the user;
- ***Save Model As Image***: saves the current model as a jpg file in a location and with a name given by the user;
- ***Print Model***: prints the current model on the printer selected by the user;
- ***Exit PNToolbox***: closes the current working session and clears all the global variables corresponding to the ***PN Toolbox***.

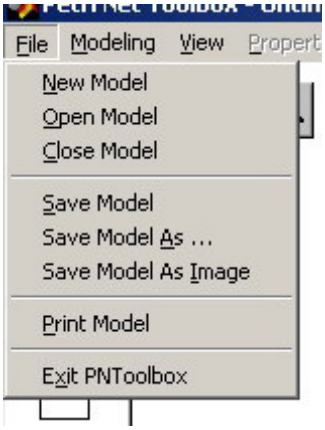


Fig. I.2. *The **File** menu.*

Fig. I.3. *The **Modeling** menu.*

I.2.2. Modeling menu

The ***Modeling*** menu (fig. I.3) provides tools for graphical editing (graph nodes, arcs, tokens, labels) a model in the ***Drawing Area***. The commands available in this menu are:

- ***Add Place***: adds a new place to the current PN model;
- ***Add Transition***: adds a new transition to the current PN model;
- ***Add Arc***: allows drawing an arc between two different nodes of the current PN model;
- ***Add Token***: adds a token in a specific place;
- ***Edit Objects***: opens the dialogue box associated with the net object that is selected in the ***Drawing Area*** by a click on the left button of the mouse. It allows the user to edit the properties of the selected object; these properties are in full accordance with the type of the PN model, so that part of them might be already disabled by the initial choice of the net type.

- ***Resolution for Conflicting Transitions***: allows assigning priorities or probabilities to conflicting transitions.

I.2.3. View menu

The ***View*** menu (fig. I.4) allows choosing specific conditions for visualization of the current model. The commands available in this menu are:

- ***Zoom In***: lets the user get a closer view to a smaller portion of the ***Drawing Area***;
- ***Zoom Out***: lets the user visualize a larger portion of the ***Drawing Area***;
- ***Show Grid:*** allows the user to add/remove grid lines to/from the ***Drawing Area***;
- ***Transitions Representation:*** allows the user to select the desired representation (as *Square*, *Horizontal Rectangle, Vertical Rectangle, Horizontal Bar* or *Vertical Bar*) for the transitions of a new model drawn in the ***Drawing Area***;
- ***Arc Weights:*** permits displaying or hiding the current values of arc weights.

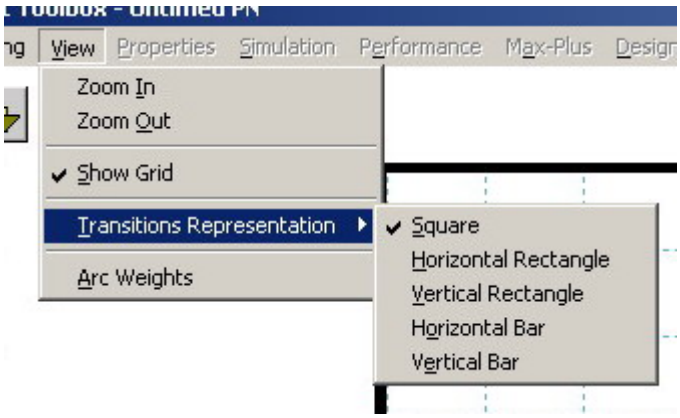


Fig. I.4. *The **View** menu*.

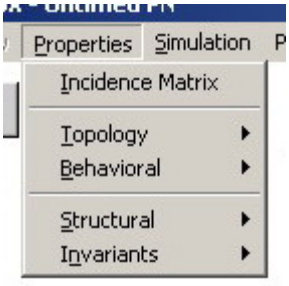


Fig. I.5. *The **Properties** menu*.

I.2.4. Properties menu

The ***Properties*** menu (fig. I.5) provides computational tools for the analysis of the behavioral and structural properties of the current PN model. The following commands are available in this menu:

- ***Incidence Matrix:*** displays the incidence matrix of the current model in a separate window;
- ***Topology:*** performs topology analysis of ordinary Petri nets and displays the *Net Class*, *Traps* and *Siphons*;
- ***Behavioral*:** constructs the coverability tree of the current model, which can be visualized in two different formats: graphic and text. The *Liveness* of an ordinary Petri net can also be investigated.
- ***Structural***: performs the analysis of the structural properties of the current model (structural boundedness, conservativeness, repetitiveness and consistency);
- ***Invariants***: provides minimal-support P- and T-invariants.

I.2.5. Simulation menu

Using the ***Simulation*** menu (fig. I.6), the user can control the simulation progress and record the results (details in section II.3). The following commands are available in this menu:

- ***Step***: simulates the PN model step by step, accompanied by animation;
- ***Run Slow***: simulates the PN model with a low speed, which is specified by the user (see the ***Preferences*** command), accompanied by animation. The simulation ends when the user clicks the right button of the mouse;
- ***Run Fast***: simulates the PN model with the maximum speed allowed by the environment. No animation is provided in this case. The simulation ends when the condition set by the user in ***Breakpoint*** is met;
- ***Breakpoint***: allows choosing the stop condition for simulation in the ***Run Fast*** mode;
- ***Reset***: brings the PN model to the initial state and resets all the results that were recorded in a previous simulation experiment;
- ***Log File***: stores the simulation history (recording all the states that the PN model passed through) in a file designated by the user. This command is operative only in the ***Step*** and ***Run Slow*** modes;
- ***View History***: opens the simulation log file;
- ***Preferences***: opens a dialogue box (fig. I.7) allowing the user to set the conditions of the simulation (***Animation Delay*** is valid only for the ***Run Slow*** simulation mode; ***Token-In-Place Color*** is meaningful only for P-timed PNs; the value of the ***Seed*** is used to initialize the random number generator of MATLAB).

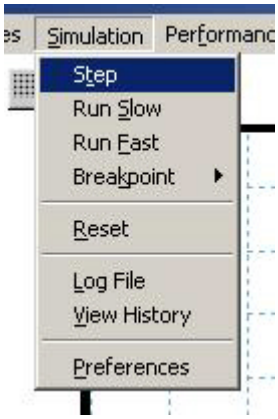


Fig. I.6. *The **Simulation** menu.*

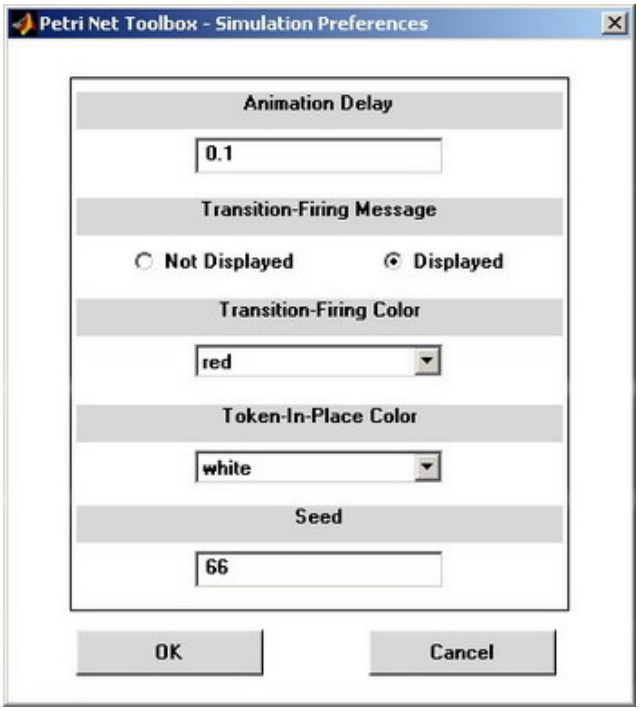


Fig. I.7. *The **Preferences** dialogue box.*

I.2.6. Performance menu

At the end of a simulation experiment, the ***Performance*** menu (fig. I.8) allows the visualization of the ***global performance indices*** (see section II.4 ***Analyzing Simulation Results***) that are written in an HTML file. These indices are separately recorded for transitions and for places. The commands available for this menu are:

- ***Place Indices***: displays the performance indices corresponding to the places;
- ***Transition Indices***: displays the performance indices corresponding to the transitions.
- ***Cycle Time***: provides the period of the steady state dynamics for a partially consistent deterministically timed PN, covered by *P-invariants*.

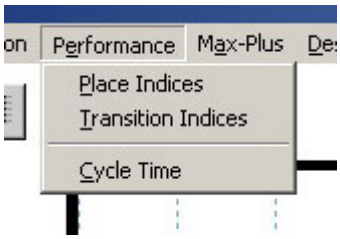


Fig. I.8. *The **Performance** menu*.

Fig. I.9. *The **Help** menu*.

I.2.7. Max-Plus menu

The ***Max-Plus*** menu (command) allows performing the simulation and analysis of an event graph (marked graph) based on its max-plus state-space model. A new MATLAB figure is opened and all the facilities available for max-plus analysis and simulation are accessible (see section II.5 **Max-Plus Models**).

I.2.8. Design menu

The ***Design*** menu is used for the synthesis of timed PN models; this allows simulations for several types of parameterizations considered in the PN architecture (see section II.6 ***Design***).

I.2.9. Help menu

The ***Help*** menu (fig. I.9) provides information about the exploitation of the ***PN Toolbox***. The commands available in this menu are as follows:

- ***Contents and Index***: opens the online help;
- ***Demos***: allows visualization of four **Flash** movies initiating the user in the exploitation of the ***PN Toolbox***;
- ***About Petri Net Toolbox***: contains information about the authors of the ***PN Toolbox***.

I.3. Quick Access Toolbar

The ***Quick Access Toolbar*** (fig. I.10) is placed as a horizontal bar just below the ***Menu Bar*** and presents six image buttons. The actions of the first three buttons are identical to those controllable by the ***New*** (1), ***Open*** (2) and ***Save*** (3) commands from the ***File*** menu. The actions of the next three buttons are identical to those controllable by the ***Zoom In*** (4), ***Zoom Out*** (5) and ***Show Grid*** (6) commands from the ***View*** menu.

