Fuzzy knowledge in control of manufacturing systems

František Čapkovič
Institute of Control Theory and Robotics, Slovak Academy of Sciences
Dúbravská cesta 9, 842 37 Bratislava, Slovak Republic
Phone: +421-7-3782544, Fax: +421-7-376045, E-mail: utrrcapk@nic.savba.sk
http://www.savba.sk/~utrrcapk/capkhome.htm

Abstract

An approach to the knowledge-based control of manufacturing systems as a kind of discrete-event dynamic systems (DEDS) is presented in this paper. Petri nets (PNs) of different kinds - ordinary (OPNs), logical (LPNs), fuzzy (FPNs) - are used to express analytically both the model of the system to be controlled and to represent knowledge about the control task specifications (like criteria, constraints, etc.). The OPNs yield the analytical model of the DEDS in the form of a linear discrete dynamical system. The analytical form of the knowledge representation - i.e. the knowledge base (KB) - is obtained by means of the LPNs or/and FPNs in the form of a linear discrete logic system. Both the model of the DEDS and the KB are simultaneously used in the procedure of the control system synthesis. The elementary control possibilities are generated (by means of the system model) in any step of the procedure. Then, they are tested with respect to the realization condition. When there are several possibilities satisfying that condition the most suitable (i.e. optimal) control possibility is chosen by means of the KB.

Keywords: Control system synthesis, discrete-event dynamic systems, manufacturing systems, knowledge representation, Petri nets.

1 Introduction

Manufacturing systems (MS) or/and flexible manufacturing systems (FMS) are a kind of DEDS. Also transport systems, communication systems, etc. belong to DEDS. Such systems are very important in human practice. Therefore, the problem of the successful automatic control of them is very actual. Because the control task specifications (like constraints, criteria, etc.) are usualy given in nonanalytical terms (also fuzzy) a suitable knowledge-based approach is chosen to master the problem of the DEDS control synthesis. Even, such an approach is necessary with respect to the concurrency of devices leading to conflict situations. On one hand the concurrency can be expressed in the system model (e.g. PN-based one), however, on the other hand the system itself is not able to solve the conflicts. A suitable knowledge representation is necessary for it.

2 The PN-based k-invariant model of the system

Let us understand the PNs in the sense of [3]. The simplest form of the OPN-based model of the DEDS in analytical terms is the following

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{B}.\mathbf{u}_k\,, \quad k = 0, N \tag{1}$$

$$\mathbf{B} = \mathbf{G}^T - \mathbf{F} \tag{2}$$

$$\mathbf{F}.\mathbf{u}_k \leq \mathbf{x}_k \tag{3}$$

where

k is the discrete step of the system dynamics development.

$\mathbf{x}_k = (x_1^k, ..., x_n^k)^T$ is the n-dimensional state vector of the system in the step k. Its components x_i^k, $i = 1, n$, express the states of the DEDS elementary subprocesses. They acquire their values from the set $\{0, 1\}$ where 0 expresses the passivity and 1 expresses the activity of the corresponding subprocess.

$\mathbf{u}_k = (u_1^k, ..., u_m^k)^T$ is the m-dimensional control vector of the system in the step k. Its components u_j^k, $j = 1, m$, represent the states of occurring the DEDS elementary discrete events (e.g. starting or ending the elementary subprocesses or other activities). They acquire their values from the set $\{0,1\}$ where 1 expresses the presence and 0 expresses the absence of the corresponding discrete event.

B, **F**, **G** are, respectively, $(n \times m)$, $(n \times m)$ and $(m \times n)$- dimensional structural matrices of constant elements. The matrices **F**, **G** are the incidence matrices (in the analogy with the incidence matrices of the mutual oriented interconnections among the PNs positions and transitions) expressing the mutual causal relations among the DEDS subprocesses and the discrete events. The incidence matrix **F** expresses the causal relations oriented from the states of the DEDS subprocesses to the discrete events occuring during the DEDS operation. The incidence matrix **G** expresses the causal relation oriented from the discrete events to the states of the DEDS subprocesses. The elements of these matrices acquire their values from the set $\{0,1\}$ where 1 expresses the existence and 0 expresses the nonexistence of the corresponding causal relation.

$(.)^T$ symbolizes the matrix or vector transposition.

3 The k-variant model of the DEDS based on oriented graphs

Oriented graphs (OGs) can be formally expressed as

$$\langle P, \Delta \rangle \tag{4}$$

where

$P = \{p_1, ..., p_n\}$ is a finite set of the OG nodes with p_i, $i = 1, n$, being the elementary nodes.

$\Delta \subseteq P \times P$ is a set of the OG edges i.e. the oriented arcs among the nodes. It can be expressed by the incidence matrix $\mathbf{\Delta} = \{\delta_{ij}\}$, $\delta_{ij} \in \{0,1\}$, $i = 1, n$; $j = 1, n$. Its element δ_{ij} represents the absence (when 0) or presence (when 1) of the edge oriented from the node p_i to the node p_j.

In order to have an opportunity to combine both the PN-based model and the OG-based one the OG nodes should be equivalent with the PN positions. The OG oriented edges should express information about the DEDS spontaneous discrete events in a suitable form. Therefore, the oriented edges should be properly weighted. To have an opportunity to combine such an approach with the PN-based one the weights should express the actual states of the PN transitions. In other words, there is only one difference between the PN structure and the OG one. The PN transitions are fixed on the oriented edges between corresponding nodes (the PN positions) in the OG structure. Formally, the set $\Delta \subseteq (P \times T) \times (T \times P)$.

To introduce exactly the weights δ_{ij}, $i = 1, n$; $j = 1, n$, the OG dynamics can be formally expressed (in analogy with the above PN-based approach) as follows

$$\langle X, \delta_1, x_0 \rangle \tag{5}$$

where

$X = \{\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_N\}$ is a finite set (practically the same like in PNs) of the state vectors of the graph nodes in different situations with $\mathbf{x}_k = (\sigma^k_{p_1}, ..., \sigma^k_{p_n})^T$, $k = 0, N$, being the n-dimensional state vector of the graph nodes in the step k; $\sigma^k_{p_i} \in \{0,1\}$, $i = 1, n$ is the state of the elementary node p_i in the step k (1 - activity 0 - passivity); k is the discrete step of the graph dynamics development.

