ICCS 2004, Cracow, Poland

Modern Technologies for Web-based Adaptive Systems

MTWAS Workshop, June 6-10, 2004

An application of the DEDS control synthesis method

František Čapkovič

Institute of Informatics, Slovak Academy of Sciences

Dúbravská cesta 9, Bratislava, Slovak Republic

Frantisek.Capkovic@savba.sk

http://www.ui.sav.sk/home/capkovic

Contents

- Introduction
  - Basic definition of DEDS
  - Petri nets in DEDS modelling
  - Directed graphs in DEDS modelling
- DEDS control synthesis
  - Definition of the control synthesis
  - Basic principle of the proposed control synthesis method

- Examples
  - Client-server cooperation
  - Two agents cooperation
- Adaptivity
  - Two kinds of the adaptivity
  - Simple example
- Conclusions

Introduction

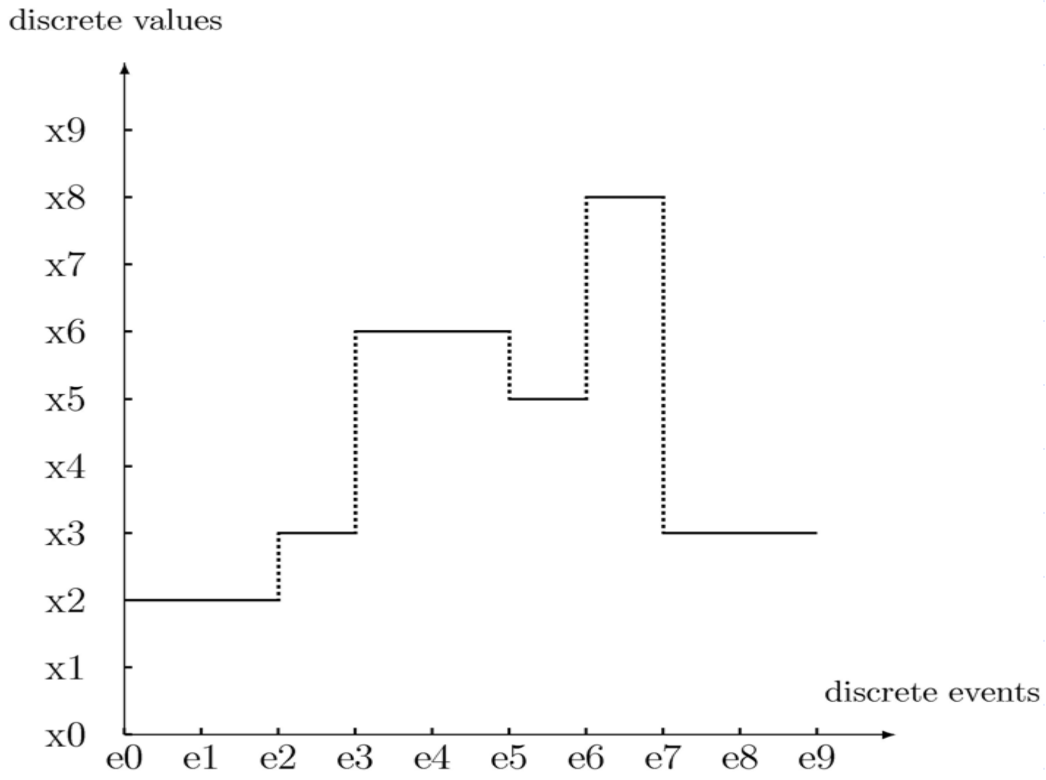
Basic definition of DEDS

DEDS (discrete event dynamic systems) are the systems where the development of the system dynamics depends on the occurency of discrete events, i.e. DEDS are systems driven by discrete events.

Typical kinds of DEDS

- flexible manufacturing systems
- communication systems
- transport systems

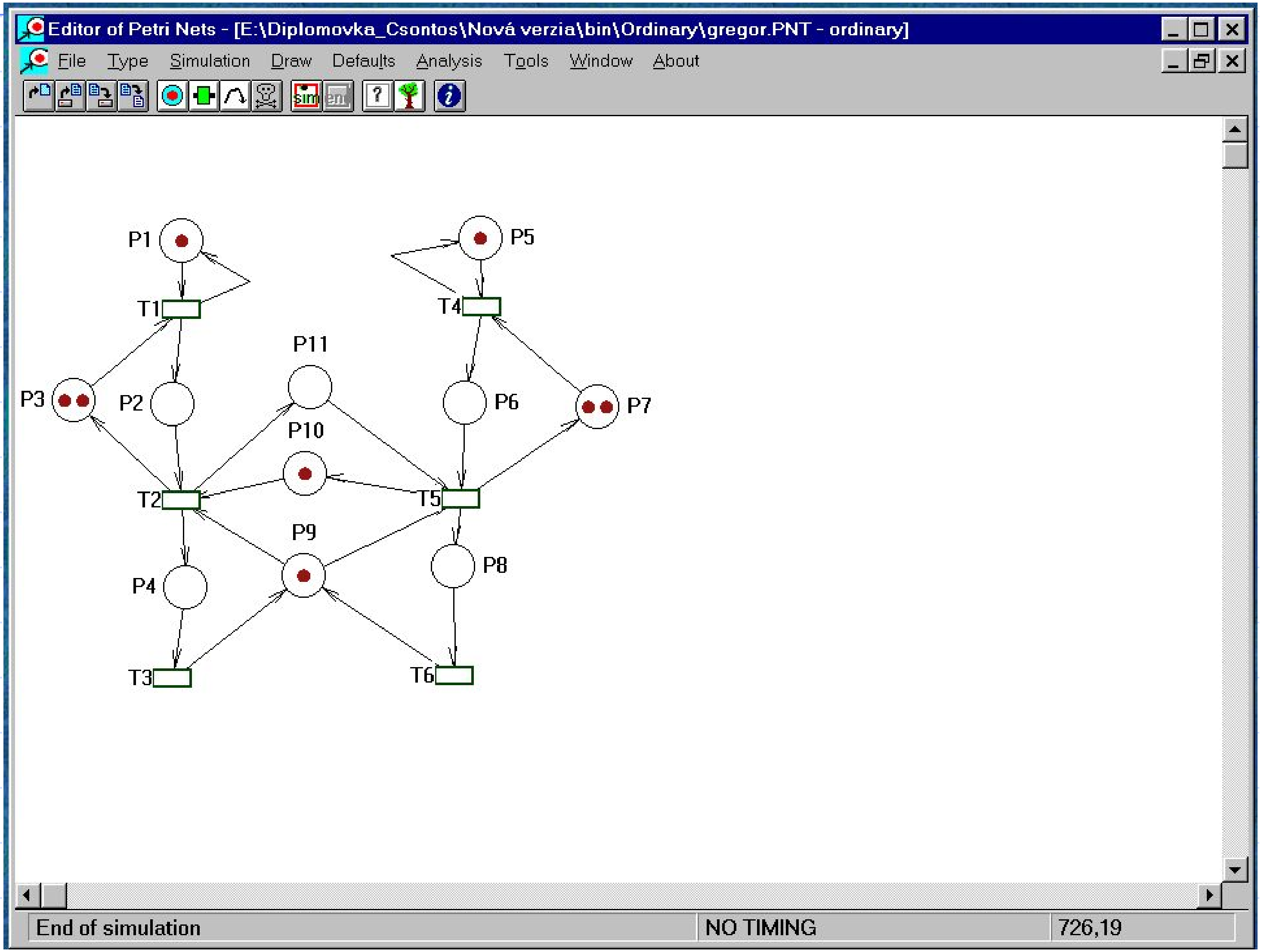
The graphical expression of a DEDS variable



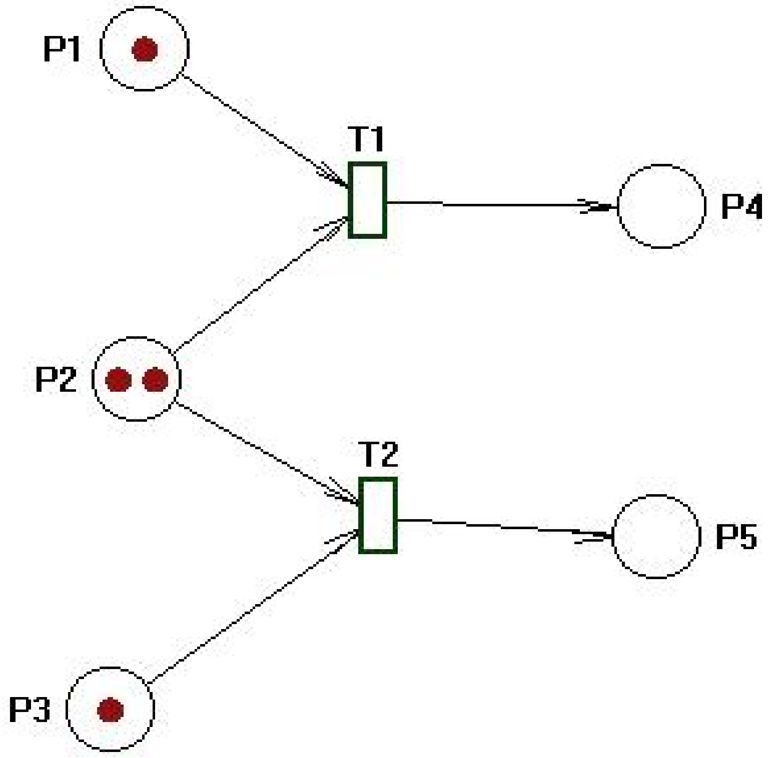
Petri nets (PN) in DEDS modelling

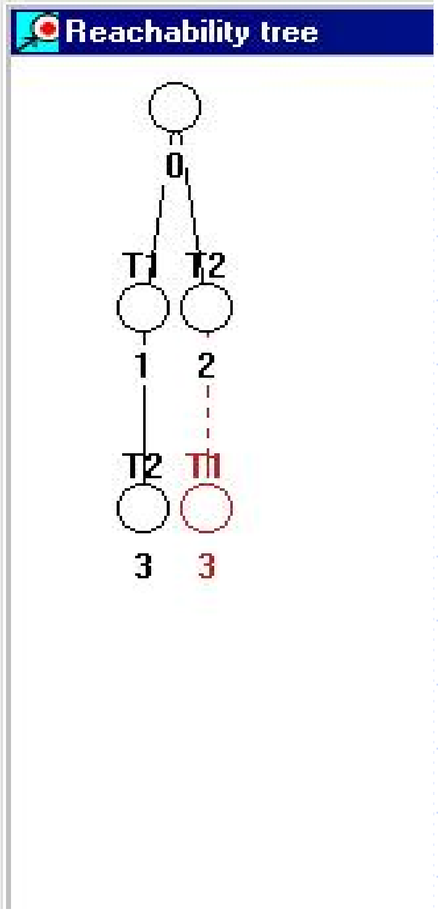
- PN are able to express parallelism and conflict situations
- PN can be expressed in analytical terms (in the form of the linear discrete system)
- PN properties can be tested by means of the reachability tree and invariants
- PN allow to use analytical approach to the DEDS control synthesis