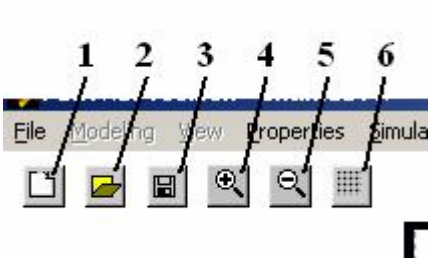


Fig. I.10. *The **Quick Access Toolbar.***

I.4. Drawing Area

The ***Drawing Area*** (located in the central and right side of the main window – see fig. I.1) is provided with a grid, where the nodes of the PN graph are to be placed, and with two scrollbars (on the right and bottom sides) for moving the desired parts of the graph into view.

The ***Drawing Area*** is an axes MATLAB object and it is organized as a matrix with 50 rows and 50 columns. The bottom left-hand corner corresponds to the (0,0) cell, whereas the upper right-hand corner to the (49,49) cell. In one cell the user can draw a single node (place or transition), so that the total number of places and transitions in a model cannot exceed 2500.

Using the ***Zoom In*** / ***Zoom Out*** commands from the ***View*** menu, the number of the cells is decreased/increased with 10 cells on each axis. The ***Show Grid*** command adds/removes the grid lines to/from the Drawing Area.

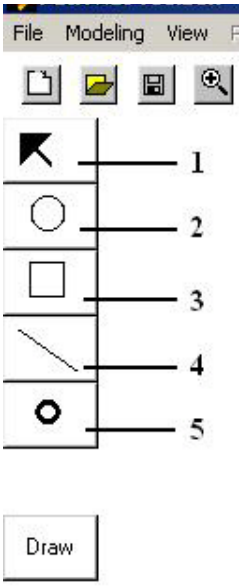


Fig. I.11. *The **Drawing Panel**.*

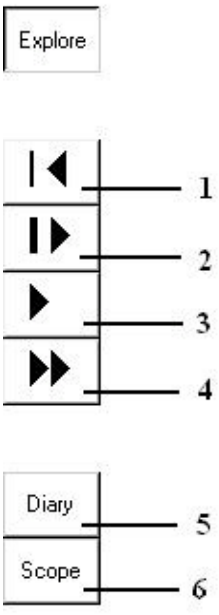


Fig. I.12. *The **Simulation Panel**.*

I.5. Drawing Panel

The ***Drawing Panel*** (fig. I.11) is placed vertically, in the left side of the main window, below the ***Quick Access Toolbar***, and presents five image buttons with actions identical to the commands ***Edit Objects*** (1), ***Add Place*** (2), ***Add Transition*** (3), ***Add Arc*** (4) and ***Add Token*** (5) available in the ***Modeling*** menu.

I.6. Draw/Explore Switch

The ***Draw / Explore Switch*** (identified as (5) in fig. I.1) allows switching between the ***Draw Mode*** (in which the user can draw a new model or modify an existing one – button unpressed) and the ***Explore Mode*** (in which the user can access all functions available for simulation and analysis – button pressed).

When switching from the ***Explore Mode*** to the ***Draw Mode***, the marking of the PN model is automatically reset to the initial token configuration and the former values of all the performance indices are lost.

I.7. Simulation Panel

The ***Simulation Panel*** (fig. I.12) is placed vertically, in the left side of the main window, just below the ***Drawing Panel***. It presents four image buttons with actions identical to the commands ***Reset*** (1), ***Step*** (2), ***Run Slow*** (3) and ***Run Fast*** (4) available in the ***Simulation*** menu. It also provides two visualization instruments for observing the progress of the simulation: ***Diary*** (5) and ***Scope*** (6) (see section **Running a Simulation**).

I.8. Status Panel

The ***Status Panel*** (identified as (7) in fig. I.1) is a message board (placed in the bottom left-hand corner of the main window), where the ***PN Toolbox*** displays the current simulation time and the total number of events. In addition, the ***Status Panel*** displays the file name corresponding to the current model.

I.9. Message Box

The ***Message Box*** (identified as (8) in fig. I.1) is a MATLAB text object used by the ***PN Toolbox*** to display messages to the user. In the ***Draw Mode***, five types of messages may be displayed, corresponding to the actions enabled by the five buttons in the ***Drawing Panel***, respectively. In the ***Explore Mode***, messages are displayed only during the simulation of a model (***Step*** or ***Run Slow*** commands) if option ***Displayed*** is checked for ***Transition-Firing Message*** in the ***Preferences*** dialogue box (from the ***Simulation*** menu). When a transition fires, the ***Message Box*** displays a text associated with this transition; the text must be previously defined in the ***Edit Transition*** dialogue box (***Draw Mode*** - see section II.1.3).

Part II.

Exploiting the Toolbox

II.1. Building a Model

II.1.1. Overview

The ***PN Toolbox*** provides a set of commands for building a PN model, which are accessible from the ***Modeling*** menu or from the corresponding buttons in the ***Drawing Panel***. The model may be easily drawn, in a natural fashion, in the ***Drawing Area***. The user can also have a read / write access to the properties of the net nodes and arcs by opening the dialog box associated with the object selected in the ***Drawing Area***. In addition, the ***PN Toolbox*** allows the assignment of priorities and / or probabilities to conflicting transitions.

The ***Drawing Area*** displays a grid with dotted gray lines. The user may hide or show these lines by using the command ***View / Show Grid*** or by pressing the corresponding button from the ***Quick Access Toolbar***. The nodes of the PN may be placed only in the grid cells; each cell can contain a single net node. This way, each node is uniquely characterized by its coordinates in the ***Drawing Area***.

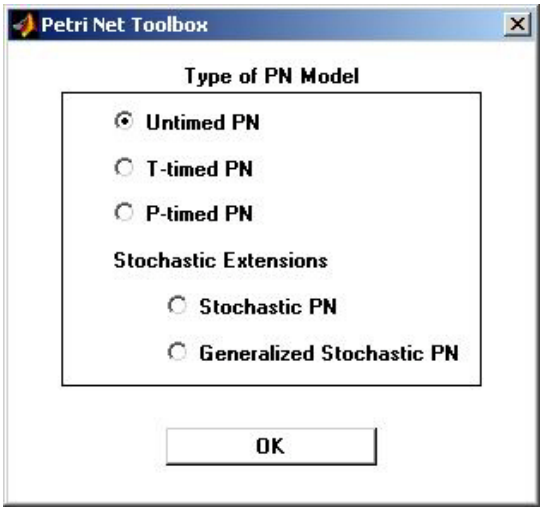


Fig. II.1. *The dialogue box for selecting the type of a PN model.*

A PN model created in the ***PN Toolbox*** is saved as an *XML* file. Besides the automatic generation of the XML file of a model, any text editor can be used to produce such a file if the user complies with the syntax given in Appendix 1.

The first step in the creation of a new model is to define its type. By selecting the type of the PN, some of the work variables are automatically changed. This is because the simulation and analysis procedures differ from one type to another. The drawing board of a new model may be opened in the ***Drawing Area*** by selecting the ***File / New Model*** command or by pressing the ***New*** button from ***Quick Access Toolbar***. This command opens a dialog box (fig. II.1) which permits the selection of the type of model to be built. The default type is ***Untimed PN***.

A previously created model may be loaded from the disk by means of the command ***Open*** available under the ***File*** menu or by pressing the ***Open*** button from the ***Quick Access Toolbar***. Changes in an existing model can be done directly in the GUI while in ***Draw Mode***, or by editing the corresponding **XML** file.

If a model is already loaded in the window, the ***Drawing Area*** is cleared by selecting the command ***File / Close Model***.

II.1.2. Places

A place is graphically represented in the ***Drawing Area*** by a circle. To draw a place in the ***Drawing Area***, the user must press the ***Add Place*** button from the ***Drawing Panel*** or select the ***Add Place*** command from the ***Modeling*** menu. Then, the user must click only once into the desired grid cell of the ***Drawing Area***. A second click in the same grid cell has no effect.
Once the circle corresponding to a place is drawn, a label is automatically attached to it. The default position of the label is below the circle.

After drawing a place, there are two ways to change its position in the ***Drawing Area*** while in ***Draw Mode*** and no button from the ***Drawing Panel*** is pressed. The first way is to left-click the desired place and then drag it in the new position. The second way is to right-click the place; as a result, a MATLAB uicontext menu (fig. II.2) appears and the command ***Move Place*** becomes available. In both cases, the label is moved together with the place.

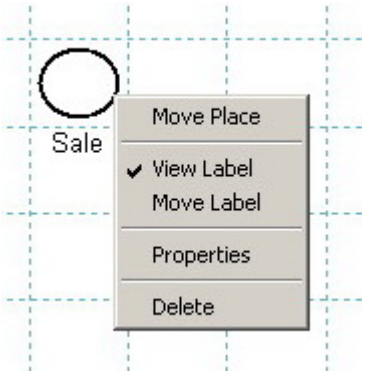


Fig. II.2. *The uicontext menu of a place*

For a selected place, the uicontext menu also allows: (*i*) controlling the label's visibility on the screen (***View Label*** command), (*ii*) changing the position of the label (***Move Label*** command), (*iii*) deleting the place (***Delete*** command) and (*iv*) opening the ***Edit Place*** dialogue box (fig. II.3) that lets the user modify the properties of the place as a MATLAB object (***Properties*** command).

The ***Edit Place*** dialogue box may be also opened by one of the procedures (EP1) or (EP2) described below, followed by a click on the desired place. (EP1) consists in selecting the ***Edit Objects*** command from the ***Modeling*** menu; (EP2) consists in pressing the ***Edit Objects*** button from the ***Drawing Panel***.

Each place is uniquely identified with an **id** that is automatically assigned by the ***PN Toolbox*** and cannot be changed by the user. This id appears in the title bar of the ***Edit Place*** dialogue box.

The option ***Label*** displays the string that is used as the label of the place. By default, this string coincides with the id of the place. The user can modify this string (without affecting the id) if necessary.

The option ***Color*** displays the color used for drawing the place. By default, this color is black, but the user can select another one from a list of eight predefined colors. Using different colors for drawing the places of a net might be helpful for complex topologies (e.g. for highlighting different phases requested by a multi-step design procedure).

The option ***Capacity*** displays the capacity of the place. By default, the capacity is **Inf** (the IEEE arithmetic representation for positive infinity). The user can set this field to a positive integer value.