$\delta_1 : (X \times U) \times (U \times X) \longmapsto X$ is a transition function of the graph dynamics. It contains implicitly the states of the transitions (the set U is practically the same like in PNs) situated on the OG edges.

$\mathbf{x}_0$ is the initial state vector of the graph dynamics.

Hence, the k-variant OG-based linear discrete dynamic model of the DEDS can be written as follows

$$\mathbf{x}_{k+1} = \mathbf{\Delta}_k . \mathbf{x}_k \quad , \quad k = 0, N \tag{6}$$

where

k is the discrete step of the DEDS dynamics development.

$\mathbf{x}_k = (\sigma^k_{p_1}, ..., \sigma^k_{p_n})^T$; $k = 0, N$ is the n- dimensional state vector of the DEDS in the step k; $\sigma^k_{p_i}$, $i = 1, n$ is the state of the elementary subprocess p_i in the step k. Its activity is expressed by 1 and its passivity by 0 (in the PN analogy it is the state of the elementary position).

$\mathbf{\Delta}_k = \{\delta^k_{ij}\}$, $\delta^k_{ij} = \gamma^k_{t_{p_i|p_j}} \in \{0, 1\}$, $i = 1, n$; $j = 1, n$, because the set Δ can be understood to be in the form $\Delta_k \subseteq (X \times U) \times (U \times X)$. This matrix expresses the causal relations between the subprocesses depending on the occurrence of the discrete events. The element $\delta^k_{ij} = \gamma^k_{t_{p_i|p_j}} \in \{0, 1\}$ expresses the actual value of the transition function of the PN transition fixed on the OG edge oriented from the node p_j to the node p_i. It is based upon understanding the PN transitions to be fixed parts of the oriented arcs among the PN positions - see Fig. 1 - very analogically to the knowledge representation in [3].

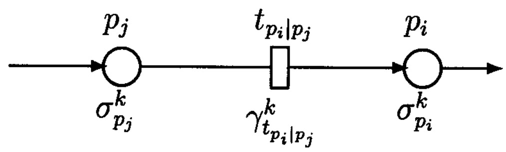

Figure 1: An example of the placement of a transition on the oriented arc between two positions p_i and p_j

The dynamical development of the k-variant model is the following

$$
\begin{aligned}
\mathbf{x}_{k+1} &= \mathbf{\Delta}_k.\mathbf{x}_k\ , \quad k = 0, N & (7)\\
\mathbf{x}_1 &= \mathbf{\Delta}_0.\mathbf{x}_0 & (8)\\
\mathbf{x}_2 &= \mathbf{\Delta}_1.\mathbf{x}_1 = \mathbf{\Delta}_1.\mathbf{\Delta}_0.\mathbf{x}_0 & (9)\\
\vdots \quad & \vdots \quad \vdots \quad \vdots & \\
\mathbf{x}_k &= \mathbf{\Delta}_{k-1}.\mathbf{x}_{k-1} = \mathbf{\Delta}_{k-1}.\mathbf{\Delta}_{k-2}.\dots.\mathbf{\Delta}_1.\mathbf{\Delta}_0.\mathbf{x}_0 & (10)\\
\mathbf{x}_k &= \mathbf{\Phi}_{k,0}.\mathbf{x}_0 & (11)\\
\mathbf{\Phi}_{k,j} &= \prod_{i=j}^{k-1} \mathbf{\Delta}_i \quad ; \quad j = 0, k-1 & (12)
\end{aligned}
$$

The multiplying is made from the left. It must be said that the meaning of the multiplying and additioning operators in the development of the k-variant model have symbolic interpretation. For example, an element $\phi^{k,0}_{i,j}$, $i = 1, n$; $j = 1, n$ of the transition matrix $\mathbf{\Phi}_{k,0}$ is either a product of k elements (the transition functions expressing the "trajectory" containing the sequence of elementary transitions that must be fired in order to go from the initial elementary state x^0_j into the final state x^k_i) or a sum of several such product (when there are several "trajectories" from the initial state to final one).

4 The analysis of the system to be controlled

The k-variant model of DEDS presented above is able to give us the system dynamics development in analytical terms. Let us introduce an illustrative example concerning the PN-based and OG-based modelling DEDS.

4.1 An example of the DEDS

Let us demonstrate the approach on the maze problem introduced by Ramadge and Wonham [5]. It is the typical example for many kinds of DEDS. Namely, two "participants" - in [5] a cat and a mouse - can be as well e.g. two mobile robots or two automatically guided vehicles (AGVs) of the FMS, two cars on a complicated crossroad, two trains in a railway network, etc. They are placed in the maze (however, it can also be e.g. the complicated crossroad, etc.) given on Fig. 2 consisting of five rooms denoted by numbers 1, 2,..., 5 connecting by the doorways exclusively for the cat denoted by $c_i, i = 1, 7$ and the doorways exclusively for the mouse denoted by $m_j, j = 1, 6$ (the doors can represent also the point in the railway network, the cross lights on the crossroad, etc.). The first of the participants (the cat) is initially in the room 3 and the second one (the mouse) in room 5. Each doorway can be traversed only in the direction indicated. Each door (with the exception of the door c_7) can be opened or closed by means of control actions. The door c_7 is uncontrollable (or better, it is continuously open in both directions). The controller to be synthetized observes only discrete events generated by sensors in the doors. They indicate that a participant ist just running through. The control problem is to find a feedback controller (or e.g. an automatic pointsman or switchman in railways, automatic cross lights in the crossroad, etc.) such that the following three constraints will be satisfied:

1. The participants never occupy the same room simultaneously.

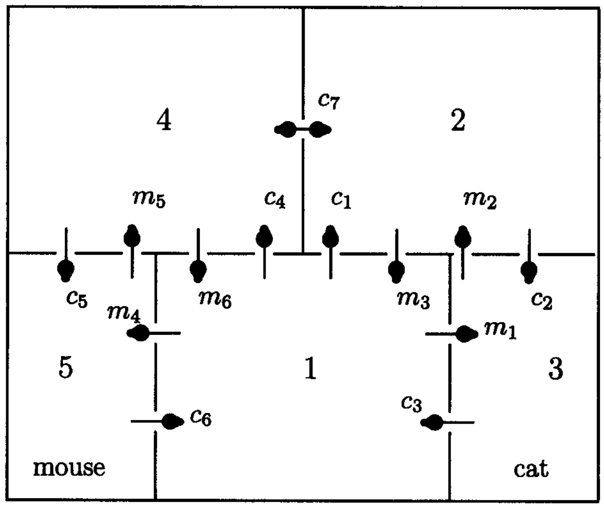

Figure 2: The maze structure.

2. It is always possible for both of them to return to their initial positions (the first one to the room 3 and the second one to the room 5).
3. The controller should enable the participants to behave as freely as possible with respect to the constraints imposed.

At the construction of the PN-based model of the system the rooms 1 - 5 of the maze will be represented by the PN positions p_1 - p_5 and the doorways will be represented by the PN transitions. The permanently open door c_7 is replaced by means of two PN transitions t_7 and t_8 symbolically denoted as c_7^k and c_8^k.

The PN-based models: The PN-based representation of the maze is given on Fig. 3. The initial state vectors of the cat and the mouse are

$$ {}^{c}\mathbf{x}_0 = (0\,0\,1\,0\,0)\ ,\ {}^{m}\mathbf{x}_0 = (0\,0\,0\,0\,1)^T \tag{13} $$

The structure of the cat and mouse control vectors is

$$ \begin{aligned} {}^{c}\mathbf{u}_k &= (c_1^k,\ c_2^k,\ c_3^k,\ c_4^k,\ c_5^k,\ c_6^k,\ c_7^k,\ c_8^k)^T \qquad (14) \\ c_i^k &\in \{0,1\},\ \ i = 1,8 \\ {}^{m}\mathbf{u}_k &= (m_1^k,\ m_2^k,\ m_3^k,\ m_4^k,\ m_5^k,\ m_6^k)^T \qquad (15) \\ m_i^k &\in \{0,1\},\ \ i = 1,6 \end{aligned} $$

The parameters of the cat model are
$n = 5 \qquad m_c = 8$

$$ \mathbf{F}_c = \begin{pmatrix} 1\,0\,0\,1\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0\,1\,0 \\ 0\,0\,1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0\,0\,1 \\ 0\,0\,0\,0\,0\,1\,0\,0 \end{pmatrix} \mathbf{G}_c = \begin{pmatrix} 0\,1\,0\,0\,0 \\ 0\,0\,1\,0\,0 \\ 1\,0\,0\,0\,0 \\ 0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,1 \\ 1\,0\,0\,0\,0 \\ 0\,0\,0\,1\,0 \\ 0\,1\,0\,0\,0 \end{pmatrix} $$

and the parameters of the mouse model are

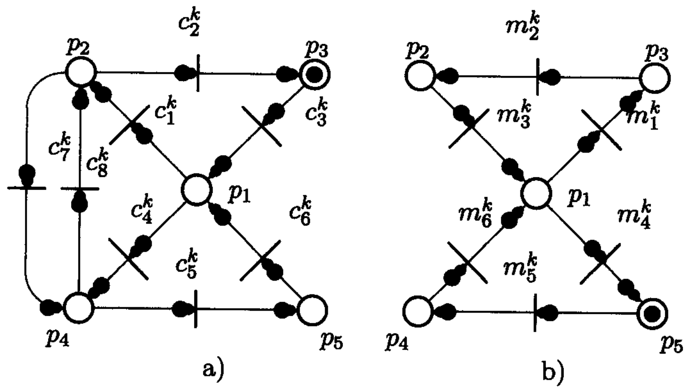

Figure 3: The PN-based representation of the maze. a) possible behaviour of the cat; b) possible behaviour of the mouse

$n = 5 \qquad m_m = 6$

$$\mathbf{F}_m = \begin{pmatrix} 1\,0\,0\,1\,0\,0 \\ 0\,0\,1\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,1\,0 \end{pmatrix} \quad \mathbf{G}_m^T = \begin{pmatrix} 0\,0\,1\,0\,0\,1 \\ 0\,1\,0\,0\,0\,0 \\ 1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,1\,0\,0 \end{pmatrix}$$

The OG-based models: At the construction of the OG-based model the matrices ${}^c\mathbf{\Delta}_k$ and ${}^m\mathbf{\Delta}_k$ of the system parameters are the following

$${}^c\mathbf{\Delta}_k = \begin{pmatrix} 0 & 0 & c_3^k & 0 & c_6^k \\ c_1^k & 0 & 0 & c_8^k & 0 \\ 0 & c_2^k & 0 & 0 & 0 \\ c_4^k & c_7^k & 0 & 0 & 0 \\ 0 & 0 & 0 & c_5^k & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & {}^c\delta_{13}^k & 0 & {}^c\delta_{15}^k \\ {}^c\delta_{21}^k & 0 & 0 & {}^c\delta_{24}^k & 0 \\ 0 & {}^c\delta_{32}^k & 0 & 0 & 0 \\ {}^c\delta_{41}^k & {}^c\delta_{42}^k & 0 & 0 & 0 \\ 0 & 0 & 0 & {}^c\delta_{54}^k & 0 \end{pmatrix}$$

$${}^m\mathbf{\Delta}_k = \begin{pmatrix} 0 & m_3^k & 0 & m_6^k & 0 \\ 0 & 0 & m_2^k & 0 & 0 \\ m_1^k & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & m_5^k \\ m_4^k & 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & {}^m\delta_{12}^k & 0 & {}^m\delta_{14}^k & 0 \\ 0 & 0 & {}^m\delta_{23}^k & 0 & 0 \\ {}^m\delta_{31}^k & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & {}^m\delta_{45}^k \\ {}^m\delta_{51}^k & 0 & 0 & 0 & 0 \end{pmatrix}$$