Example of a Petri net

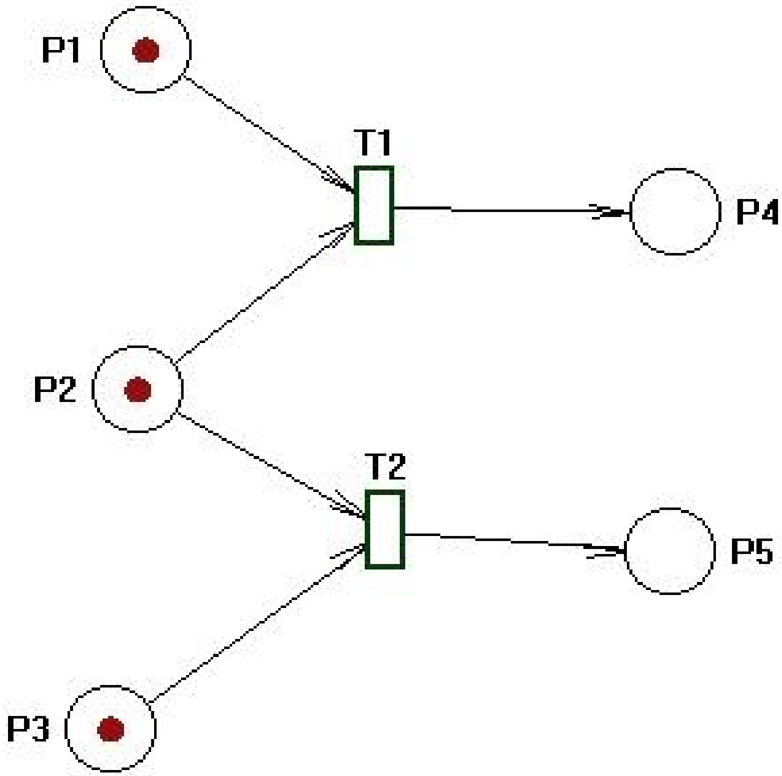


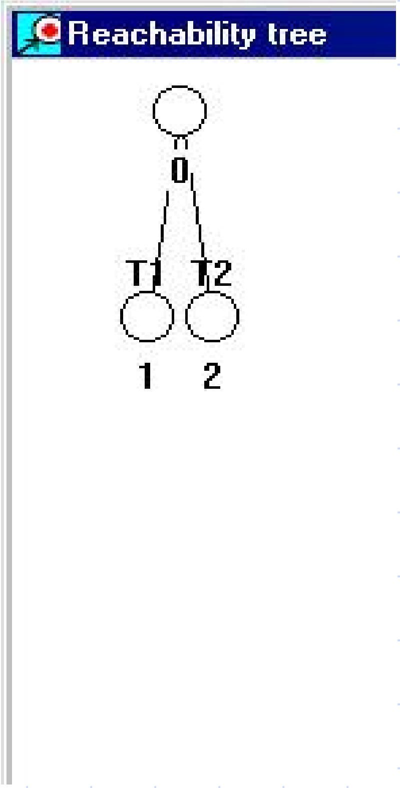
Parallelism





Conflict situation





Formal expression of the Petri net structure

$$\langle P, T, F, G \rangle; \quad P \cap T = \emptyset; \quad F \cap G = \emptyset$$

$$P = \{p_1, \ldots, p_n\}$$

$$T = \{t_1, \ldots, t_m\}$$

$$F \subseteq P \times T$$

$$G \subseteq T \times P$$

Formal expression of the Petri net dynamics

$$\langle\, X, U, \delta, \mathbf{x}_0 \,\rangle\,; \quad X \cap U = \emptyset$$

$$X = \{\mathbf{x}_0,\ \mathbf{x}_1 \ldots,\ \mathbf{x}_N\}$$

$$U = \{\mathbf{u}_0,\ \mathbf{u}_1 \ldots,\ \mathbf{u}_N\}$$

$$\delta : X \times U \longrightarrow X$$

$\mathbf{x}_0$ is an initial state

Mathematical model of Petri net

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k + \mathbf{B}.\mathbf{u}_k \quad , \qquad k = 0, N \\ \mathbf{B} &= \mathbf{G}^T - \mathbf{F} \\ \mathbf{F}.\mathbf{u}_k &\leq \mathbf{x}_k \end{aligned}$$

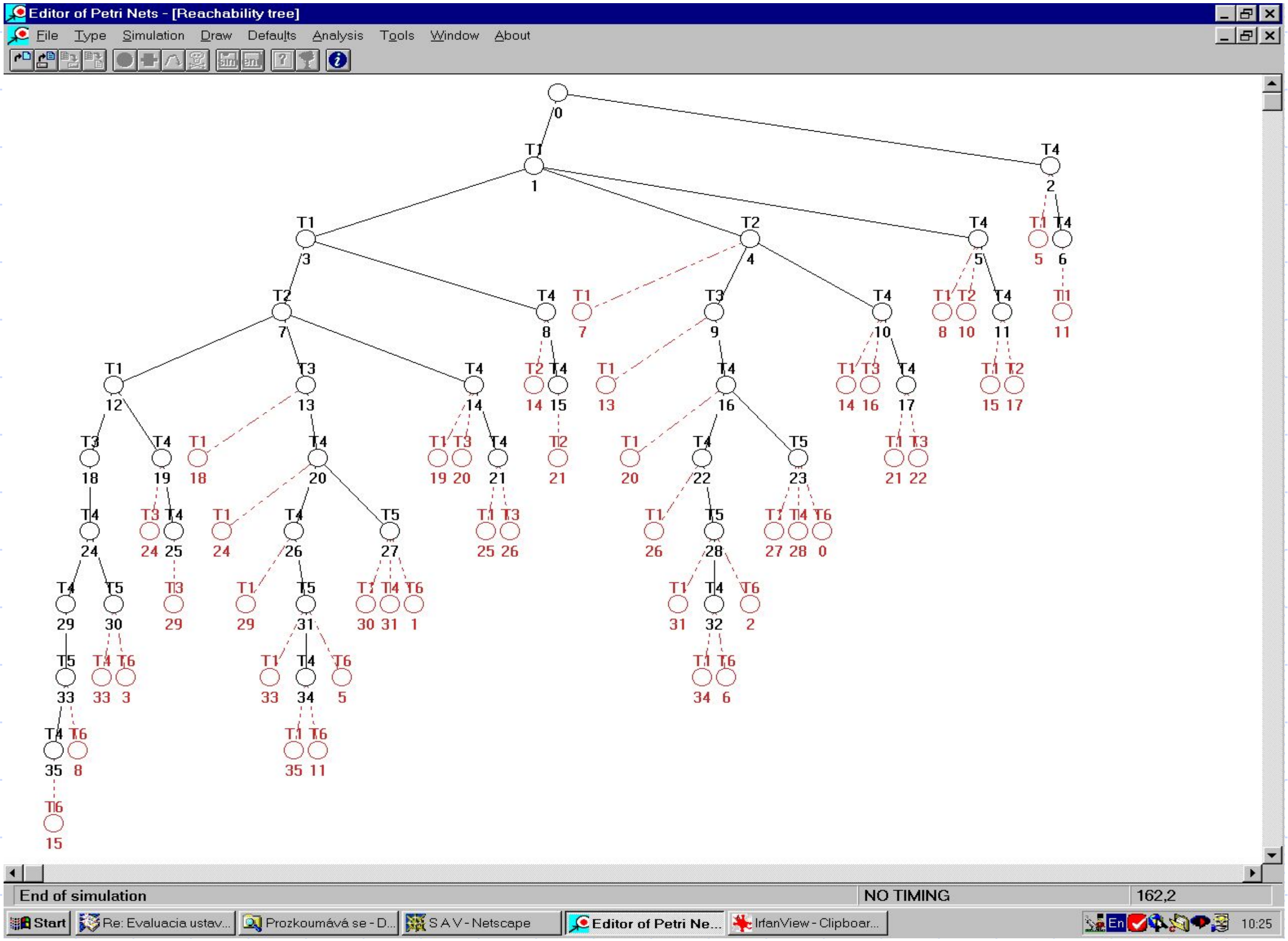
$$\mathbf{x}_k = (\sigma_{p_1}^k, \dots, \sigma_{p_n}^k)^T$$