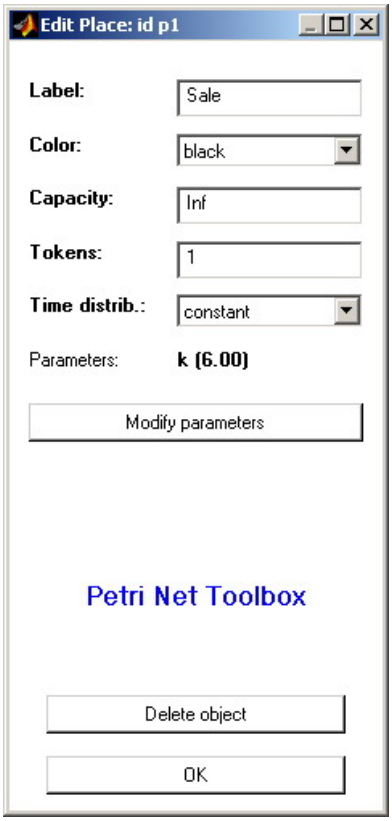


Fig. II.3. *The **Edit Place** dialogue box for modifying the properties of a place.*

The option ***Tokens*** displays the number of tokens in the place. By default, this number is 0. The user can set this field to a positive integer value. If the marking of a place is greater than 0, this marking is shown as a number inside the circle corresponding to that position. Void marking is not explicitly shown.

There are two more possibilities to add a token to a place (AT1) or (AT2) described below, followed by a click on the desired place. (AT1) consists in selecting the ***Add Token*** command from the ***Modeling*** menu; (AT2) consists in pressing the ***Add Token*** button from the ***Drawing Panel***.

In case of place-timed PN models (see section II.3.4), the ***Time distribution*** option associated with a place allows the user to specify the probability distribution and the necessary parameter(s) which define the corresponding time-duration. This option is not available for untimed or transition-timed models.

The ***Delete object*** button placed at the bottom of the ***Edit Place*** dialogue box lets the user delete the place from the model.

II.1.3. Transitions

A transition is graphically represented in the ***Drawing Area*** by a square. To draw a transition in the ***Drawing Area***, the user must press the ***Add Transition*** button from the ***Drawing Panel*** or select the ***Add Transition*** command from the ***Modeling*** menu. Then, the user must click only once into the desired grid cell of the ***Drawing Area***. A second click in the same grid cell has no effect.

Once the square corresponding to a transition is drawn, a label is automatically attached to it. The default position of the label is below the square.

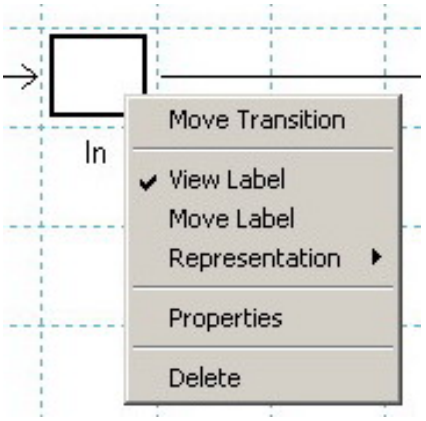


Fig. II.4. *The uicontext menu of a transition.*

The same as for a place, after drawing a transition, there are two ways to change its position in the ***Drawing Area*** while in ***Draw Mode*** and no button from the ***Drawing Panel*** is pressed. The first way is to left-click the desired transition and then drag it in the new position. The second way is to right-click the transition; as a result, a MATLAB uicontext menu (fig. II.4) appears and the command ***Move Transition*** becomes available. In both cases, the label is moved together with the transition.

For a selected transition, the uicontext menu also allows: (*i*) controlling the label's visibility on the screen (***View Label*** command), (*ii*) changing the position of the label (***Move Label*** command), (*iii*) deleting the transition (***Delete*** command) and (*iv*) opening the ***Edit Transition*** dialogue box (fig. II.5) that lets the user modify the properties of the transition as a MATLAB object (***Properties*** command).

The ***Edit Transition*** dialogue box can be also opened by one of the procedures (ET1) or (ET2) described below, followed by a click on the desired transition. (ET1) consists in selecting the ***Edit Objects*** command from the ***Modeling*** menu; (ET2) consists in pressing the ***Edit Objects*** button from the ***Drawing Panel***.

Each transition is uniquely identified with an **id** that is automatically assigned by the ***PN Toolbox*** and cannot be changed by the user. This id appears in the title bar of the ***Edit Transition*** dialogue box.

The option ***Name*** displays the string that is used as the label of the transition. By default, this string coincides with the id of the transition. The user can modify this string (without affecting the id) if necessary.

The ***Message*** text-box contains the string that is displayed in the ***Message Box*** during the simulation of the model when firing that transition if option ***Displayed*** is checked for ***Transition-Firing Message*** in the ***Preferences*** dialogue box. By default, this string is "*Firing transition x*" where *x* is the id of that transition. The user can modify this string if necessary.

The option ***Color*** displays the color used for drawing the transition. By default, this color is black, but the user can select another one from a list of eight predefined colors. Using different colors for the transitions of a net might be helpful for complex topologies (e.g. for highlighting different phases requested by a multi-step design procedure).

In case of transition-timed PN models (see section II.3.3), the ***Distribution*** option associated with a transition allows the user to specify the probability distribution and the necessary parameter(s) which define the corresponding time-duration. This option is not available for untimed or place-timed models.

For Stochastic PN models, only exponentially distributed firing rates can be assigned to the transitions in the net. In the case of Generalized Stochastic PN models, only constant and exponentially distributed firing rates can be assigned to the transitions in the net; obviously, an instantaneously fireable transition can be modeled by setting the constant distribution to 0. For these two types of models, the user has the possibility to enable or disable the dependence between the firing rate of a transition and the marking of its input places by checking the ***Marking dependent*** option.

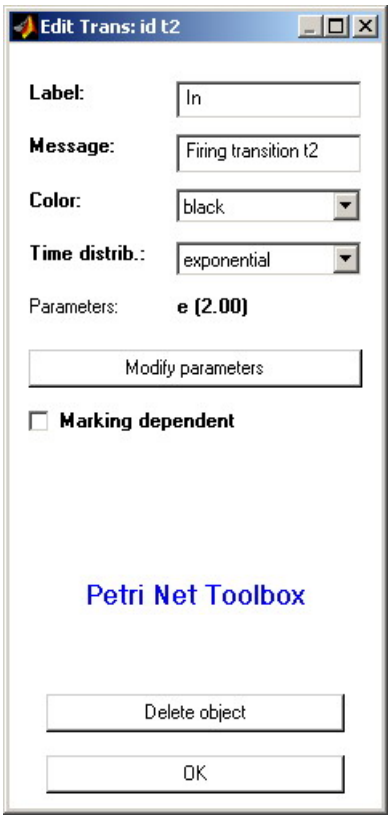


Fig. II.5. *The **Edit Transition** dialogue box for modifying the properties of a transition*

The ***Delete object*** button placed at the bottom of the ***Edit Place*** dialogue box lets the user delete the transition from the model.

II.1.4. Arcs

To draw an arc in the ***Drawing Area***, the user must press the ***Add Arc*** button from the ***Drawing Panel*** or select the ***Add Arc*** command from the ***Modeling*** menu. Then, the user must click on the start node and then on the end node. The implementation in the ***PN Toolbox*** of PN models complies with the basic rule: *an arc of a PN can only connect a place to a transition* (pre-arc) *or a transition to a place* (post-arc)*, but never two nodes of the same kind.* The role of splitting arcs in two categories (pre- and post-arcs) will become apparent below, when talking about inhibitor arcs.

By default, an arc is represented as a straight arrow between the two selected nodes of the net. While in ***Draw Mode*** and no button from the ***Drawing Panel*** is pressed, a right-click on an arc of the net opens a MATLAB uicontext menu (fig. II.6) that allows: (*i*) modifying the graphical

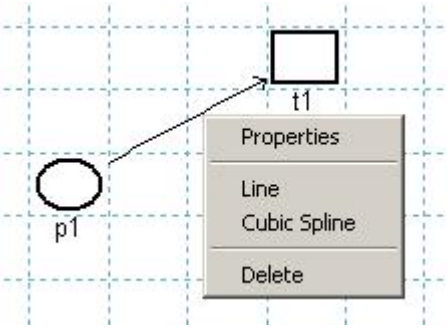


Fig. II.6. *The uicontext menu of an arc.*

representation in the ***Drawing Area*** (***Line*** command for a straight line or ***Cubic Spline*** command for a curve), (*ii*) deleting the arc (***Delete*** command) and (*iii*) opening the ***Edit Arc*** dialogue box (fig. II.7) that lets the user modify the properties of the arc as a MATLAB object (***Properties*** command).

The ***Edit Arc*** dialogue box can be also opened by one of the procedures (EA1) or (EA2) described below, followed by a click on the desired arc. (EA1) consists in selecting the ***Edit Objects*** command from the ***Modeling*** menu; (EA2) consists in pressing the ***Edit Objects*** button from the ***Drawing Panel***.

Fig. II.7. *The **Edit Arc** dialogue box for modifying the properties of an arc.*

Each arc is uniquely identified with an id that is automatically assigned by the ***PN Toolbox*** and cannot be changed by the user. This id appears in the title bar of the ***Edit Arc*** dialogue box.

The option ***Color*** displays the color used for drawing the arc. By default, this color is black, but the user can select another one from a list of eight predefined colors. Using different colors for the arcs of a net might be helpful for complex topologies (e.g. for highlighting different phases requested by a multi-step design procedure).

The option ***Type*** displays the type of an arc. By default, the type of an arc of a PN model is regular, but the user can change it into bidirectional or inhibitor, if necessary. A *regular* arc represents the standard connection between two nodes of different types. A *bidirectional* arc is equivalent to a pair of arcs with the same weight, one connecting a place to a transition and the other one connecting the same transition to the same place. The graphical representation of a bidirectional arc is a line with arrows at both ends.

An *inhibitor* arc can connect only a place to a transition. The transition is enabled only if the number of tokens in the input place is strictly smaller than the weight of the inhibitor arc. The graphical representation of an inhibitor arc is a line between the two nodes ending with a small circle (near the inhibited transition).

The option ***Weight*** displays the weight (multiplicity) of the arc. By default, the weight of an arc is equal to 1, but the user can set this field to a positive integer value.

By default, the weights of the arcs in a net are not shown in the ***Drawing Area***. At any stage of PN drawing, the user can visualize the current values of the weights for all the arcs in the net by selecting the ***Arc Weights*** command from the ***View*** menu. This command opens the dialogue box presented in fig. II.8 and the user must click the ***Yes*** button.