The system dynamics development: The transitions matrices for the cat and mouse are the following

$${}^c\mathbf{\Phi}_{k+2,k} = {}^c\mathbf{\Delta}_{k+1}.{}^c\mathbf{\Delta}_k =$$

$$= \begin{pmatrix} 0 & c_3^{k+1}.c_2^k & 0 & c_6^{k+1}.c_5^k & 0 \\ c_8^{k+1}.c_4^k & c_8^{k+1}.c_7^k & c_1^{k+1}.c_3^k & 0 & c_1^{k+1}.c_6^k \\ c_2^{k+1}.c_1^k & 0 & 0 & c_2^{k+1}.c_8^k & 0 \\ c_7^{k+1}.c_1^k & 0 & c_4^{k+1}.c_3^k & c_7^{k+1}.c_8^k & c_4^{k+1}.c_6^k \\ c_5^{k+1}.c_4^k & c_5^{k+1}.c_7^k & 0 & 0 & 0 \end{pmatrix}$$

$${}^c\mathbf{\Phi}_{k+3,k} = {}^c\mathbf{\Delta}_{k+2}.{}^c\mathbf{\Delta}_{k+1}.{}^c\mathbf{\Delta}_k =$$

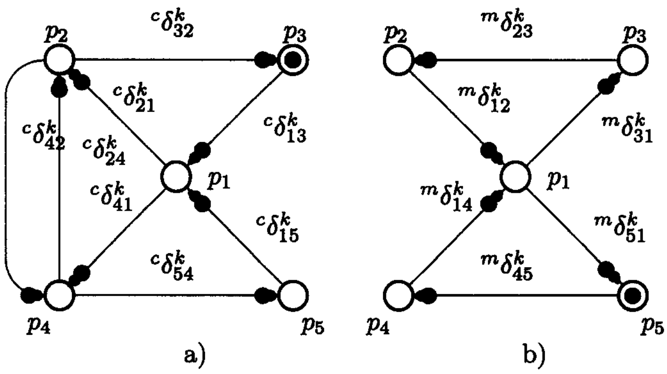

Figure 4: The OG-based model of the maze. a) possible behaviour of the cat; b) possible behaviour of the mouse

$$
= \left(\begin{array}{ccc}
c_3^{k+2}.c_2^{k+1}.c_1^k + c_6^{k+2}.c_5^{k+1}.c_4^k & c_6^{k+2}.c_5^{k+1}.c_7^k & \vdots \\
c_8^{k+2}.c_7^{k+1}.c_1^k & c_1^{k+2}.c_3^{k+1}.c_2^k & \vdots \\
c_2^{k+2}.c_8^{k+1}.c_4^k & c_2^{k+2}.c_8^{k+1}.c_7^k & \vdots \\
c_7^{k+2}.c_8^{k+1}.c_4^k & c_4^{k+2}.c_3^{k+1}.c_2^k + c_7^{k+2}.c_8^{k+1}.c_7^k & \vdots \\
c_5^{k+2}.c_7^{k+1}.c_1^k & 0 & \vdots
\end{array} \right.
$$

$$
\left. \begin{array}{cccc}
\vdots & 0 & c_3^{k+2}.c_2^{k+1}.c_8^k & 0 \\
\vdots & c_8^{k+2}.c_4^{k+1}.c_3^k & c_1^{k+2}.c_6^{k+1}.c_5^k + c_8^{k+2}.c_2^{k+1}.c_8^k & c_8^{k+2}.c_4^{k+1}.c_6^k \\
\vdots & \boxed{c_2^{k+2}.c_1^{k+1}.c_3^k} & 0 & c_2^{k+2}.c_1^{k+1}.c_6^k \\
\vdots & c_7^{k+2}.c_1^{k+1}.c_3^k & c_4^{k+2}.c_6^{k+1}.c_5^k & c_7^{k+2}.c_1^{k+1}.c_6^k \\
\vdots & c_5^{k+2}.c_4^{k+1}.c_3^k & c_5^{k+2}.c_7^{k+1}.c_8^k & \boxed{c_5^{k+2}.c_4^{k+1}.c_6^k}
\end{array} \right)
$$

$$
{}^m\mathbf{\Phi}_{k+2,k} = {}^m\mathbf{\Delta}_{k+1}.{}^m\mathbf{\Delta}_k =
$$

$$
= \begin{pmatrix}
0 & 0 & m_3^{k+1}.m_2^k & & m_6^{k+1}.m_5^k \\
m_2^{k+1}.m_1^k & 0 & 0 & 0 & 0 \\
0 & m_1^{k+1}.m_3^k & 0 & m_1^{k+1}.m_6^k & 0 \\
m_5^{k+1}.m_4^k & 0 & 0 & 0 & 0 \\
0 & m_4^{k+1}.m_3^k & 0 & m_4^{k+1}.m_6^k & 0
\end{pmatrix}
$$

$$
{}^m\mathbf{\Phi}_{k+3,k} = {}^m\mathbf{\Delta}_{k+2}.{}^m\mathbf{\Delta}_{k+1}.{}^m\mathbf{\Delta}_k =
$$

$$
= \left(\begin{array}{ccc}
m_3^{k+2}.m_2^{k+1}.m_1^k + m_6^{k+2}.m_5^{k+1}.m_4^k & 0 & \vdots \\
0 & m_2^{k+2}.m_1^{k+1}.m_3^k & \vdots \\
0 & 0 & \vdots \\
0 & m_5^{k+2}.m_4^{k+1}m_3^k & \vdots \\
0 & 0 & \vdots
\end{array} \right.
$$

$$
\begin{array}{c|ccc}
\vdots & 0 & 0 & 0 \\
\vdots & 0 & m_2^{k+2}.m_1^{k+1}.m_6^k & 0 \\
\vdots & \boxed{m_1^{k+2}.m_3^{k+1}.m_2^k} & 0 & m_1^{k+2}.m_6^{k+1}.m_5^k \\
\vdots & 0 & m_5^{k+2}.m_4^{k+1}.m_6^k & 0 \\
\vdots & m_4^{k+2}.m_3^{k+1}.m_2^k & 0 & \boxed{m_4^{k+2}.m_6^{k+1}.m_5^k}
\end{array}
\right)
$$