$$\sigma_{p_i}^k \in \{0, c_{p_i}\}, \; i = 1, n$$

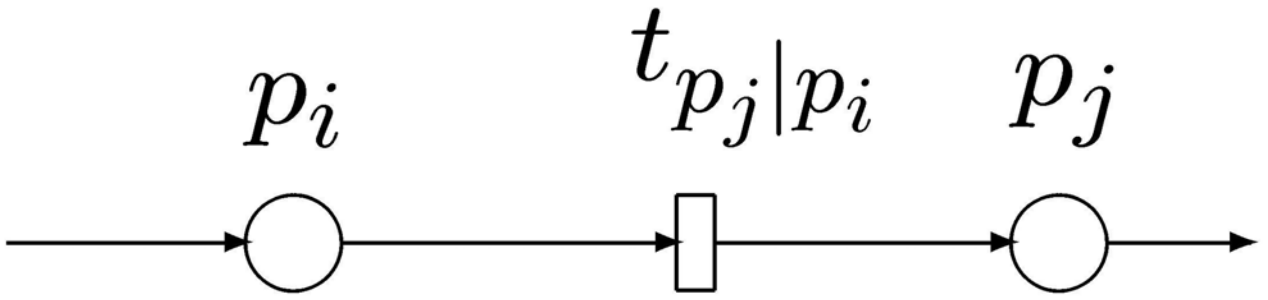
$$\mathbf{u}_k = (\gamma_{t_1}^k, \dots, \gamma_{t_m}^k)^T$$

$$\gamma_{t_j}^k \in \{0, 1\}, \; j = 1, m$$

Reachability tree of the above introduced Petri net



Directed graphs (DG) in DEDS modelling



$$\gamma_{t_{\pi_i|\pi_j}}^{(k)} \in \{0,\ 1\}$$

$$\mathbf{X}(k+1) = \mathbf{\Delta}_k . \mathbf{X}(k) \quad , \quad k = 0, N$$

$$\mathbf{X}(k) = (\sigma_{\pi_1}^{(k)}(\gamma), ..., \sigma_{\pi_{n_{RT}}}^{(k)}(\gamma))^T, \, k = 0, N$$

$$\mathbf{\Delta}_k = \{\delta_{ij}^{(k)}\}_{n_{RT} \times n_{RT}}$$

$$\delta_{ij}^{(k)} = \gamma_{t_{\pi_i | \pi_j}}^{(k)}, \, i = 1, n_{RT}, \, j = 1, n_{RT}$$

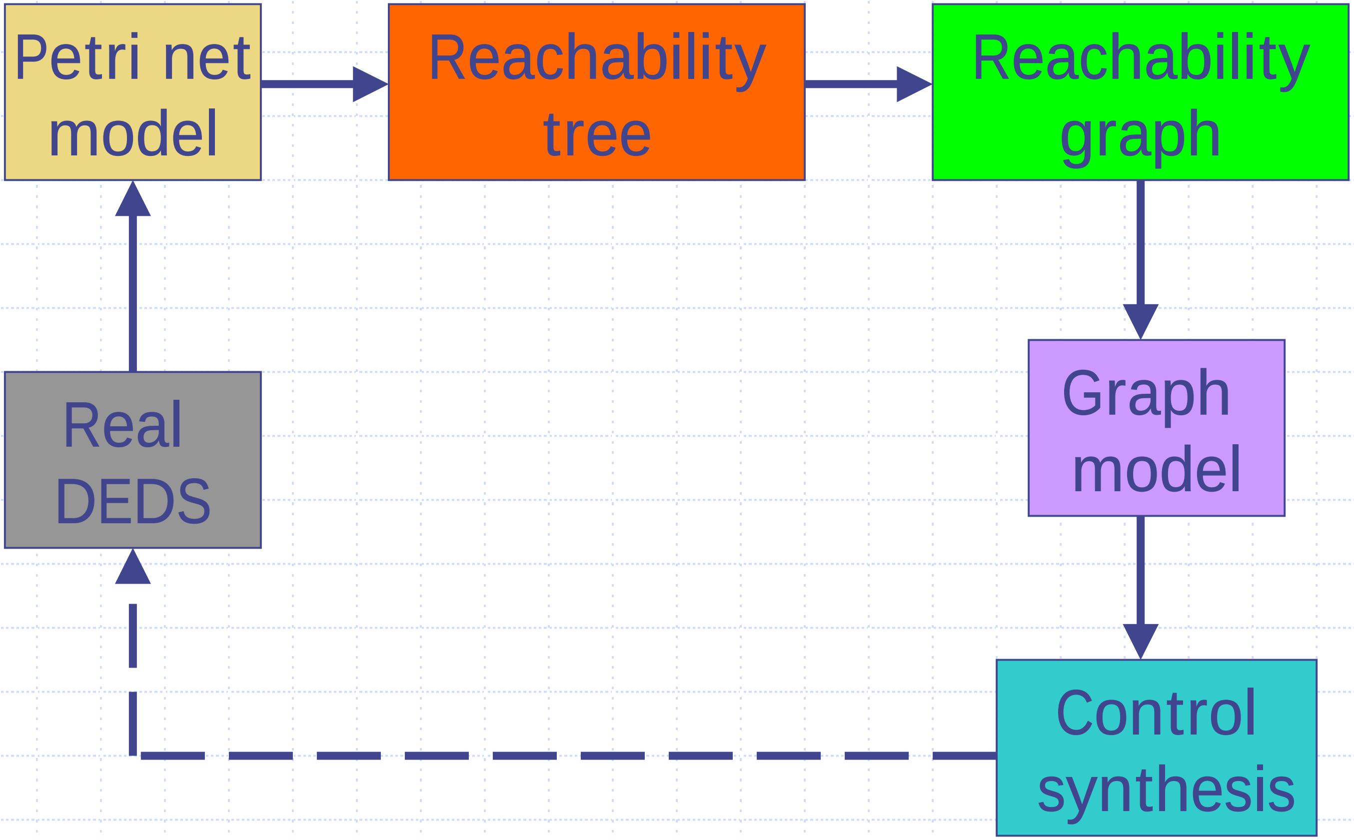
State machines

Petri nets where each transition has only one input and only one output position are named state machines.
They can be modelled by directed graphs (DG) without any problem.

Petri nets with general structure

In case of the general structure, when any transition is allowed to have more input positions and more output ones, the PN reachability graph has to be used.

Transforming the PN model to the DG model



DEDS control synthesis

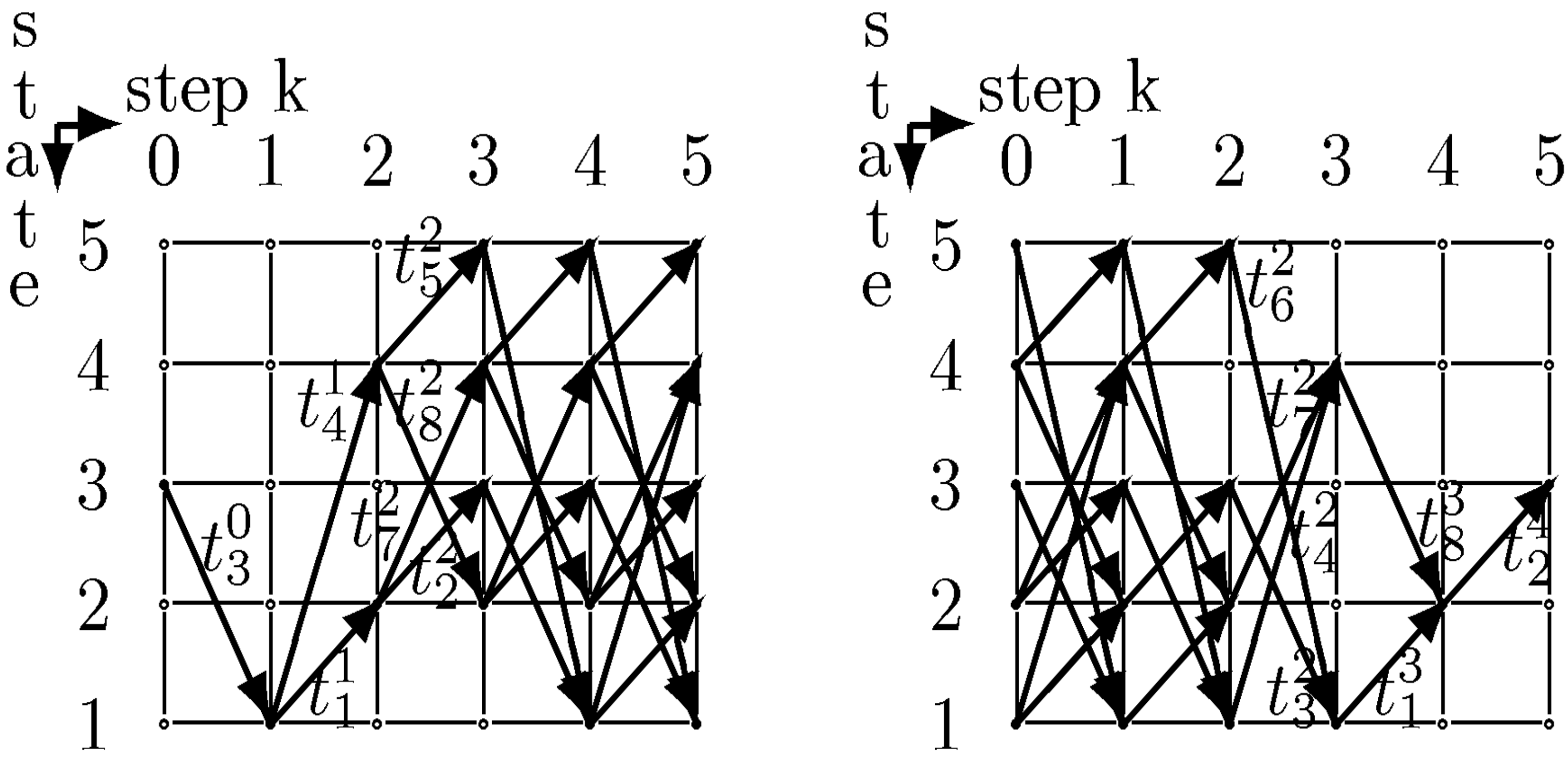
Definition of the control synthesis

Control synthesis = finding the most suitable sequence of discrete events (control interferences) which is able to ensure the transition (transformation) of the system from a given initial state into a prescribed terminal state at simultaneous fulfilling control task specifications that are imposed on the control task.