Fig. II.8. *The **View Arc Weights** dialogue box.*

II.1.5. Setting Priorities and / or Probabilities for Conflicting Transitions

By default, the ***PN Toolbox*** sets equal probabilities and / or priorities to conflicting transitions. These probabilities and / or priorities are used by the ***PN Toolbox***, when simulating a model, for selecting, from a set of conflicting transitions enabled at the same time, the next one to be fired. By selecting the ***Resolution for Conflicting Transitions*** option available under the ***Modeling*** menu, the corresponding dialogue window is opened (fig. II.9) and the user can add, delete or edit the probabilities and / or priorities for each group of conflicting transitions.

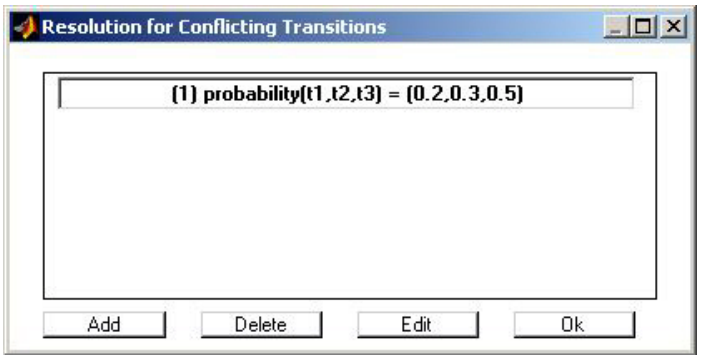


Fig. II.9. *The **Resolution for Conflicting Transitions** dialogue window.*

II.2. Exploring Model Properties

II.2.1. Overview

The ***PN Toolbox*** offers several possibilities for analyzing discrete event systems modeled with PNs that are briefly presented in the sequel.

II.2.2. Incidence Matrix

The incidence matrix of a PN is computed directly from the graphical model and can be visualized in a MATLAB window opened by selecting the ***Incidence Matrix*** command from the ***Properties*** menu. This command is available for all types of PNs.

II.2.3. Behavioral Properties

For untimed PN models, the behavioral properties (e.g. boundedness, liveness, reversibility, etc.) may be studied based on the coverability tree of the net. Relying on the topology and the initial marking of the net, the ***PN Toolbox*** can automatically construct this tree and display it in either text or graphical mode. The coverability tree is built with or without the ω-convention (fig. II.10). The ω-convention means the usage of a generic symbol (herein denoted by "ω") for referring to unbounded markings (Murata, 1989). Since the boundedness of a net cannot be a priori known, it is recommended to start the construction of the coverability tree by answering ***No*** to the question ***Do you want to use the "ω - convention"?***.

Fig II.10. *The **Coverability Tree** dialogue window.*

II.2.4. Structural Properties

The *structural properties* of a PN are independent of the initial marking and may be characterized in terms of linear inequalities based on the incidence matrix of the net. The structural properties can be investigated from the ***Properties / Structural Properties*** menu, by means of the ***Structural Boundedness***, ***Conservativeness***, ***Repetitiveness*** and ***Consistency*** options.

II.2.5. Invariants

From the ***Properties / Invariants*** menu, the minimal-support *P*- and *T-invariants* (Martinez and Silva, 1982) can be visualized as vectors displayed in separate windows, automatically opened by the ***PN Toolbox***. New invariants may be constructed as linear combinations of these vectors.

II.3. Running a Simulation

II.3.1. Overview

After a model was drawn or retrieved from the disk, the user can simulate it. Three modes of simulation are implemented in the ***PN Toolbox***, namely: ***Step***, ***Run Slow*** and ***Run Fast***, that are called from the ***Simulation*** menu or from the ***Simulation Panel***.

The ***Step*** simulation mode ensures the progress of the simulation by a single step, when the status of a transition is changed from enabled to fired, or from fired to disabled. With the occurrence of each event two successive steps are associated, meaning the complete treatment of the firing transition.

The ***Step*** and ***Run Slow*** simulation modes are accompanied by animation. The animation displays the current number of tokens in the places of the net and colors transitions when they are fired. In the ***Run Slow*** mode, the speed of animation is adjustable from the ***Animation Delay*** option in the ***Simulation / Preferences*** dialogue box. The simulation can be stopped with a right-click.

In these two modes, the ***Status Panel*** (placed in the bottom left-hand corner of the main window) presents the number of events and the current time.

Also, the user can record the progress of the simulation by using the ***Log File*** command from the ***Simulation*** menu. This journal is automatically saved in the project directory as an *HTML* file and can be read by means of any Internet browser using the ***Simulation / View history*** command available in the ***PN Toolbox***.

Note that in ***Step*** or ***Run Slow*** modes, the ***Scope*** and ***Diary*** facilities are available during the simulation. Their usage is relevant when studying timed PN.

The ***Run Fast*** simulation mode is not accompanied by animation and the simulation speed depends on the computer performances. This mode is recommended when the user is interested in estimating global performance indices of the PN model over a large time horizon (or, equivalently, for the occurrence of a large number of events).

From the ***Simulation / Breakpoint*** submenu the user can control the end point of the simulation in the ***Run Fast*** mode. The simulation is stopped by default when 1000 events have occurred or if no transition can fire any longer (i.e. when deadlock appears).

In all simulation modes, immediately after the end of the simulation, the values of the Performance Indices which characterize the simulated dynamics are available on request, by accessing the menu ***Performance***.

The simulation principle is different from one type of PN to another. Details about the simulation method implemented in the ***PN Toolbox*** for each type of PN model are given in the next pages.

II.3.2. Untimed Petri Nets

The sequencing of the events is reduced to simply ordering their occurrences. The simulation runs by firing the transitions one-by-one, in accordance with the *transition firing rule*. The ***PN Toolbox*** stores all the enabled transitions in a MATLAB array but only one transition fires at a time. The function that returns the next transition to be fired makes the decision according to the priorities or probabilities assigned to conflicting transitions from the ***Modeling / Conflicting Transitions*** command.

For the simulation modes ***Step*** and ***Run Slow***, the ***Events*** counter in the ***Status Panel*** is indexing and the currently firing transition gets the red color (by default) or an arbitrary color (selected by the user from the option ***Simulation / Preferences***). After it is fired, a transition returns to the background color.

The usage of the ***Scope*** and ***Diary*** facilities is not relevant for this type of PN.

II.3.3. T-timed Petri Nets

For *transition-timed* PN (*T*-timed PN), time durations can be assigned to the transitions; tokens are meant to spend that time as reserved in the input places of the corresponding transitions. In simulation, all the transitions that can fire due to the current marking are fired at the same time. For conflicting transitions, priorities or probabilities allow the choice of the transition(s) to fire. A transition can fire several times, in accordance with the marking of its input places and, from a theoretical point of view, an infinitesimal delay is considered to separate any two successive firings. After a transition fires, the enabling condition is tested for all the transitions of the net.

For the time durations assigned to the transitions, appropriate functions can be used to generate random sequences corresponding to probability distributions with positive support. When a transition is waiting to fire (i.e. during the period when its input places contain reserved tokens) the reserved tokens are graphically removed from the input places, but the

computation procedures of the performance indices consider them as remaining in those places (in full accordance with the theoretic approach).

The ***Diary*** facility opens a new window that displays (dynamically) the instant(s) when the next fireable transition(s) is (are) to fire (see section II.3.7).

The ***Scope*** facility opens a new figure window that displays (dynamically) the evolution of the selected performance indices. The final value of the global index shown by the Scope is identical to that displayed on request, at the end of simulation, when the user explores the ***Performance*** menu (see section I.2.6).

II.3.4. P-timed Petri Nets

For *place-timed* PNs (*P*-timed PN), time durations can be assigned to the places; tokens are meant to spend that time as reserved in the corresponding places, immediately after their arrival. In simulation, all the transitions that can fire due to the current marking, fire at the same time. A transition can fire several times, in accordance with the marking of its input places and, from a theoretical point of view, an infinitesimal delay is considered to separate any two successive firings.

For the time durations assigned to the places, appropriate functions can be used to generate random sequences corresponding to probability distributions with positive support. During the simulation, the places that contain reserved tokens get the white color (default) or a color selected by using the option ***Simulation / Preferences***.

The ***Diary*** facility opens a new window that displays (dynamically) the instant(s) when the next fireable transition(s) is (are) to fire (see section II.3.7).

The ***Scope*** facility opens a new figure window that displays (dynamically) the evolution of the selected performance indices. The final value of the global index shown by the Scope is identical to that displayed on request, at the end of simulation, when the user explores the ***Performance*** menu (see section I.2.6).

II.3.5. Stochastic Petri Nets

For *stochastic* PNs (SPNs), only exponential type distributions can be used to assign the time durations of the transitions. For conflicting transitions, it is the shortest time duration that allows the choice of the transition to fire, without using priorities or probabilities. Multiple firing of the same transition is not permitted, even if the token content of its input places allows this; i.e. the transition fires once and after the allocated time elapses, it will fire again if the current marking is appropriate. (Notice that the mechanism for selecting the firing transition and the condition for firing only once make the difference from the *T-timed PNs* with exponential type distributions for the time durations).

The sequencing of the firing transitions is exclusively controlled by the time durations of exponential type, which ensures the equivalence with the Markov chains. The firing rate of the transitions (i.e. the inverse of the mean time-duration) is, by default, marking dependent, but the user can select a marking-independent operation.

The ***Diary*** facility opens a new window that displays (dynamically) the instant(s) when the next fireable transition(s) is (are) to fire.

The ***Scope*** facility opens a new figure window that displays (dynamically) the evolution of the selected performance indices. The final value of the global index shown by the Scope is identical to that displayed on request, at the end of simulation, when the user explores the ***Performance*** menu (see section I.2.6).

II.3.6. Generalized Stochastic Petri Nets

Generalized Stochastic PNs (GSPNs) have two different classes of transitions: *immediate* transitions and *timed* transitions. Once enabled, immediate transitions fire in zero time. Timed transitions fire after a random, exponentially distributed enabling time as in the case of SPNs. For *timed* transitions, the firing rate (i.e. the inverse of the mean time-duration) is, by default, marking dependent, but the user can select a marking-independent operation (the same way as for SPNs).