The system analysis: The states reachability trees are the following

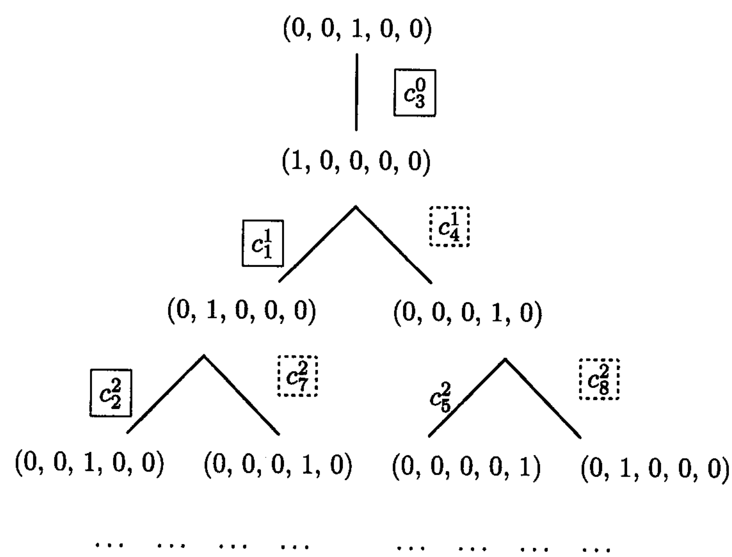

Figure 5: The fragment of the reachability tree of the cat

It can be seen that in order to fulfille the prescribed control task specifications introduced in the part 4.1 the comparing of the transition matrices of both animals in any step of their dynamics development is sufficient. Because the animals start from the defined rooms given by their initial states, it is sufficient to compare the columns 3 and 5. Consequently,

1. the corresponding (as to indices) elements of the transition matrices for the cat and mouse have to be in these columns mutually disjuct in any step of the dynamics development in order to avoid their meeting on the coresponding "trajectories".

2. if they are not disjunct they must be removed. Only elements with indices 3,3 and 5,5 of the matrices $\mathbf{\Phi}_{k+3,0}$ represent the exception, because they express the trajectories of returning the animals to their initial states. In case of the elements with indices 3,3 the element of the matrix ${}^c\mathbf{\Phi}_{k+3,0}$ should be chosen, because it represent the trajectory of the cat making their come back possible. In case of the elements with indices 5,5 the element of the matrix ${}^m\mathbf{\Phi}_{k+3,0}$ should be chosen, because it represent the trajectory of the mouse making their come back possible

3. in the matrix ${}^c\mathbf{\Phi}_{k+3,k}$ two elements in the column 3 (with the indices 2,3 and 4,3) stay unremoved, because of the permanently open door. It corresponds to the prescribed condition that othervise the movement of the animals in the maze should be free.

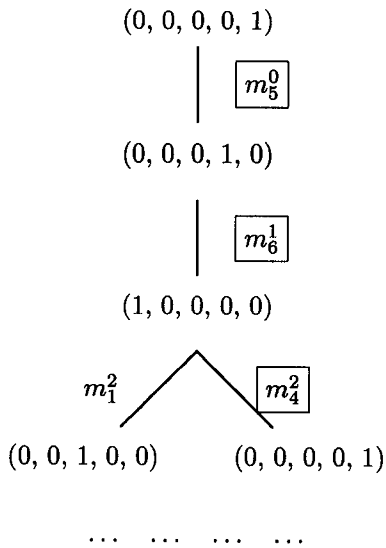

Figure 6: The reachability tree of the mouse

4.2 The conclusion of the analysis

What is the advantage of such a solution is that the complete solution is found on such a way in the elegant form, even in analytical terms. However, such an approach cannot be used for more complicated or large-scale systems. Especially, the exponencial state explosion is very inconvenient. It leads to the term '*curse of dimensionality*' (defined by R. Bellman at the definition of his dynamic programming method). Hence, another approach has to be found in order to automate solving the problems of the DEDS control synthesis. Consequently, the below introduced approach was proposed in order to avoid the control synthesis problems mentioned above.

5 The idea of the control synthesis

The control synthesis problem is that of finding a sequence of the control vectors $\mathbf{u}_k$, $k = 0, N$ able to convert the state of the system from an initial state $\mathbf{x}_0$ into a terminal (final) state $\mathbf{x}_t$. However as a rule, the DEDS control policy cannot be expressed in analytical terms. Knowledge concerning the control task specifications (e.g. constraints, criteria, and further demands on the system behaviour) is usually expressed only verbally. Even, sometimes it can be fuzzy. Consequently, the proper knowledge representation (e.g. the rule-based one) is needed in form of a domain oriented KB. Usually, there are several different possibilities how to choose the vector $\mathbf{u}_k$ in any step k. The further development of the DEDS dynamics will undoubtedly depend on this choice. Consequently, the proper knowledge representation is expected to be at disposal in order to avoid any ambiguity. The KB is utilized at the choice of the most suitable (optimal) control vector $\mathbf{u}_k$ in any step k in order to avoid any ambiguity as to the further development of the DEDS dynamics.

In order to find in any step k the suitable control vector $\mathbf{u}_k$ able to convert the state of the system from the existing state $\mathbf{x}_k$ into a following state $\mathbf{x}_{k+1}$ the simple procedure can be used. It consists in the following principal steps: finding the control base, generating the elementary control possibilities, and choice of the actual control possibility - i.e. the most suitable (optimal) possibility with respect to control task specifications.