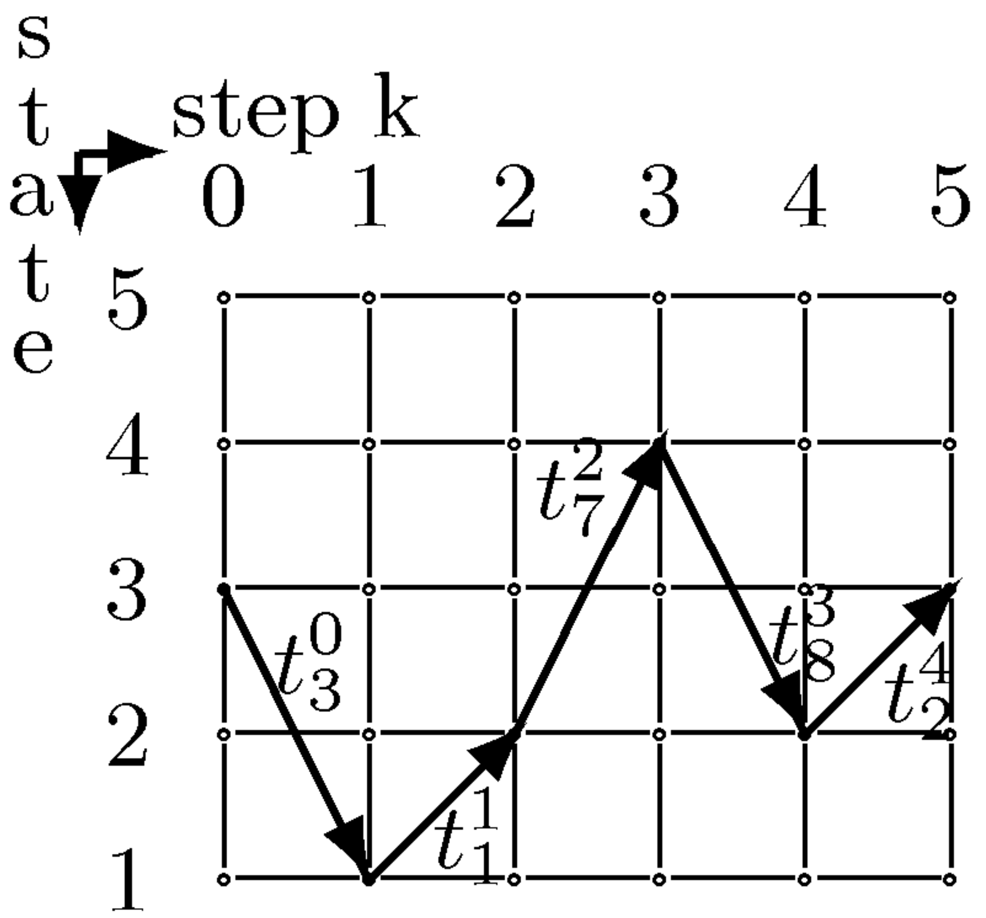
Control task specifications = criteria, constraints, etc. Usually, they are not given in analytical terms. Even, often they are given only verbally.

Basic principle of the proposed control synthesis method

Straight-lined reachability tree and the backtracking one



Intersection of the trees = state trajectory(-ies)



Procedure in analytical terms

- The staight-lined reachability tree (SLRT)

$$
\begin{aligned}
\{\mathbf{X}_1\} &= \mathbf{\Delta}.\mathbf{X}_0 \\
\{\mathbf{X}_2\} &= \mathbf{\Delta}.\{\mathbf{X}_1\} = \mathbf{\Delta}.(\mathbf{\Delta}.\mathbf{X}_0) = \mathbf{\Delta}^2.\mathbf{X}_0 \\
& \quad \dots \qquad \dots \qquad \dots \\
\{\mathbf{X}_N\} &= \mathbf{\Delta}.\{\mathbf{X}_{N-1}\} = \mathbf{\Delta}^N.\mathbf{X}_0
\end{aligned}
$$

- The backtracking reachability tree (BTRT)

$$
\begin{aligned}
\{\mathbf{X}_{N-1}\} &= \boldsymbol{\Delta}^T.\mathbf{X}_N \\
\{\mathbf{X}_{N-2}\} &= \boldsymbol{\Delta}^T.\{\mathbf{X}_{N-1}\} = (\boldsymbol{\Delta}^T)^2.\mathbf{X}_N \\
& \dots \qquad \dots \qquad \dots \\
\{\mathbf{X}_0\} &= \boldsymbol{\Delta}^T.\{\mathbf{X}_1\} = (\boldsymbol{\Delta}^T)^N.\mathbf{X}_N
\end{aligned}
$$

The intersection of the SLRT and BTRT

$$\begin{aligned}
\mathbf{M}_1 &= (\mathbf{X}_0, {}^1\{\mathbf{X}_1\}, \ldots, {}^1\{\mathbf{X}_{N-1}\}, {}^1\{\mathbf{X}_N\}) \\
\mathbf{M}_2 &= ({}^2\{\mathbf{X}_0\}, {}^2\{\mathbf{X}_1\}, \ldots, {}^2\{\mathbf{X}_{N-1}\}, \mathbf{X}_N) \\
\mathbf{M} &= \mathbf{M}_1 \cap \mathbf{M}_2 \\
\mathbf{M} &= (\mathbf{X}_0, \{\mathbf{X}_1\}, \ldots, \{\mathbf{X}_{N-1}\}, \mathbf{X}_N)
\end{aligned}$$

Using the principle of causality

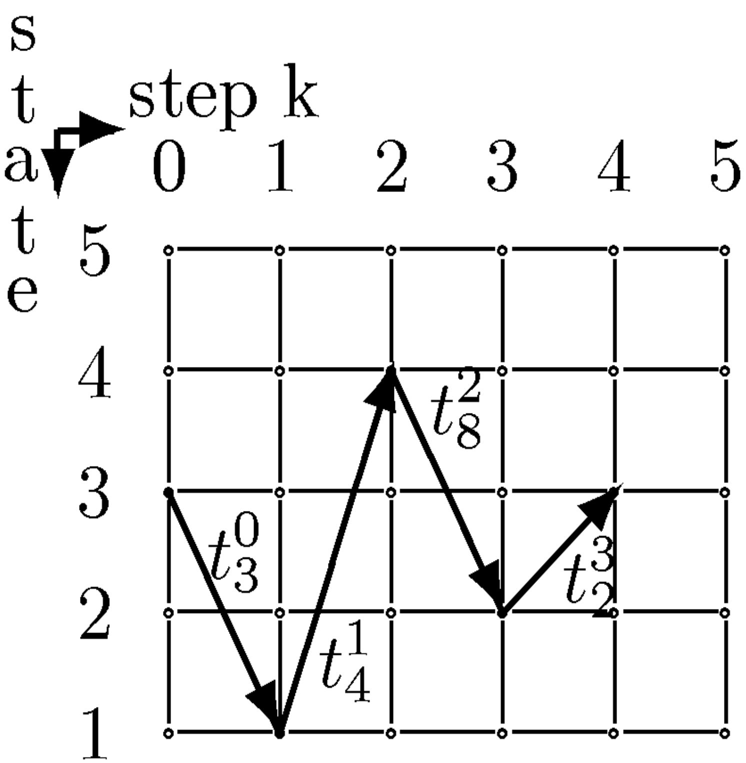
Due to the principle of causality any shorter feasible solution is involved in the longer feasible one. Hence, when

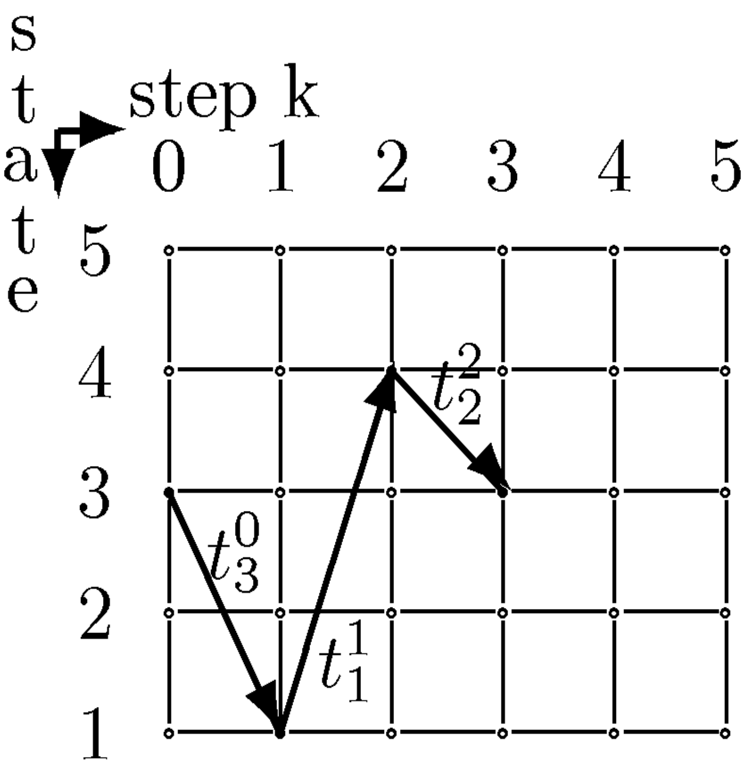
$\mathbf{M}_2$ is shifted to the left before the intersection.

$$^{-1}\mathbf{M} = (\mathbf{x}_0, \{\mathbf{x}_1\}, \ldots, \{\mathbf{x}_{N-2}\}, \mathbf{x}_{N-1}) \qquad (18)$$

where $\mathbf{x}_{N-1} = \mathbf{x}_t$.
Shifting (finding the $(n \times (N-k+1))$ matrices $^{-k}\mathbf{M}, k = 1, 2, \ldots$) can continue until the intersections exists, i.e. until $\mathbf{x}_0 \in {}^2\{\mathbf{x}_k\}$ and $\mathbf{x}_t \in {}^1\{\mathbf{x}_{N-k}\}$.