The simulation procedure is similar to the SPN case, the only difference occurring in the case of the *immediate* transitions that fire first; priorities / probabilities can be associated to these transitions, in order to resolve the conflicts.

The ***Diary*** facility opens a new window that displays (dynamically) the instant(s) when the next fireable transition(s) is (are) to fire.

The ***Scope*** facility opens a new figure window that displays (dynamically) the evolution of the selected performance indices. The final value of the global index shown by the Scope is identical to that displayed on request, at the end of simulation, when the user explores the ***Performance*** menu (see section I.2.6).

II.3.7. Diary

When active, this facility opens a new MATLAB window (fig. II.11) that displays (dynamically - during the simulation) the instant(s) when the next fireable transition(s) is (are) to fire. For several fireable transitions, the displayed list of transitions is ordered by the instants when the firings will occur.

The ***Diary*** facility is available only in the ***Step*** and ***Run Slow*** simulation modes. The usage of this facility is recommended for timed PN models and it gives no relevant information in the case of untimed models.

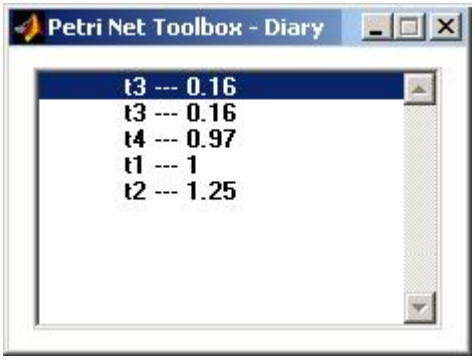


Fig II.11. *The **Diary** window.*

II.3.8. Scope

The ***Scope*** facility opens a new MATLAB window (fig. II.12) that displays (dynamically) the evolution of the selected performance index. Each index provides two types of information, namely a current value (that characterizes the PN at the current moment of the simulation) and a global value (that characterizes the whole evolution of the PN, as an average over the whole

time-horizon from the beginning of the simulation till the current moment). For the selected performance index, both current and global values are plotted versus time.

The usage of the ***Scope*** facility is available only for timed PNs in the ***Step*** and ***Run Slow*** simulation modes.

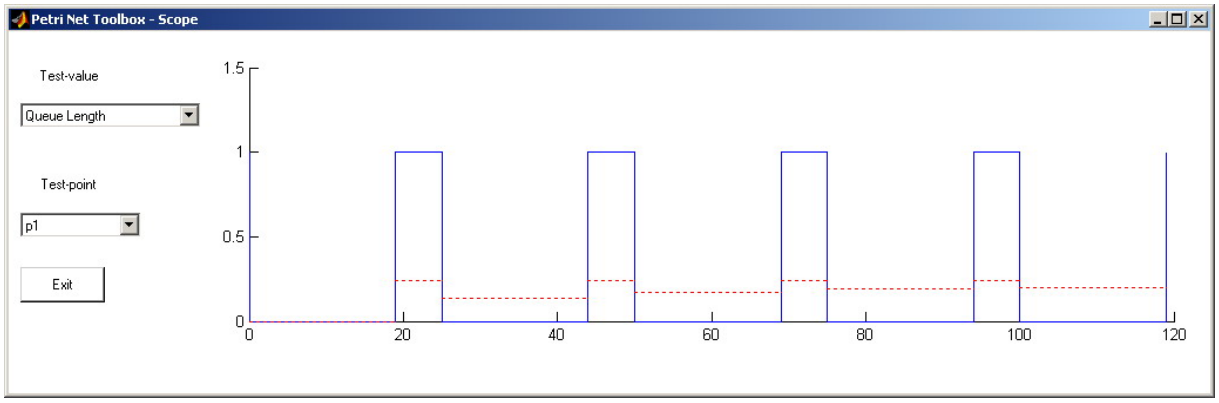


Fig II.12. *The **Scope** window.*

The current / global performance indices displayed by ***Scope*** are:

- for a transition:
  - ***Service Distance***: means the duration between two successive firings of the selected transition;
  - ***Service Time***: means the duration between the enabling and the firing of the selected transition;
  - ***Utilization***: means the current status of the selected transition: 1 for the time-interval between the enabling and the firing of the selected transition; 0 otherwise;
- for a place:
  - ***Arrival Distance***: means the duration between two successive instants when tokens arrive in the selected place;
  - ***Throughput Distance***: means the duration between two successive instants when tokens leave the selected place;
  - ***Queue Length***: means the number of tokens in the selected place.

During a simulation experiment, only one of the six performance indices can be dynamically displayed by the ***Scope***. The red color is automatically set for plotting the evolution of the global performance index. The blue color is automatically set for plotting the evolution of the current performance index. The final value shown by the ***Scope*** for the global performance index is identical to the value displayed on request, at the end of simulation, by accessing the ***Performance*** menu.

II.4. Analyzing Simulation Results

After simulation ends, the global performance indices described in the ***Scope*** section are stored by the ***PN Toolbox*** and can be visualized by using the ***Performance*** menu. Besides these, there are also a number of global indices for which the current values are not defined.

The following two tables present the complete lists of global indices associated with the places (displayed by the ***Place Indices*** command) and the transitions (displayed by the ***Transition Indices*** command), respectively:

- for a transition:
  - ***Service Sum***: total number of firings;
  - ***Service Distance***: average value of the current index ***Service Distance***;
  - ***Service Rate***: average frequency of firings (inverse of ***Service Distance***);
  - ***Service Time***: average value of the current index ***Service Time***;
  - ***Utilization***: average value of the current index ***Utilization***;
- for a place:
  - ***Arrival Sum***: total number of arrived tokens;
  - ***Arrival Distance***: average value of the current index ***Arrival Distance***;
  - ***Arrival Rate***: average frequency of token-arrivals (inverse of ***Arrival Distance***);
  - ***Throughput Sum***: total number of departed tokens;
  - ***Throughput Distance***: average value of the current index ***Throughput Distance***;
  - ***Throughput Rate***: average frequency of token-departures (inverse of ***Throughput Distance***);
  - ***Waiting Time***: average waiting time per token;
  - ***Queue Length***: average value of the current index ***Queue Length***.

These indices may be saved in *HTML* format in a file placed in the working directory.

II.5. Max-Plus Models

For place-timed event graphs, the ***PN Toolbox*** is able to directly derive the max-plus state-space representation from the topology and initial marking of a marked graph, in an implicit form:

$$\begin{aligned} \boldsymbol{x}(k) &= \bigoplus_{i=0}^{M} \left[\boldsymbol{A}_i \otimes \boldsymbol{x}(k-i) \oplus \boldsymbol{B}_i \otimes \boldsymbol{u}(k-i) \right], \\ \boldsymbol{y}(k) &= \bigoplus_{i=0}^{M} \left[\boldsymbol{C}_i \otimes \boldsymbol{x}(k-i) \oplus \boldsymbol{D}_i \otimes \boldsymbol{u}(k-i) \right], \end{aligned} \quad , \quad k = \overline{1, N},$$

where M denotes the maximal number of tokens in the initial marking and N stands for the number of simulated iterations. The components of the input vector $\boldsymbol{u}(k) = \left[u_1(k)\ u_2(k)\ \ldots\ u_m(k) \right]^{\mathrm{T}}$ and those of the output vector $\boldsymbol{y}(k) = \left[y_1(k)\ y_2(k)\ \ldots\ y_p(k) \right]^{\mathrm{T}}$ represent the k-th firing moments of the m source transitions and of the p sink transitions, respectively. In a similar manner, the state vector $\boldsymbol{x}(k) = \left[x_1(k)\ x_2(k)\ \ldots\ x_n(k) \right]^{\mathrm{T}}$ corresponds

to the *n* transitions in the net that have both input and output places (i.e. those transitions which are neither sources nor sinks).

The max-plus model is automatically constructed by the ***PN Toolbox*** by using the ***Max-Plus*** menu. This command opens the ***Max-Plus*** window (denoted by (1) in fig. II.13) providing access to the following features available for max-plus analysis:

- ***Equations*** button: displays the max-plus equations in a separate window (denoted by (2) in fig. II.13);
- ***Input*** button: allows the user to set the values of the input vectors (time instants). This can be done (*i*) by introducing the time instants as a matrix with *m* rows and *N* columns directly in the *dialogue box*, or (*ii*) by creating the matrix ***input_time*** with the structure described above in MATLAB's *Workspace*;
- ***Plot***: allows the user to select the components of the input, state or output vectors that will be plotted, by using vectors with components equal to 0 (enabled) or 1 (disabled);
- ***Iteration***: runs an iteration of the simulation at a time. The values of the state and output vectors are displayed in the ***Numerical Results*** box located on the left side of the ***Max-Plus*** window. The selected components are plotted;
- ***Reset***: resets the model to the initial state and clears the MATLAB axes;
- ***Exit***: closes the ***Max-Plus*** window and returns the control to the ***PN Toolbox*** GUI.