5.1 The control synthesis procedure

The procedure of the control synthesis is schematically illustrated on Fig. 7. Hence, it can be concisely described in the form of the verbal flow chart as follows:

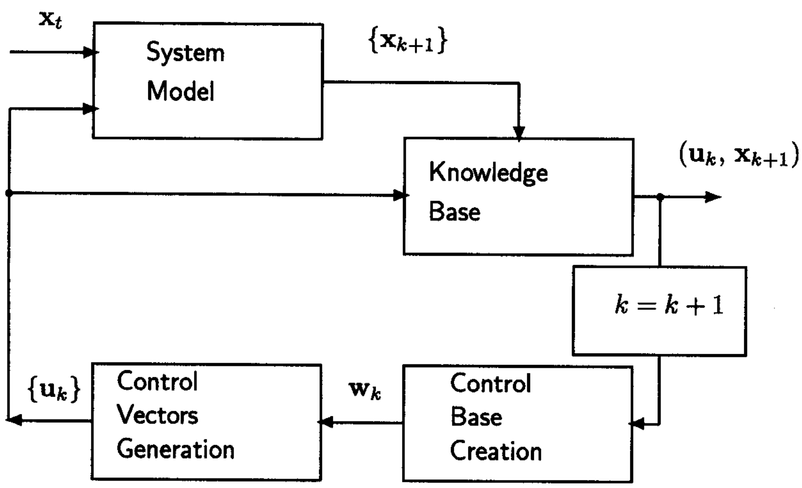

Figure 7: The principial procedure of the control synthesis.

START

- $k = 0$ i.e. $\mathbf{x}_k = \mathbf{x}_0$; $\mathbf{x}_0$ is an initial state; $\mathbf{x}_t$ is a terminal state

LABEL:

- generation of the control base $\mathbf{w}_k$
- generation of the possible control vectors $\mathbf{u}_k \in \mathbf{w}_k$
- generation of the corresponding model responses $\mathbf{x}_{k+1}$
- consideration of the possibilities in the **KB** (built on IF-THEN rules and expressing the control task specifications)
- choice of the most suitable (optimal) control possibility
- *if* (the $\mathbf{x}_t$ or another stable state was found) *then (goto* END) *else (begin* $k = k + 1$; *goto* LABEL; *end)*

END

On Fig. 7 the procedure of the control synthesis is concisely introduced. The vector $\mathbf{w}_k$ is the m-dimensional control base vector.

5.2 Finding the control base vector

In order to mathematize the process of finding the control base in the step k let us perform the following procedure with generating the auxiliary vectors $\mathbf{y}_k$, $\mathbf{v}_k$, $\mathbf{z}_k$

$$\mathbf{x}_k = ({x_1}^k, ..., {x_n}^k)^T \quad (16)$$

$$\mathbf{y}_k = ({y_1}^k, ..., {y_n}^k)^T \; ; \; {y_i}^k = \begin{cases} 1 & \text{if } {x_i}^k > 0 \\ 0 & \text{otherwise} \end{cases} \; ; \; i = 1, n \quad (17)$$

$$\bar{\mathbf{y}}_k = \underline{neg}\, \mathbf{y}_k = \mathbf{1}_n - \mathbf{y}_k \quad (18)$$

$$\mathbf{v}_k = \mathbf{F}^T \underline{and}\, \bar{\mathbf{y}}_k \quad (19)$$

$$\mathbf{v}_k = ({v_1}^k, ..., {v_m}^k)^T \quad (20)$$

$$\mathbf{z}_k = ({z_1}^k, ..., {z_m}^k)^T \; ; \; {z_j}^k = \begin{cases} 1 & \text{if } {v_j}^k > 0 \\ 0 & \text{otherwise} \end{cases} \; ; \; j = 1, m \quad (21)$$

$$\mathbf{w}_k = \underline{neg}\, \mathbf{z}_k = \mathbf{1}_m - \mathbf{z}_k \quad (22)$$
$$\mathbf{w}_k = ({w_1}^k, ..., {w_m}^k)^T \quad (23)$$

where

$\underline{neg}$ is the operator of logical negation.
$\underline{and}$ is the operator of logical multiplying.
$\mathbf{1}_n$ is the n-dimensional constant vector with all of its elements equalled to the integer 1.
$\mathbf{y}_k$ is n-dimensional auxiliary logical vector.
$\mathbf{v}_k$, $\mathbf{z}_k$ are, respectively, m-dimensional auxiliary vector and m-dimensional auxiliary logical vector.
$\mathbf{w}_k$ is m-dimensional vector of the base for the control vector choice (i.e. the control base vector).

To interpret verbally the previous procedure it should be said that the auxiliary vector $\mathbf{y}_k$ represents a logical form of the state vector $\mathbf{x}_k$ (because there can be the nonzero components, having their integer value greater than 1, in the original state vector $\mathbf{x}_k$). The nonzero components of $\mathbf{y}_k$ point out the subprocesses being in the active state in the step k of the system dynamics development. After its logical negation the auxiliary vector $\overline{\mathbf{y}}_k$ is obtained. Its nonzero components point out the subprocesses being in the passive state in the step k. Because the integer components of the auxiliary vector $\mathbf{v}_k$ can be greater than 1, the auxiliary vector $\mathbf{z}_k$ representing a logical form of the vector $\mathbf{v}_k$ is enumerated. Its nonzero components point out the discrete events that cannot be enabled in the step k. The logical negation of $\mathbf{v}_k$ yields the vector $\mathbf{w}_k$. Finally, the nonzero components of $\mathbf{w}_k$ point out the discrete events that can be enabled in the step k and, consequently, can contribute to the system dynamics development. Having generated the control base vector $\mathbf{w}_k$, the information about the control possibilities in the step k is at disposal. However, sorrow, only in an aggregated form. Hence, the vector $\mathbf{w}_k$ is only the vector expressing information for generation of control possibilities in the step k. To obtain the elementary control vectors $\mathbf{u}_k \in \mathbf{w}_k$ a suitable generation procedure is necessary. It is the following:

$$\mathbf{u}_k = ({u_1}^k,\ ...,\ {u_m}^k)^T$$
$$\mathbf{u}_k \subseteq \mathbf{w}_k\ ;\ {u_j}^k = \begin{cases} {w_j}^k & \text{if it is chosen} \\ 0 & \text{otherwise} \end{cases}\ ;\ j = 1, m \quad (24)$$

More details about the choice of the ${w_j}^k$ are given in the following subsection.