Shorter trajectories obtained by shifting





Example 1 – Client-server connection

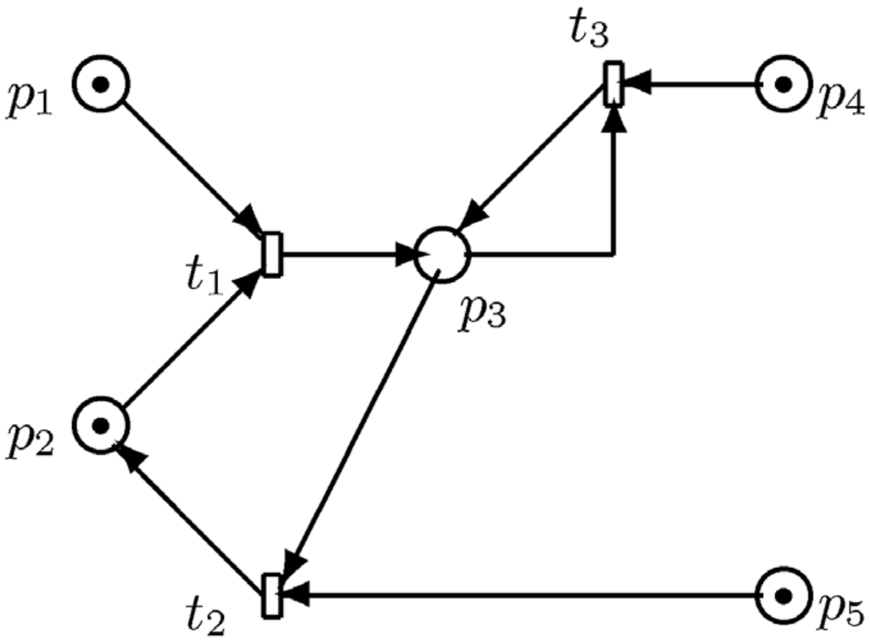
The places of the PN-based model

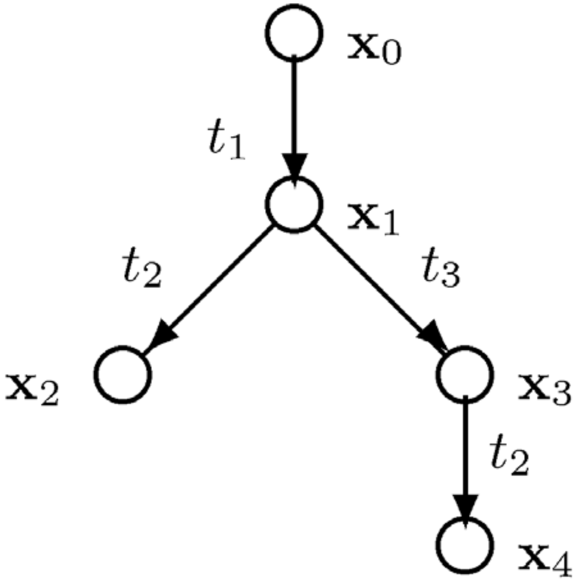
p1 - the client (C) requests for the connection
p2 - the server (S) is listening
p3 - the connection of C with S
p4 - the data sent by C to S
p5 - the disconnection of C by the C himself

The transitions of the PN-based model

t1, t2, t3 – discrete events realizing the system dynamics

PN-based model and reachability tree (RT)





Parameters of the PN-based model

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad \mathbf{G} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$\mathbf{x}_0 = (1,\ 1,\ 0,\ 1,\ 1)^T$$

MATLAB procedure for enumerating the RT

```
Xreach=x0
Art=[0]
[n,m]=size(F);
B=Gt-F
i=0
while i < size(Xreach,2)
        i=i+1;
        for k=1:m
            x(k)=all(Xreach(:,i) >= F(:,k));
        end
        findx=find(x)
        for k=1:size(findx,2)
```

```
            bb = Xreach(:,i)+B(:,findx(k));
            matrix=[];
            for j=1:size(Xreach,2)
                matrix=[matrix,bb];
            end;
            v=all(matrix == Xreach);
            j=find(v);
            if any(v)
                Art(i,j)= findx(k);
            else
                Xreach=[Xreach,bb];
                Art(size(Art,1)+1,size(Art,
                Art(i,size(Art,2))=findx(k)
            end;
        end;
Xreach;
Art;
end
```

Enumerated RT

Quasi-functional adjacency matrix of RT

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \qquad \boldsymbol{\Delta}_k^T = \begin{pmatrix} 0 & t_1 & 0 & 0 & 0 \\ 0 & 0 & t_2 & t_3 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & t_2 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Space of reachable states

$$\mathbf{X}_{reach} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Example 2 – Two agents cooperation

The agent A needs to do the activity (i.e. to solve a problem) P. However, A is not able to do P. Consequently, A requests the agent B to do P for him.

The places of the PN-based model:

p1 – A wants to do P
p2 - A waits for an answer from B
p3 - A waits for a help from B
p4 - the failure of the cooperation
p5 - the satisfying cooperation
p6 - A requests B to do P
p7 - B refuses to do P

p8 - B accepts the request of A to do P

p9 - B is not able to do P

p10- doing P by B

P11- B receives the request of A

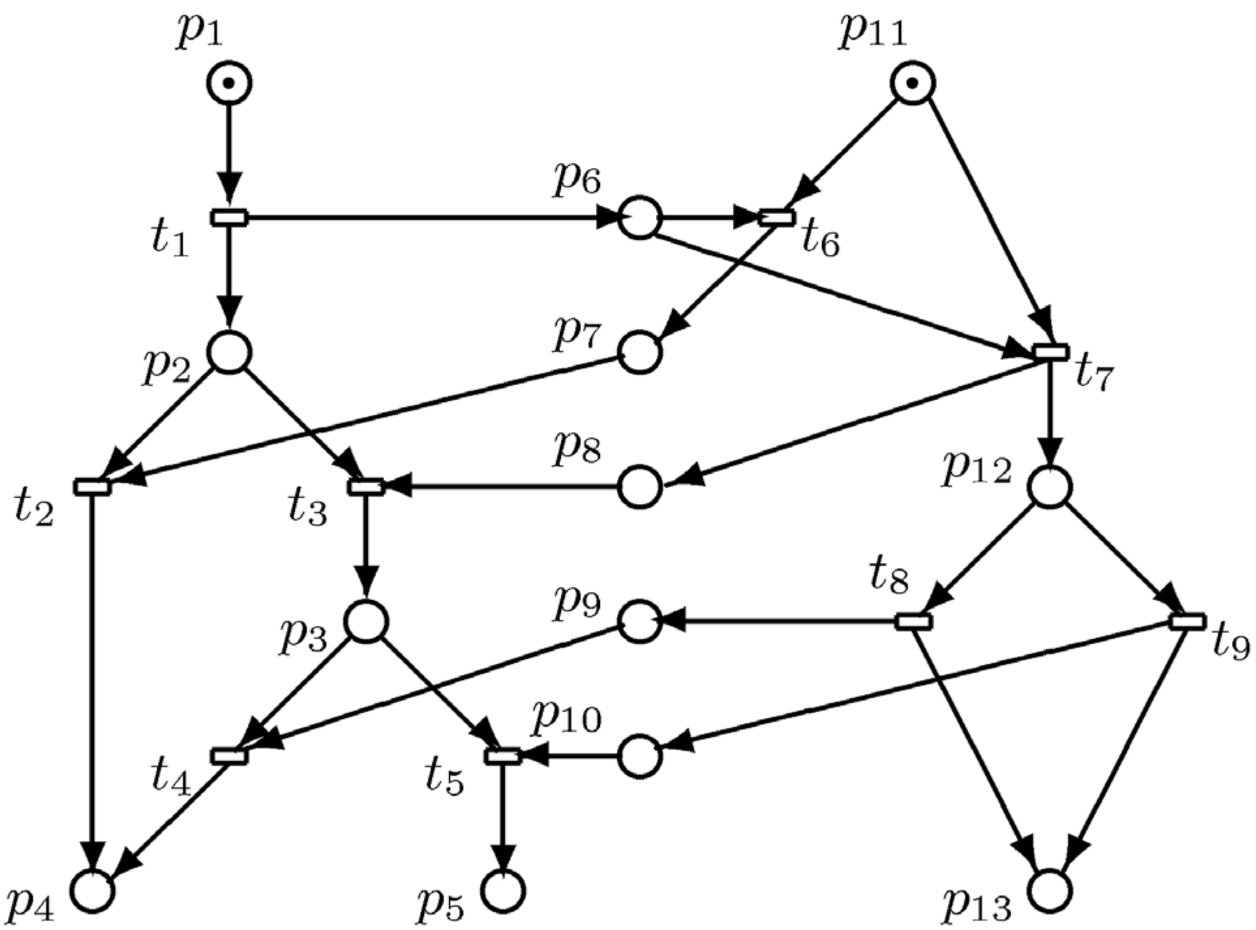
p12- B is willing to do P for A

p13- the end of the work of B

The transitions of the PN-based model:

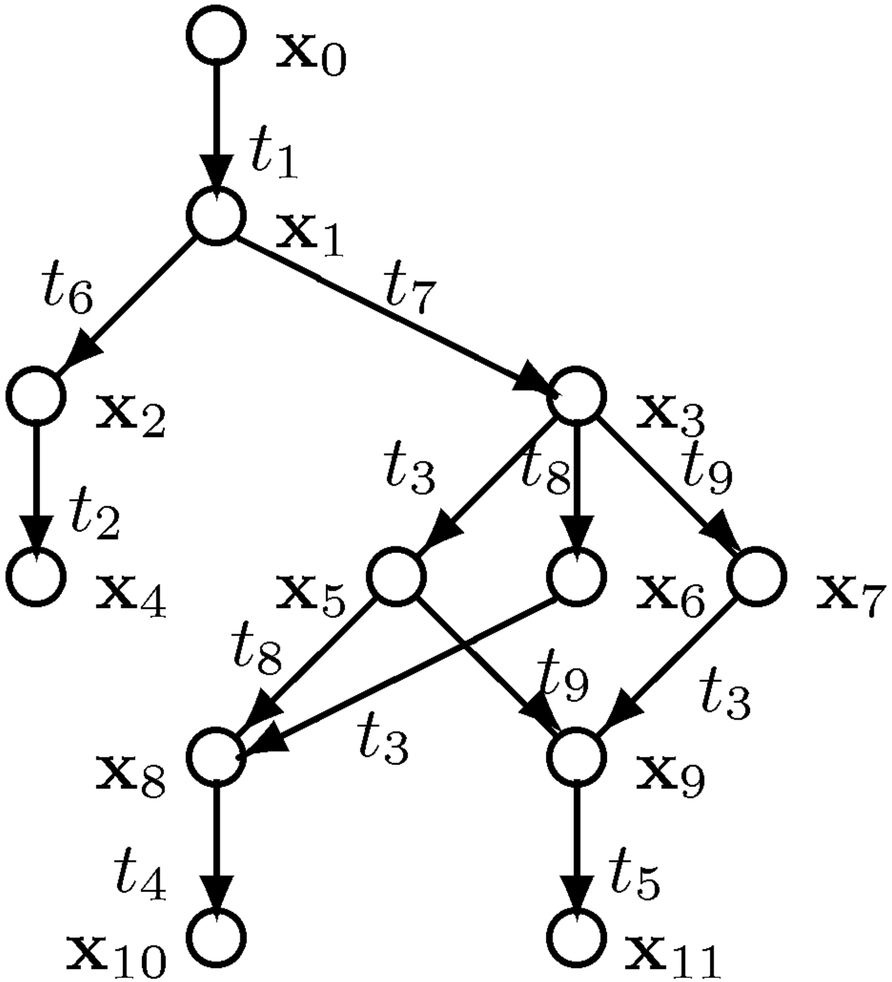
t1 – t9 represent discrete events realizing the system dynamics

PN-based model



Enumerated RT

$$\mathbf{A}_k = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6 & 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 & 8 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 & 9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$



$\mathbf{x}_0$
t_1
$\mathbf{x}_1$
t_6
t_7
$\mathbf{x}_2$
$\mathbf{x}_3$
t_2
t_3
t_8
t_9
$\mathbf{x}_4$
$\mathbf{x}_5$
$\mathbf{x}_6$
$\mathbf{x}_7$
t_8
t_9
t_3
t_3
$\mathbf{x}_8$
$\mathbf{x}_9$
t_4
t_5
$\mathbf{x}_{10}$
$\mathbf{x}_{11}$

The space of reachable states

$$\mathbf{X}_{reach} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Control synthesis

The initial state

$$\mathbf{x}_0 = (1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0)^T$$

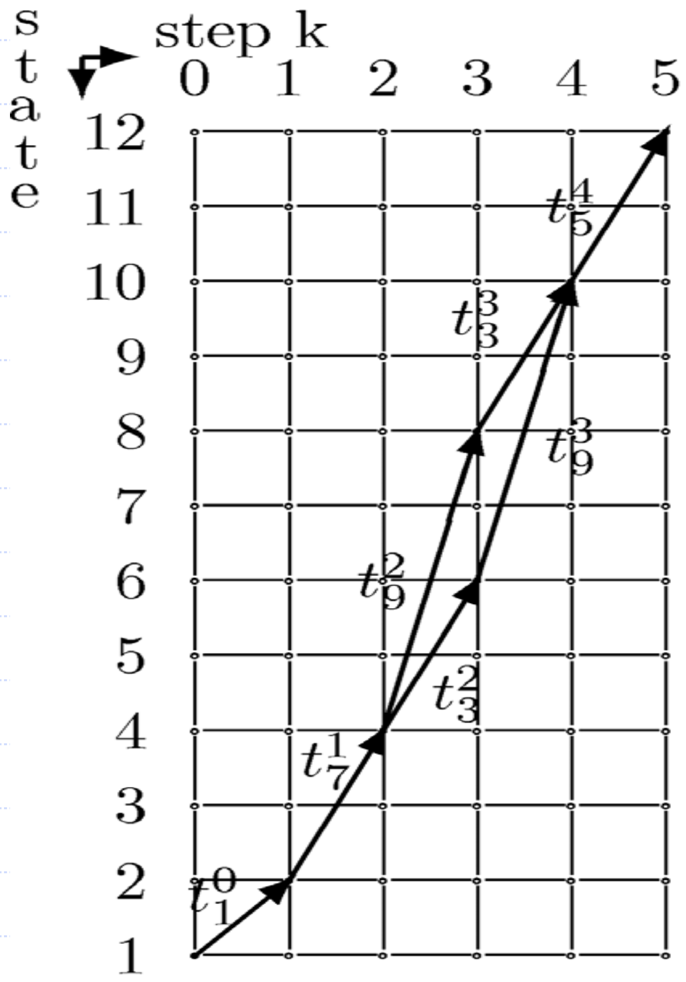
The terminal state – the successful cooperation