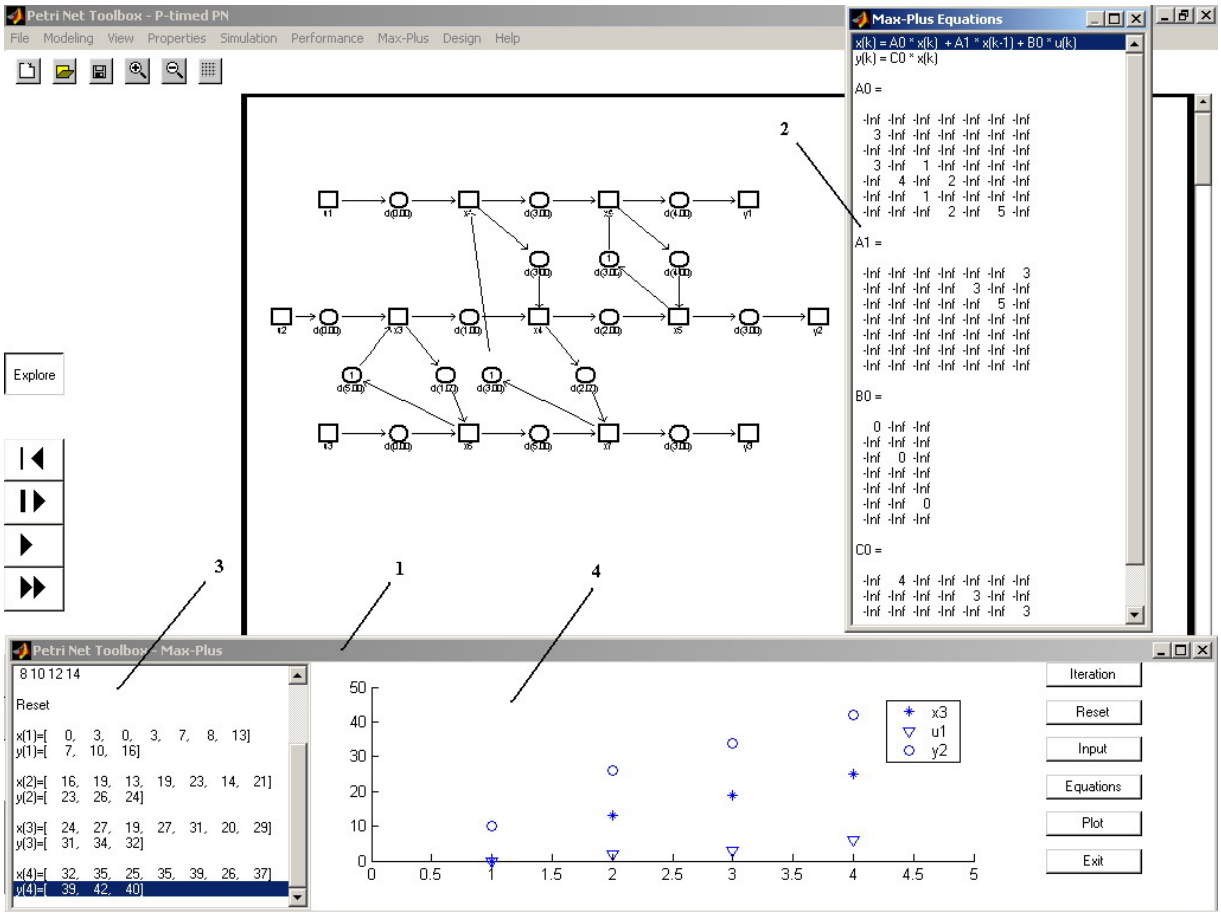


Fig. II.13. *Facilities available in the **PN Toolbox** for max-plus analysis.*

The ***Numerical Results*** box (located in the left side of the ***Max-Plus*** window, denoted by (3) in fig. II.13) displays the input vectors as well as information reflecting the stage of the analysis (the state and output vectors after each iteration).

In the MATLAB axes (denoted by (4) in fig. II.13) placed in the middle of the ***Max-Plus*** window, all the selected components are plotted. The components of the input, state and output vectors are plotted with the symbols "∇", "∗" and "○", respectively. For two or more components of the same vector, different colors are used with the same symbol.

II.6. Design

A facility for the synthesis of timed or (generalized) stochastic PN models is ***Design***, which allows exploring the dependence of a ***Design Index* (*I*)** on one or two ***Design Parameters*** that vary within intervals defined by the user.

A ***Design Parameter*** may be selected as (*i*) the initial marking of a place, (*ii*) a parameter of the distribution function defining the duration associated with a place or a transition in timed PNs, (*iii*) the mean value of the exponential distribution function associated with a transition in (generalized) stochastic PNs. The ***Design Parameters*** are generically denoted by ***x*** and ***y***; any other notation is not accepted. To ensure the correspondence between the symbol ***x*** (or ***y***) and the selected ***Design Parameter***, this symbol must be used as a numerical value when filling out the appropriate dialogue box (exactly as detailed in sections II.1.2 and II.1.3). The place or transition subject to parameterization is automatically colored in red.

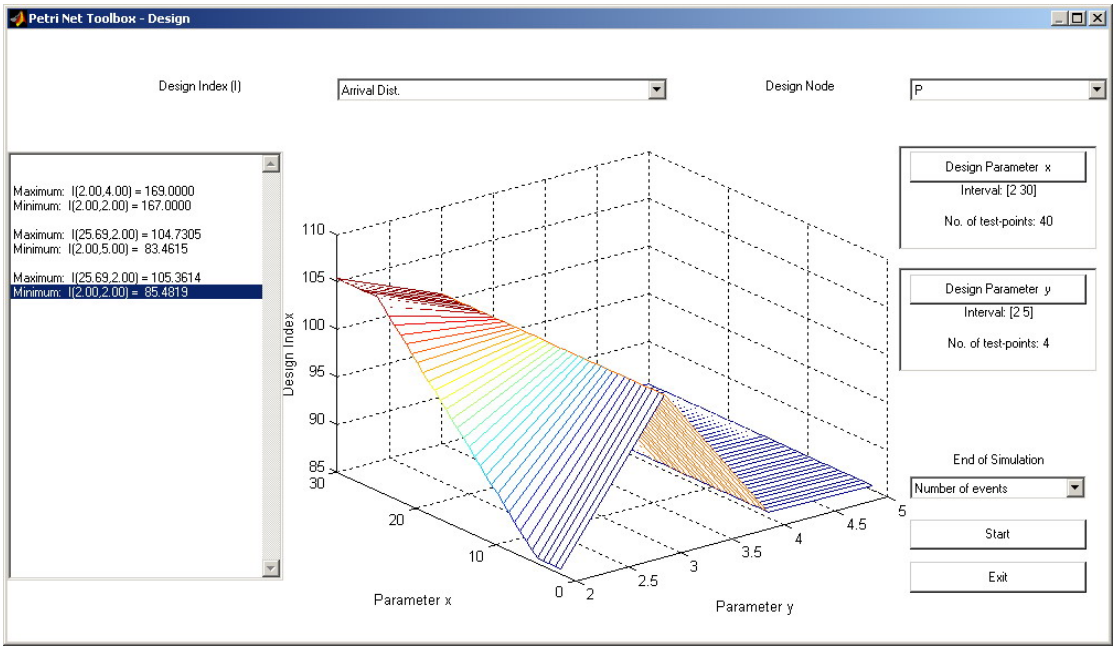


Fig. II.14. *The window opened after selecting the **Design** option of the **PN Toolbox**.*

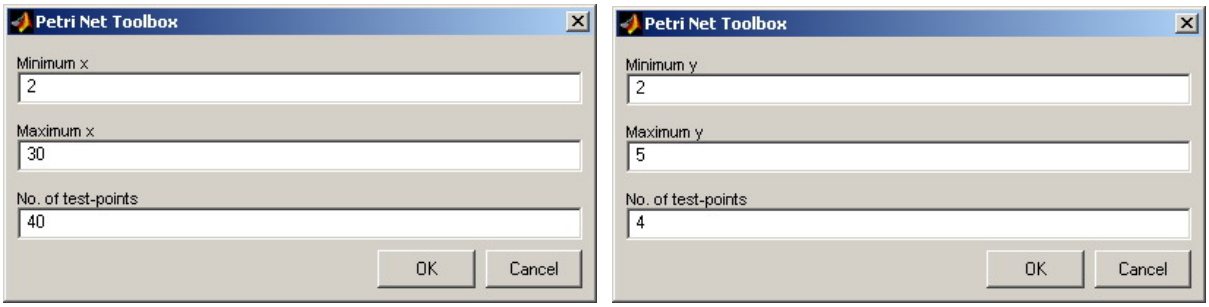
The ***Design Index*** may be selected as a global performance index associated with a ***Design Node***, namely ***Service Rate***, ***Service Distance***, ***Service Time*** or ***Utilization*** for a

transition, or ***Arrival Rate***, ***Arrival Distance***, ***Throughput Rate***, ***Throughput Distance***, ***Waiting Time*** or ***Queue Length*** for a place – see section II.4. The ***Design Index*** and the ***Design Parameter*** do not necessarily refer to the same node.

When selecting the ***Design*** command from the ***Menu Bar*** of the ***PN Toolbox***, a new window is opened (see fig. II.14). The choice of the ***Design Index*** and ***Design Node*** is made from the corresponding combo-boxes.

By pushing the ***Design Parameter x*** button, the dialogue box presented in fig. II.15.(a) is opened, allowing the user to set the numerical information corresponding to ***x***, consisting in the extreme values to be considered and the number of equally-spaced test-points (where the ***Design Index*** will be calculated as commented bellow). The same way, numerical information can be set for ***y*** by pushing the ***Design Parameter y*** button, which opens the dialogue box presented in fig. II.15.(b).

The set of simulation-experiments defined by the ***Design Parameters*** is launched by pushing the ***Start*** button and cannot be interrupted until scanning all the test-points. The ***Exit*** button serves only for closing the ***Design*** window (i.e. after finishing all the simulation-experiments).



(a) (b)

Fig. II.15. *The dialogue boxes for setting* ***Design Parameter*** (a) ***x*** *and* (b) ***y.***

For each test-point (in the case of a single parameter) or for each pair of test-points (in the case of two parameters) automatically constructed by the ***PN Toolbox*** in accordance with the information given in the ***Design Parameter*** dialogue boxes, a simulation experiment is performed in the ***Run Fast*** mode. Each simulation experiment ends when the condition selected via the combo-box ***End of Simulation*** is fulfilled. The results of all these simulation experiments yield a graphical plot (2-D or 3-D, respectively) defining the dependence of the selected ***Design Index*** **(*I*)** on the ***Design Parameter*****(*s*)**. Besides the graphical plot, the toolbox displays the extreme values of the ***Design Index***, in the message box placed in the left side of the ***Design*** window.

Note that the simulation experiments performed for the given design parameter(s) provide the whole set of global indices associated with all the nodes of the net. Each of these indices can be visualized by appropriately choosing the ***Design Index*** and / or ***Design Node***. Such information remains available as long as the user does not press the buttons ***Design Parameter x***, ***Design Parameter y***, and does not alter the condition in the combo-box ***End of Simulation*** (because these operations reset the specific work variables).

Appendix 1.

XML file-format of a PN model

The ***PN Toolbox*** uses an XML file for saving a model. This file is automatically generated by the ***PN Toolbox*** when the ***Save*** or ***Save As…*** commands are used. It can also be edited by using any text editor, if the format described below is respected.