5.3 Generating the elementary control possibilities

The control base vector $\mathbf{w}_k$ contains an aggregated form of information about all control possibilities, because its nonzero elements point out the FMS discrete events (i.e. in the metaphorical interpretation the PNs transitions) that could be theoretically enabled in the step k and contribute in such a way to the system dynamics development. However, not always simultaneously. Of course, the elementary possibilities must be generated and tested with respect to the condition (3). Only the possibilities satisfying this condition can be realized. To answer the question why the condition must be tested in any step k, it must be said that the condition (3) prevents taking more tokens from input PNs positions of the enabeled transition in question than there are actualy placed in them in the actual step k. To analyze the control synthesis problem entirely we have to take into account the following possibilities of constructing the control vectors: those containing only single components of the vector $\mathbf{w}_k$, those containing pairs of components of the vector $\mathbf{w}_k$, those containing triplets of components of the vector $\mathbf{w}_k$, those containing quadruplets, etc., and finally, that one containing the full N_t^k-plet of the components of the vector $\mathbf{w}_k$ (i.e. all of them - it is the case when $\mathbf{u}_k = \mathbf{w}_k$).

Theoretically (i.e. from the combinatorics point of view) there exist

$$N_p{}^k = \sum_{i=1}^{N_t{}^k} \binom{N_t{}^k}{i} = 2^{N_t{}^k} - 1 \quad (25)$$

possibilities of the control vector choice in the step k. Here,

$$N_t{}^k = \sum_{j=1}^{m} {w_j}^k. \quad (26)$$

It is the number of nonzero elements of the vector $\mathbf{w}_k$ (i.e. the number of the PNs transitions that can theoretically be enabled in the step k).

5.4 The choice of the actual control vector

The vector $\mathbf{w}_k$ represents the control base because it implicitly expresses the possible candidates for generating the control vector $\mathbf{u}_k$ in the step k. Its nonzero components point out the enabled transitions (when the PNs analogy is used) in the step k, i.e. the possible discrete events which could occur in the DEDS in the step k and which could be utilized in order to convert the state of the system from the present state $\mathbf{x}_k$ into another state $\mathbf{x}_{k+1}$. When only one of the $\mathbf{w}_k$ components is different from zero, it can be used (when the condition (3) is met) to be the actual control vector, i.e. $\mathbf{u}_k = \mathbf{w}_k$. When there are several components of the $\mathbf{w}_k$ different from zero (their number is $N_t{}^k$) several candidates (their number is exactly equal to $N_p{}^k$) for the actual control vector can be generated in the step k. The candidates that do not satisfy the condition (3) are automatically eliminated. In spite of this, there can be more than one candidate satisfying the condition. Such candidates can be devided into two main groups:

1. the *single* candidates containing only *one* component different from zero (i.e. only single enabled transition)
2. the *parallel* candidates containing *more than one* components different from zero (i.e. several transitions enabled simultaneously).

Any single candidate can be chosen to be the alternative control vector without any regard to the fact whether it is in a conflict with other single candidates or not. Hovewer, any single candidate being a subset of a parallel candidate can be used simultaneously with all of the single candidates being also a subset of the same parallel candidate. The simultaneous using such single candidates is equivalent to using the corresponding parallel candidate in question.

5.5 The conclusion of the control synthesis

In order to automate the DEDS control synthesis procedure, knowledge about the control task specification should be expressed in a suitable form. The KB intervention should reflect both the actual state of the system behaviour and the external conditions (EC) expressing the control task specifications. Especially the EC can be verbal or fuzzy.

The form of a simple rule (as a fragment of the KB) can be (in case of N_p control possibilities) e.g. the following: IF $((\mathbf{u}_k^1, \mathbf{x}_{k+1}^1)$ *and* ... *and* $(\mathbf{u}_k^i, \mathbf{x}_{k+1}^i)$ *and* ... *and* $(\mathbf{u}_k^{N_p}, \mathbf{x}_{k+1}^{N_p})$ *and* EC) THEN ($\mathbf{u}_k^i$ corresponding to the EC).

6 The knowledge representation

Consider the knowledge representation in the form introduced in [1] - [4]. Consequently, the KB "dynamics" development - i.e. the KB truth propagation - can be expressed in analytical terms as follows

$$\mathbf{\Phi}_{K+1} = \mathbf{\Phi}_K \ \underline{or}\ \mathbf{\Delta}\ \underline{and}\ \mathbf{\Omega}_K\,, \quad K = 0, N_1 \tag{27}$$

$$\mathbf{\Delta} = \mathbf{\Gamma}^T\ \underline{or}\ \mathbf{\Psi} \tag{28}$$

$$\mathbf{\Psi}\ \underline{and}\ \mathbf{\Omega}_K \leq \mathbf{\Phi}_K \tag{29}$$

where

$\mathbf{\Phi}_K = (\phi_{S_1}^K, ..., \phi_{S_{n_1}}^K)^T$; $K = 0, N_1$ is the elementary state vector of the KB in the step K (the discrete step of the KB dynamics development). N_1 is an integer (the number of different situations). $\phi_{S_i}^K$, $i = 1, n_1$ is the state of the truth of the elementary statement S_i in the step K. It means that the statement is false (when 0), true (when 1) or that the statement is true with a fuzzy measure (when this parameter acquires its value from the real interval $< 0,\ 1 >$).

$\mathbf{\Omega}_K = (\omega_{R_1}^K, ..., \omega_{R_{m_1}}^K)^T$; $K = 0, N_1$ is the "control" vector of the KB (i.e. the state of the rules evaluability) in the step K. $\omega_{R_j}^K$, $j = 1, m_1$ is the state of the evaluability of the elementary rule R_j in the step K. It means that the rule is not able to be evaluated (when 0), the rule is able to be evaluated (when 1) or that the rule is able to be evaluated with a fuzzy measure (when this parameter acquires its value from the real interval $< 0,\ 1 >$ between these two boundary values).