$$\mathbf{x}_N = (0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1)^T$$

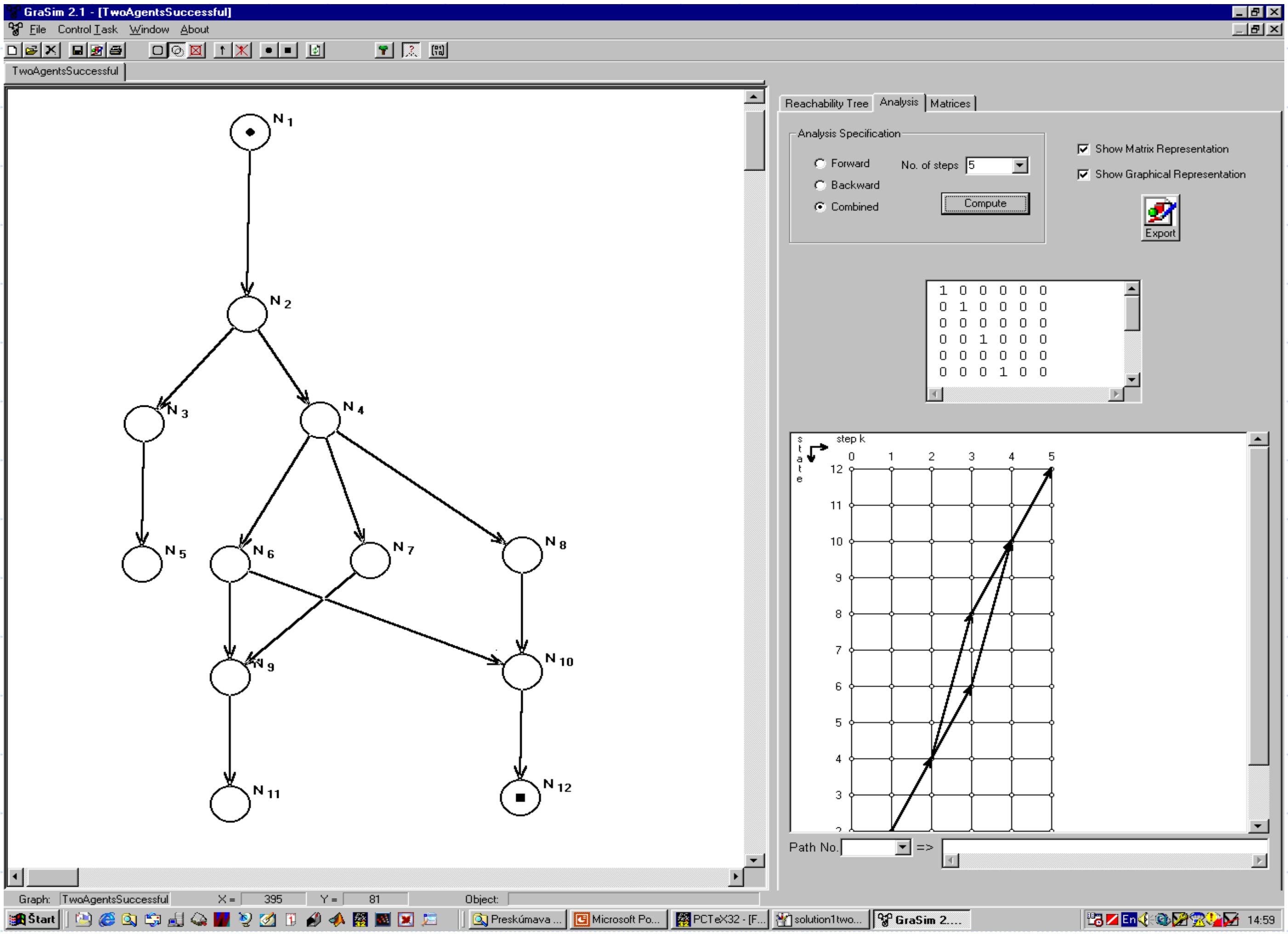
The intersection of the SLRT and BTRT

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

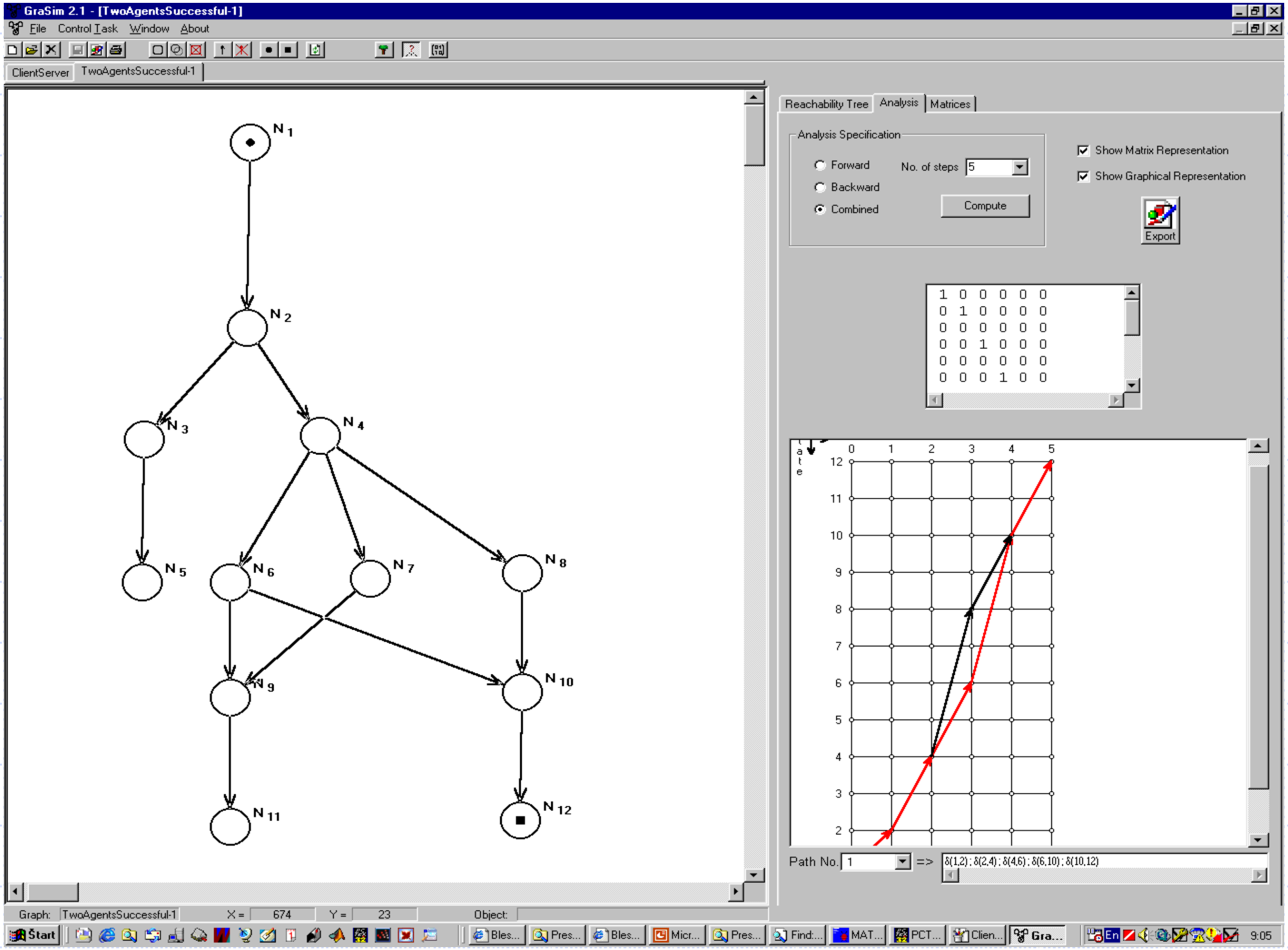
The state trajectories – the successful cooperation