The root tag for the PN model is **PNToolbox**. The tags for global information are:

- **Model_name** – the name of the file containing the model, including the extension ".xml";
- **Type** – a number that represents the PN type which can take one of the following values: 1 – untimed PN, 2 – T-timed PN, 3 – P-timed PN, 4 – stochastic PN, 5 – generalized stochastic PN;
- **Seed** – a positive integer used to initialize the random number generator of MATLAB;
- **Place** – the necessary information related to a place of the model, depending on the PN type;
- **Transition** – the necessary information related to a transition of the model, depending on the PN type;
- **Arc** – the information related to an arc of the model;
- **Probability** – the information related to the firing probabilities set for a group of conflicting transitions, if any;
- **Priority** – the information related to the firing priorities set for a group of conflicting transitions, if any.

The **Place** tag corresponding to a place in the PN model has the following structure:

- **Id** – the unique identifier of the place; a string in the form "px" with "x" a distinct positive integer;
- **Value** – the coordinates of the cell of the ***Drawing Area*** that contains the place (a couple of integers between 0 and 49);
- **Color** – a string representing the MATLAB color used for drawing the place in the ***Drawing Area***;
- **Label** – additional information used for the graphical representation of the place:
  - **Name** – label of the place that is shown in the ***Drawing Area*** (any string character);

  - **Offset** – a couple of real values representing the offset from the default position of the label (bottom of place);
  - **Visible** – boolean variable corresponding to the status of the label visibility ("yes" or "no").
- **InitialMarking** – initial marking of the place;
- **Capacity** – capacity of the place;
- **Time** – information corresponding to the probability distribution of the time duration assigned to the place; used only in the case of P-timed PNs:
  - **Distribution** – name of the MATLAB distribution function;
  - **Parameters** – values for the parameters of the selected distribution function.

The **Transition** tag corresponding to a transition in the PN model has the following structure:

- **Id** – the unique identifier of the transition; a string in the form "ty" with "y" a distinct positive integer;
- **Value** – the coordinates of the cell of the ***Drawing Area*** that contains the transition (a couple of integers between 0 and 49);
- **Color** – a string representing the MATLAB color used for drawing the transition in the ***Drawing Area***;
- **Message** – a string representing the message displayed by the ***PN Toolbox*** in the ***Message Box*** when the transition is fired;
- **Label** – additional information used for the graphical representation of the transition:
  - **Name** – label of the transition that is shown in the ***Drawing Area*** (any string character);
  - **Offset** – a couple of real values representing the offset from the default position of the label (bottom of transition);
  - **Visible** – boolean variable corresponding to the status of the label visibility ("yes" or "no").
- **Time** – information corresponding to the probability distribution of the time duration assigned to the transition; used only in the case of T-timed, stochastic and generalized stochastic PNs:
  - **Distribution** – name of the MATLAB distribution function;
  - **Parameters** – values for the parameters of the selected distribution function;
  - **Marking_Dependent** – boolean variable showing the dependence of the firing rate of the transition on the marking of its input places ("yes" or "no"); used only in the case of stochastic and generalized stochastic PNs.

The **Arc** tag corresponding to an arc in the PN model has the following structure:

- **Id** – the unique identifier of the arc; a string in the form "az" with "z" a distinct positive integer;
- **From** – the **Id** of the departure node;
- **To** – the **Id** of the arrival node;
- **Style** – arc style (1 – regular, 2 – bidirectional, 3 – inhibitor);
- **Type** – type of graphical representation of the arc (1 – line, 2 – cubic spline);

- **Cubic** – the coordinates in the ***Drawing Area*** of the four points defining the spline curve; used only for the arcs with type = 2;
- **Color** – a string representing the MATLAB color used for drawing the arc in the ***Drawing Area***;
- **Weight** – the weight of the arc.

The **Probability** tag has the following structure:

- **Transitions** – the Ids of the conflicting transitions (separated by comma) subject to the probability assignment;
- **Values** – the corresponding values of the firing probabilities (separated by comma); nonnegative values whose sum equals 1.

The **Priority** tag has the following structure:

- **Transitions** – the **Id**s of the conflicting transitions (separated by comma) subject to the priority assignment;
- **Values** – the corresponding values of the firing priorities (separated by comma); nonnegative values, 0 meaning the highest priority.

Example: The XML file corresponding to the PN model in fig. II.16 is given below.

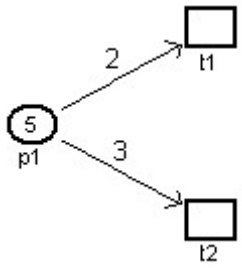


Fig. II.16. *ThePN model used for illustrating the XML file-format used by the **PN Toolbox.***

```
<?xml version="1.0"?>
<PNToolbox>
        <Model_name>XMLexample.xml</Model_name>
        <Type>2</Type>  <!-- T-timed PN -->
        <Seed>66</Seed> <!-- initial seed -->
        <Place> <!-- place definition-->
                <Id>p1</Id> <!-- place's id -->
                <Value>5,43</Value> <!-- coordinates in the Drawing Area-->
                <Color>black</Color> <!-- drawing color-->
                <Label> <!-- information of the label-->
                        <Name>p1</Name> <!-- string label-->
                        <Offset>0.50,-0.20</Offset> <!-- the offset of the representation-->
                        <Visible>yes</Visible> <!-- the visibility of the label-->
                </Label>
                <InitialMarking>5</InitialMarking> <!-- initial marking of the place-->
```

```
        <Capacity>Inf</Capacity> <!-- capacity of the place-->
</Place>
<Transition> <!—transition's definition-->
        <Id>t1</Id> <!-- transition's id-->
        <Value>8,45</Value> <!-- coordinates in the Drawing Area-->
        <Color>black</Color> <!-- drawing color-->
        <Message>Firing transition t1</Message> <!-- message displayed when it is fired-->
        <Label> <!-- information of the label-->
                <Name>t1</Name> <!-- string label-->
                <Offset>0.50,-0.20</Offset> <!-- the offset of the representation-->
                <Visible>yes</Visible> <!-- the visibility of the label-->
        </Label>
        <Time> <!-- information for time distribution function>
                <Distribution>constant</Distribution> <!-- constant distribution-->
                <Parameters>3</Parameters><!-- duration equal to 3-->
        </Time>
</Transition>
<Transition> <!-- second transition's definition -->
        <Id>t2</Id>
        <Value>8,41</Value>
        <Color>black</Color>
        <Message>Firing transition t2</Message>
        <Label>
                <Name>t2</Name>
                <Offset>0.50,-0.20</Offset>
                <Visible>yes</Visible>
        </Label>
        <Time>
                <Distribution>cont. uniform</Distribution> <!-- uniform distribution between 1 and 5-->
                <Parameters>1,5</Parameters>
        </Time>
</Transition>
<Arc> <!-- arc definition>
        <Id>a1</Id> <!-- arc's ID-->
        <From>p1</From> <!-- departure node-->
        <To>t1</To> <!-- arrival node-->
        <Style>1</Style> <!-- regular arc-->
        <Type>1</Type> <!—the graphical representation is a line -->
        <Color>black</Color> <!-- drawing color-->
        <Weight>2</Weight> <!-- arc's weight-->
</Arc>
<Arc>
        <Id>a2</Id>
```

```
            <From>p1</From>
            <To>t2</To>
            <Style>1</Style>
            <Type>1</Type>
            <Color>black</Color>
            <Weight>3</Weight>
        </Arc>
        <Probability> <!-- probability for conflicting transitions t1 and t2; 25% for t1 and 75% for t2 -->
            <Transitions>t1,t2</Transitions>
            <Values>0.25,0.75</Values>
        </Probability>
</PNToolbox>
```

Appendix 2.

Configuration File for the PN Toolbox

The ***PN Toolbox*** uses a configuration file to store the global information needed for initialization. This file, called *PNTconfig.txt*, is stored in the folder where the ***PN Toolbox*** is installed. The user can modify the values of all the parameters; the modifications are effective the next time when the ***PN Toolbox*** is started.

The options from *PNTconfig.txt* file used by ***PN Toolbox*** are:

- **Seed** – the initial state of the *random number generator* version 5 from MATLAB. This parameter is called "Seed" for history reasons; first version of ***PN Toolbox*** used version 4 of the *random number generator* where the 'seed' option of the MATLAB **rand** function was used. The version 5 uses 'state' parameters of **rand** function. (this parameter can have any integer value);
- **DisplayMessages** – boolean value that gives the status of the transitions' messages visibility in the ***Message Box*** in ***Step*** and ***Run Slow*** simulation modes (1 – the messages are visible, 0 – the messages are invisible);
- **Background** – the background color for all MATLAB figures opened by the ***PN Toolbox*** (a vector with three elements ranging from 0 to 1);
- **HTMLBackground** –the background color for all HTM files opened by the ***PN Toolbox*** (this parameter is given in hexadecimal);
- **HTMLText** – the text color for all HTML files opened by the ***PN Toolbox*** (given in hexadecimal);
- **Breakpoint** – the default breakpoint for ***Run Fast*** simulation mode (1 – Number of events, 2 – Simulation time, 3 – Firings for transition, 4 – Tokens arrived in place);
- **Breakpoint_Event** – default number of events, if breakpoint is selected on *Number of events*;
- **Breakpoint_Time** – default simulation time, if breakpoint is selected on *Simulation time*;
- **Breakpoint_Transition** – default index of the transition in the transitions array, if breakpoint is selected on *Firings for transition*;
- **Breakpoint_ServiceSum** – the number of firings for the **Breakpoint_Transition**;
- **Breakpoint_Place** – default index of the place in the places array, if breakpoint is selected on *Tokens arrived in place*;

- **Breakpoint_ArrivalSum** – the number of tokens arrived in the **Breakpoint_Place**;
- **Projects_Save** – string with a MATLAB expression (constructed with the **fullfile** command) that, after execution returns, the whole path of the folder where the ***PN Toolbox*** saves the new models;
- **Color_Wait_Token** – MATLAB color used for the graphical representation of places that contain reserved tokens (only in the case of *P*-timed PNs);
- **Color_Fire_Transition** – MATLAB color used for the graphical representation of firing transitions;
- **NewFileName** – the default name of the new model, including extension (.xml);
- **ColorPlaces** – MATLAB color used for drawing the places;
- **ColorTransitions** – MATLAB color used for drawing the transitions;
- **ColorArcs** – MATLAB color used for drawing the arcs.

Appendix 3.

Available Demos ...

The current version of the Petri Net Toolbox is accompanied by four movies illustrating its usage in the simulation, analysis and design of discrete-event systems.