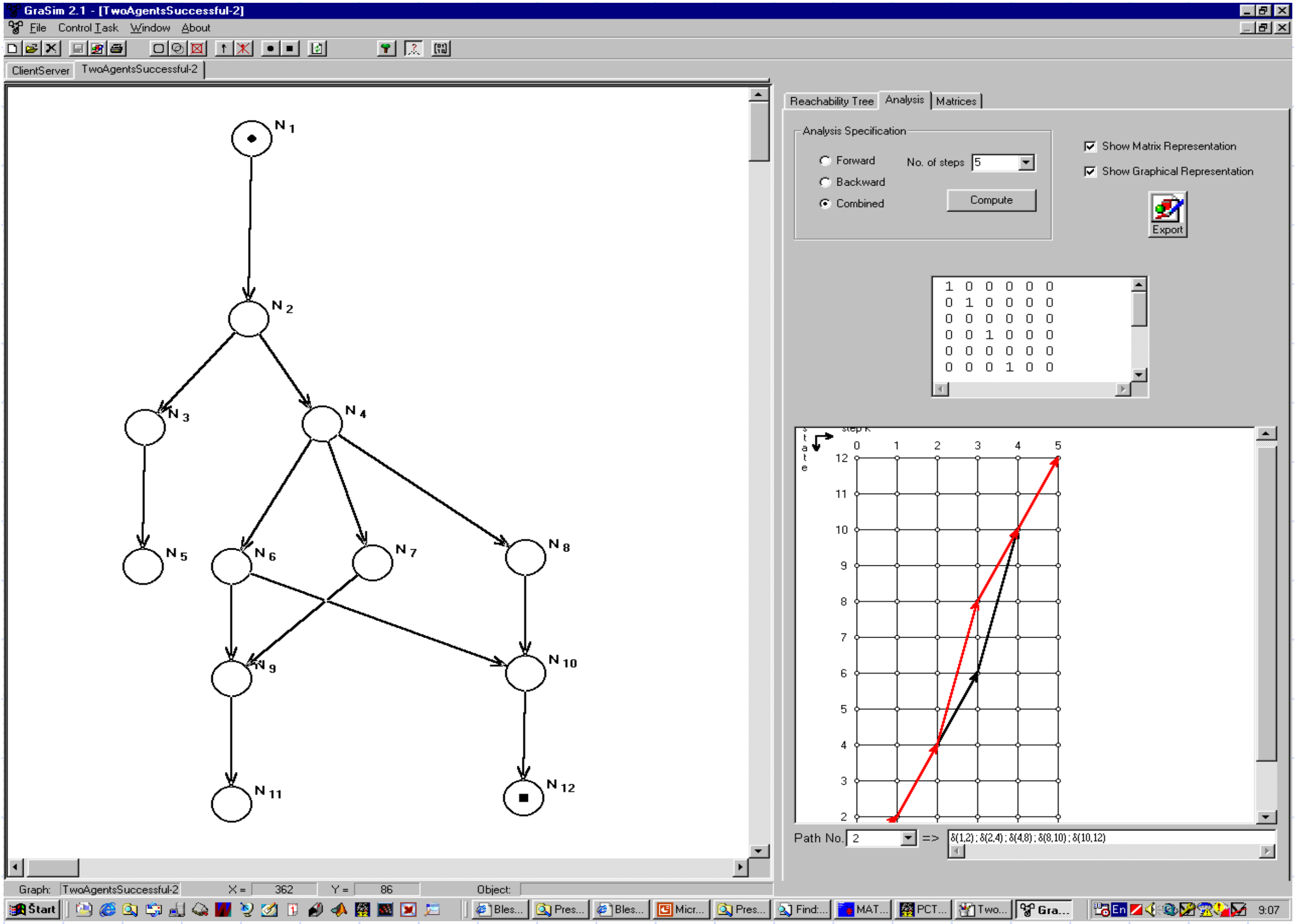
Graphic tool - GraSim



Succesfull cooperation 1

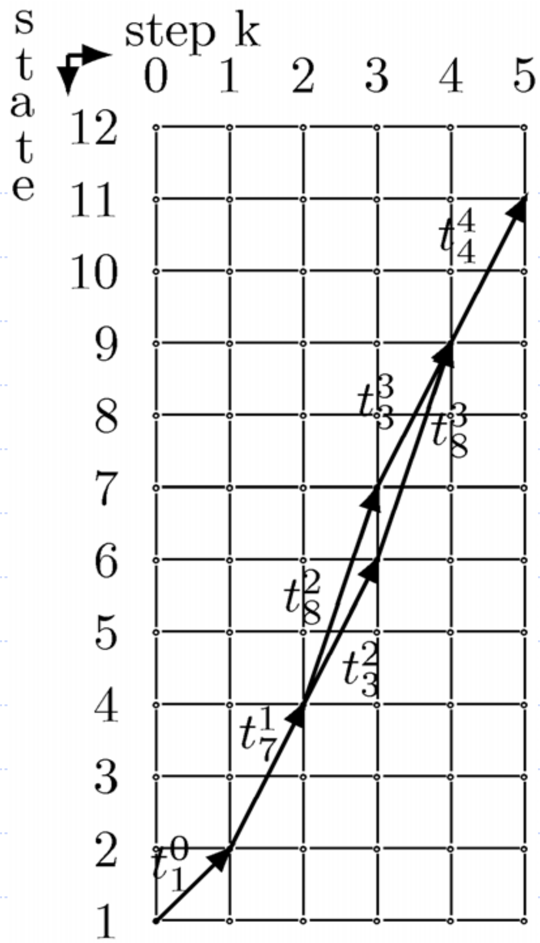


Succesfull cooperation 2



The failured cooperation – when B is not able to do P

$$\mathbf{x}_N = (0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1)^T$$

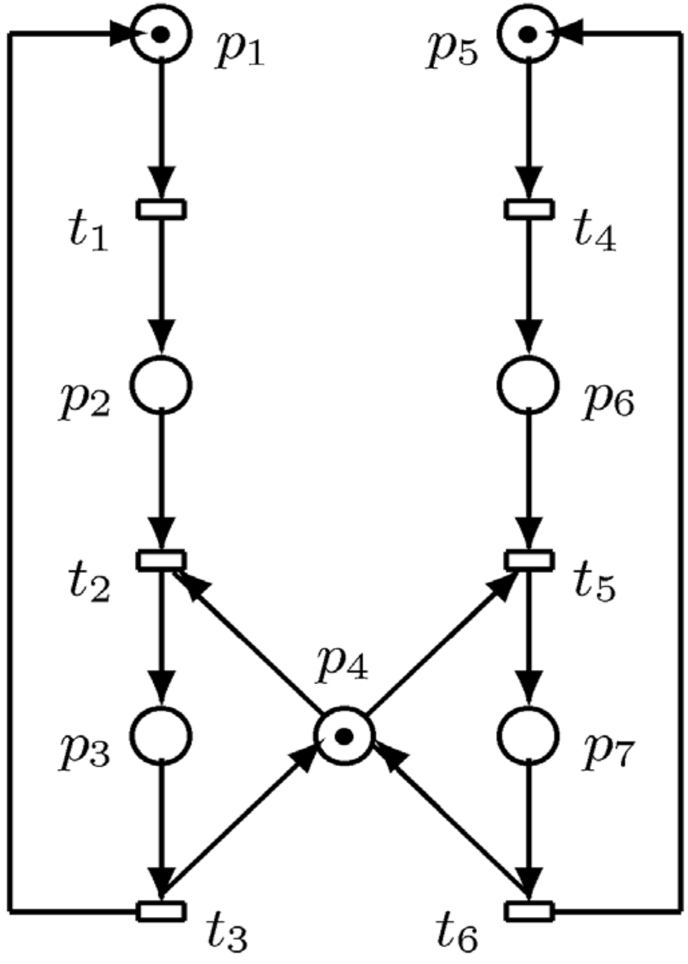


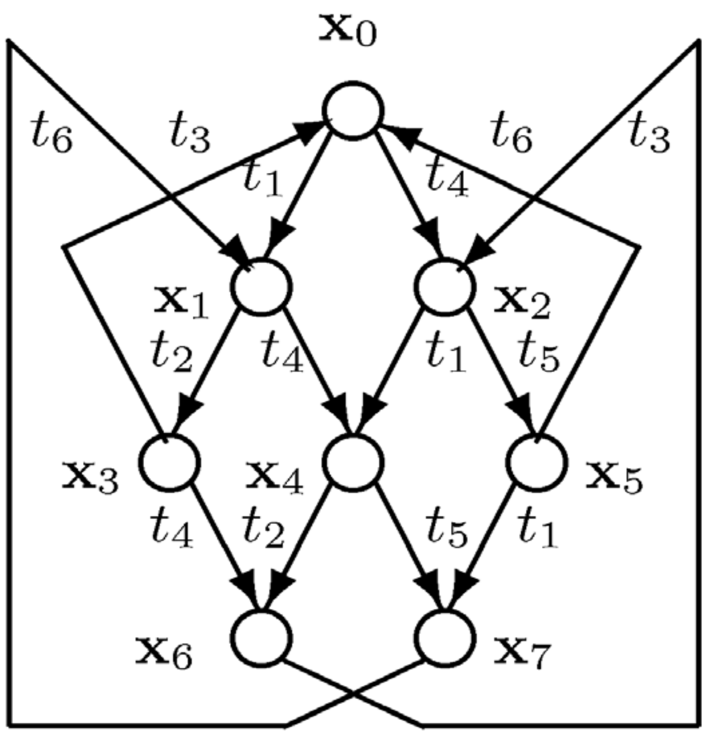
Adaptivity

There are two kinds of the adaptivity in the DEDS control synthesis

- Choosing the most suitable trajectory from the feasible ones in order to adapt the system behaviour to external demands (conditions)
- Changing the structure of the system model in order to express more kinds of the system behaviour.
  - Choosing the most suitable behaviour from the feasible ones. It is illustrated in the next example.

Example 3 – Two processes





{p1, p2, p3} – 1st process ***P1***
{p5, p6, p7} – 2nd process ***P2***

p4 – the structural element that is able to influence

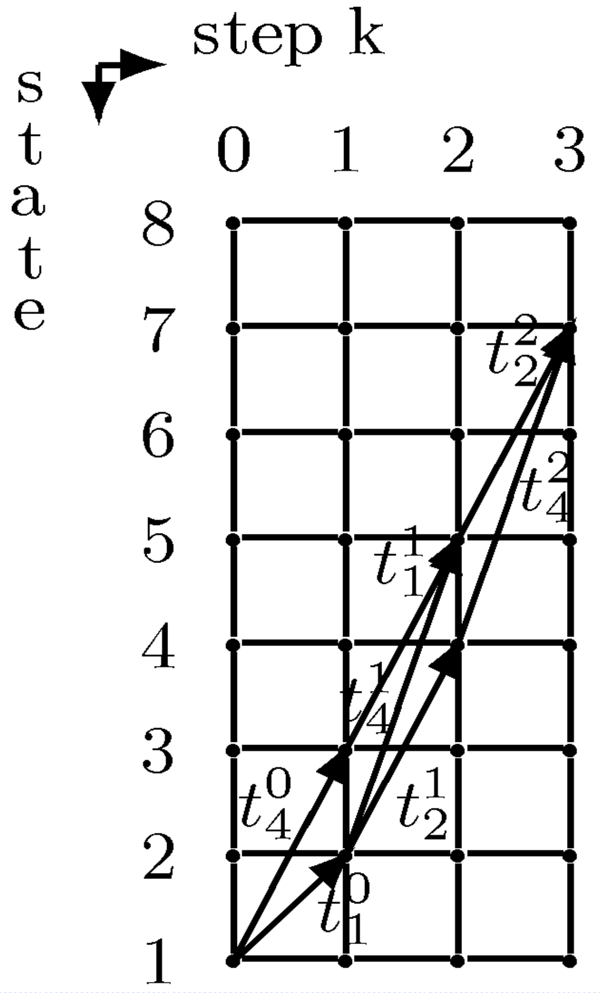
- the mutual exclusion of ***P1*** and ***P2***
- the sequencing of ***P1*** and ***P2***
- the re-running of ***P1*** and ***P2***

$$\mathbf{x}_0 = (1,\ 0,\ 0,\ 1,\ 1,\ 0,\ 0)^T \qquad \mathbf{x}_6 = (0,\ 0,\ 1,\ 0,\ 0,\ 1,\ 0)^T$$

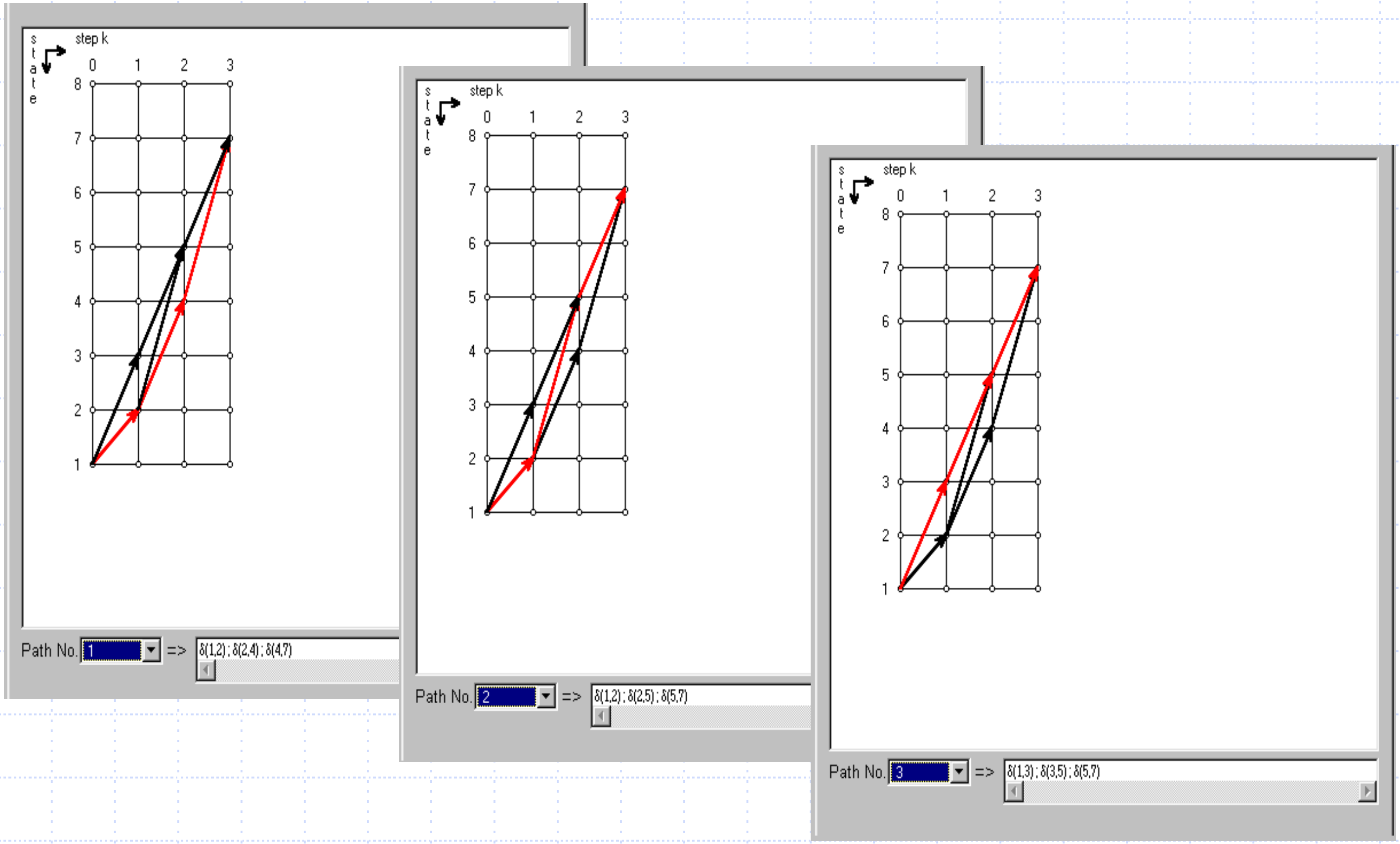
$$\mathbf{A}_k = \begin{pmatrix} 0 & 1 & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 5 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 5 \\ 6 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{X}_{reach} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Exclusion of the process *P2*



Three possibilities of the *P2* exclusion



Conclusions

- Simple general method of DEDS modelling and control synthesis was presented
- Its applicability to the special communication systems (client-server cooperation, two agents cooperation) was demonstrated
- Two kinds of adaptivity were described and illustrated

Future work on this way

- To innovate the method permanently in order to extend its reasonable applicability for larger and larger class of DEDS able to be modelled by Petri nets
- To find new simulation procedures and tools