Demo 1. Computer system with two processors sharing two disks (in parallel) Version of the Two Dinning Philosophers Problem (Dijkstra, 1968).
This demo illustrates:

- construction of an untimed Petri net model
- analysis of deadlock (via the coverability tree)
- prevention of deadlock through lookahead feedback
- access to the following information about the Petri net model:
- incidence matrix
- minimal-support P- and T-invariants
- structural properties

Demo 2. Manufacturing system with a sequentially shared robot .
This demo illustrates:

- construction of a P-timed Petri net model
- analysis of deadlock (via simulation)
- prevention of deadlock by limiting the number of pallets
- analysis of time-dependent performance indices
- study of a performance index depending on two design parameters

Demo 3. Flow-shop system with three machines.
This demo illustrates:

- simulation and animation in the Run Slow mode
- record of the simulation results in a log file
- computation of the cycle time
- max-plus analysis of a place-timed event graph
- max-plus state-space representation
- setting of the values for the input vectors
- max-plus based simulation

- plots of the components for the input, state or output vectors

Demo 4. Open markovian queueing network.
This demo illustrates:

- construction of a generalized stochastic Petri net model
- usage of the Scope and Diary facilities
- analysis of time-dependent performance indices

References

Ajmone Marsan, M., Balbo, G., Bobio, A., Chiola, G., Conte, G. and Cumani, A. (1985), "On Petri nets with stochastic timing", *Int. Wksp. Timed Petri Nets*, Torino, pp. 80-87.

Ajmone Marsan, M. (1990), "Stochastic Petri nets: An elementary introduction", in Rozenberg, G.: *Lecture Notes in Computer Science*, Vol. **424**, *Advances in Petri Nets 1989*, pp. 1-29, Springer-Verlag, Berlin.

Antsaklis, P.J. and Moody, J.O. (1998) *Supervisory Control of Discrete Event Systems Using Petri Nets*, Kluwer Academic Publishers.

Bacelli, F., Cohen, G., Olsder, G.J. and Quadrat, J.P. (1992), *Synchronization and Linearity. An Algebra for Discrete Event Systems*, John Wiley and Sons, New York.

Bolch, G., Greiner, S., de Meer, H. and Trivedi, K. (1998). *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*, John Wiley and Sons, New York.

Cao, X.R. and Ho, Y.C. (1990), "Models of discrete event dynamic systems", *IEEE Control Systems Magazine*, Vol. **10**, pp. 69-75.

Cassandras, C.G. (1993), Discrete Event *Systems*: *Modeling and Performance Analysis*, Irwin.

Cassandras, C.G., Lafortune, S. and Olsder, G.J. (1995), "Introduction to the modelling, control and optimization of discrete event systems", in Isidori, A. (Ed.), *Trends in Control - A European Perspective*, Springer-Verlag, pp. 217-292.

Cohen, G., Gaubert, S. and Quadrat, J.P. (1998), "Max-plus algebra and system theory: where we are and where to go now", *IFAC Conf. on System Structure and Control*, Nantes.

Cohen, G., Gaubert, S. and Quadrat, J.P. (2001) “Max-plus algebra in Scilab and applications”, *Scilab Workshop*, Liama, Pekin.

David, R. et Alla, H. (1992), *Du Grafcet aux réseaux de Petri* (2e édition), Hermes, Paris.

David, R. and Alla, H. (1994), "Petri nets for modeling of dynamic systems - A survey", *Automatica*, Vol. **30**, pp. 175-202.

David, R. et Alla, H. (2005), *Discrete, Continuous, and Hybrid Petri nets*, Springer-Verlag, Berlin-Heidelberg.

Desrochers, A.A. and Al-Jaar, R.Y. (1993), Petri *Nets* for *Automated Manufacturing Systems*: *Modeling*, Control *and Performance Analysis*, Rensselaer Polytechnic Institute.

Dijkstra, E. W. (1968), "Co-operating sequential processes", in Genyus, F. (ed.) *Programming Languages*, New York Academic, pp. 43-112.

Gürel, A., Pastravanu, O.C. and Lewis, F.L. (1994) "A robust approach in deadlock-free and live FMS design", *Proc. 2nd IEEE Mediterranean Symp. New Directions in Control and Automation*, pp. 40–47, Crete.

Gürel, A., Pastravanu, O.C. and Lewis, F.L. (1998) "A system-theoretic approach for discrete-event control of manufacturing systems", in Gunawardena, J. (Ed.) *Idempotency*, pp. 242-261, Cambridge University Press.

Ho, Y.C. and Cassandras, C.G. (1983) "A new approach to the analysis of discrete event dynamic systems", *Automatica*, Vol. **19**, pp. 149–167.

Huang, H.H., Lewis, F.L., Pastravanu, O.C. and Gürel, A. (1995) "Flowshop scheduling design in an FMS matrix framework", *Control Engineering Practice*, Vol. **3**, pp. 561–568.

Ichikawa, A. and Hiraishi, K. (1987) "Analysis and control of discrete event systems represented by Petri nets", *Lecture Notes in Control and Information Science*, Vol. **103**, pp. 115–134, Springer-Verlag.

Iordache, M.V. and Antsaklis, P.J. (2002) "Software tools for the supervisory control of Petri nets based on place invariants", Technical Report ISIS-2002-003, University of Notre Dame, IN, USA, http://www.nd.edu/~isis/techreports/isis-2002-003.pdf.

Jeng, M.D. and DiCesare, F. (1993) "A review of synthesis techniques for Petri nets with applications to automated manufacturing systems", *IEEE Trans. on Systems, Man and Cybernetics*, Vol. **23**, pp. 301–313.

Jeng, M.D. and DiCesare, F. (1995) "Synthesis using resource control nets for modelling shared-resource systems", *IEEE Trans. on Robotics and Automation*, Vol. **11**, pp. 317–327.

Lewis, F.L. Pastravanu, O. and Huang, H.H. (1993) "Controler design and conflict resolution for discrete event manufacturing systems", *Proc. 32nd IEEE Conf. Decision and Control,* pp. 3288–3293, San Antonio.

Lewis, F.L., Huang, H.H., Pastravanu, O. and Gürel, A. (1995) "Control systems design for flexible manufacturing systems", in Raouf, A. and Ben-Daya, M. (Eds.) *Flexible Manufacturing Systems: Recent Developments*, pp. 253–290, Elsevier Science.

Lewis, F.L., Tacconi, D., Gürel, A., Huang, H.H. and Pastravanu, O. (1997) "Manufacturing controller design and deadlock avoidance using a matrix model for discrete event systems", in Tzafestas, S.G. (Ed.) *Methods and Applications of Intelligent Control*, pp. 485-508, Kluwer Academic Publishers.

Lewis, F.L., Gürel, A., Bogdan, S., Doganalp, A. and Pastravanu, O. (1998) "Analysis of deadlock and circular waits using a matrix model for flexible manufacturing systems", *Automatica*, Vol. **34**, No. 9, pp. 1083-1100.

Lindemann, C. (1998), *Performance Modelling with Deterministic and Stochastic Petri Nets*. John Wiley and Sons, Chichester.

Martinez, J. and Silva, M. (1982), "A simple and fast algorithm to obtain all invariants of a generalized Petri net", in Girault, C. and Reisig, W. (Eds.), *Application and Theory of Petri Nets*, Informatik Fachberichte 52, Springer, pp. 301-310.

The MathWorks Inc., *Creating Graphical User Interfaces*, Natick, Massachusetts, 2005.

The MathWorks Inc., *MATLAB Programming*, Natick, Massachusetts, 2005.

Mortensen, K.H. (2003), *Petri Nets Tools and Software*, http://www.daimi.au.dk/PetriNets/tools.

Murata, T. (1989), "Petri nets: properties, analysis and applications", *Proc. IEEE*, Vol. **77**, pp. 541-580.

Pastravanu, O. (1997), *Discrete Event Systems. Qualitative Techniques in a Petri Net Framework* (in Romanian), Matrix Rom Publishing House, Bucharest.

Pastravanu, O., Gürel, A., Huang, H.H. and Lewis, F.L. (1994) "Rule–based controller design algorithm for discrete event manufacturing systems", *Proc. American Control Conf.*, Baltimore, pp. 299–305.

Pastravanu, O., Gürel, A. and Lewis, F.L. (1994) "Petri net based deadlock analysis in flowshops with kanban-type controllers", *Proc. 10th ISPE/IFAC Int. Conf. on CAD/CAM, Robotics and Factories of Future*, Ottawa, pp. 75–80.

Pastravanu, O., Gürel. A. and Lewis, F.L. (1997) "Teaching discrete event control of manufacturing systems", in Bir, A., Dinibutun, A.T., Eksin, I. and Istefanopulos,Y. (Ed.) *Proc. of the 4th IFAC Symposium on Advances in Control Education ACE'97*, Istanbul, pp. 307-312.

Pastravanu, O., Matcovschi, M.H. and Mahulea, C. (2002), *Applications of Petri Nets in the Study of Discrete Event Systems* (in Romanian), Gh. Asachi Publishing House, Iasi.

Peterson, J.L. (1981) *Petri Net Theory and the Modeling of Systems*, Prentice–Hall.

Petri, C.A (1962), *Kommunikation mit Automaten*, Institut für Instrumentelle Mathematik Bonn, Schriften des IIM Nr. 2.

Ramadge, P.J. and Wonham, W.M. (1987) "Supervisory control of a class of discrete event processes", *SIAM J. Control and Oprimization*, Vol. **25**, pp. 206–230.

Ramamoorthy, C.V. and Ho, G.S. (1980) "Performance evaluation of asynchronous concurrent systems using Petri nets", *IEEE Trans. on Software Eng.*, Vol. **6** (5), pp. 440-449.

Reisig, W. (1985), *Petri Nets - An Introduction*, Springer-Verlag.

Sinclair, J.B. (2002). *Simulation of Computer Systems and Computer Networks: A Process-Oriented Approach*, http://www.owlnet.rice.edu/~elec428/yacsim/book.ps.

Trivedi, K. (2000). *Computer Science Applications*, John Wiley & Sons, New York.

Zhou, M.C. and DiCesare, F. (1993), *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*, Kluwer, Boston.

Zurawski, R. and Zhou, M.C. (1994), "Petri nets and industrial applications: A tutorial", *IEEE Trans. on Ind. Electronics*, Vol. **41**, No. *6*, pp. 567-583.