$\mathbf{\Psi} = \{\psi_{ij}\}$, $i = 1, n_1$; $j = 1, m_1$ is the incidence matrix of the causal interconnections among the statements entering the rules and the rules themselves. $\psi_{ij} \in \{0, 1\}$ in the analogy with the LPNs and $\psi_{ij} \in < 0, 1 >$ in the analogy with the FPNs. In other words the element ψ_{ij} represents the absence (when 0), presence (when

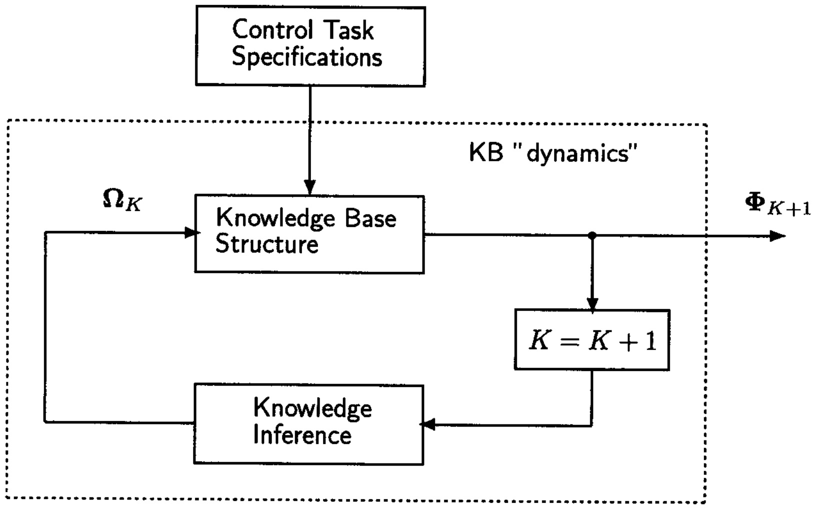

Figure 8: The principial schema of the KB.

1) or a fuzzy measure of existence (when its real fuzzy value is between these boundary values) of the causal relation between the input statement S_i and the rule R_j.

$\Gamma = \{\gamma_{ij}\}$, $i = 1, m_1$; $j = 1, n_1$ is incidence matrix of the causal interconnections among the rules and the statements emerging from them, where $\gamma_{ij} \in \{0, 1\}$, in case of the LPNs or $\gamma_{ij} \in <0, 1>$ in case of the FPNs. γ_{ij} expresses a measure of the occurrence of the causal relation between the rule R_i and its output statement S_j.

and is the operator of logical multiplying in general. For both the bivalued logic and the fuzzy one it can be defined (for scalar operands) to be the minimum of its operands. For example the result of its application on the scalar operands a, b is a scalar c which can be obtained as follows: $a \,\underline{and}\, b = c = min\,\{a, b\}$.

or is the operator of logical additioning in general. For both the bivalued logic and the fuzzy one it can be defined (for scalar operands) to be the maximum of its operands. For example the result of its application on the scalar operands a, b is a scalar c which can be obtained as follows: $a \,\underline{or}\, b = c = max\,\{a, b\}$.

The KB can be schematically illustrateded by means of Fig. 8.

This paper presents - as a continuation on the author's works [1]-[4] - the application of the fuzzy knowledge-based approach to the synthesis of the MS or/and FMS control. The illustrative example of real system is introduced too.

6.1 The example of the knowledge base construction

Let us try to represent knowledge in order to make the automatic solving of the previous control synthesis problem possible. The above introduced control task specifications and constraints can be expressed in the form of rule-based knowledge and the corresponding KB can be created. Utilizing the following statements
$S_1 = ({}^c\mathbf{x}_{k+1} = {}^m\mathbf{x}_{k+1})$; $S_2 = ({}^c\mathbf{x}_{k+1} = {}^c\mathbf{x}_0)$; $S_3 = ({}^m\mathbf{x}_{k+1} = {}^m\mathbf{x}_0)$; $S_4 = ({}^c\mathbf{x}_{k+1} = {}^m\mathbf{x}_0)$
$S_5 = ({}^m\mathbf{x}_{k+1} = {}^c\mathbf{x}_0)$; $S_6 = ({}^c\mathbf{x}_{k+1} \neq {}^m\mathbf{x}_{k+1})$; $S_7 = ({}^c\mathbf{x}_{k+1} \neq {}^c\mathbf{x}_0)$; $S_8 = ({}^m\mathbf{x}_{k+1} \neq {}^m\mathbf{x}_0)$; $S_9 = ({}^c\mathbf{x}_{k+1} \neq {}^m\mathbf{x}_0)$
$S_{10} = ({}^m\mathbf{x}_{k+1} \neq {}^c\mathbf{x}_0)$; S_{11} = there is only one vector ${}^c\mathbf{u}_k$; S_{12} = there is only one vector ${}^m\mathbf{u}_k$
S_{13} = there are several vectors ${}^c\mathbf{u}_k$; S_{14} = there are several vectors ${}^m\mathbf{u}_k$; S_{15} = accept ${}^c\mathbf{u}_k$; S_{16} = accept ${}^m\mathbf{u}_k$
S_{17} = elimine ${}^c\mathbf{u}_k$; S_{18} = elimine ${}^m\mathbf{u}_k$; S_{19} = solution does not exist; S_{20} = take another ${}^c\mathbf{u}_k$;
S_{21} = take another ${}^m\mathbf{u}_k$; S_{22} = (accept ${}^c\mathbf{u}_k$ and elimine ${}^m\mathbf{u}_k$) or (accept ${}^m\mathbf{u}_k$ and elimine ${}^c\mathbf{u}_k$); S_{23} = I do not know
and the following rules (where the symbol $\circ$ replaces the the symbol and)
R_1: *IF* $(S_1 \circ S_7 \circ S_{12} \circ S_{13})$ *THEN* $(S_{16} \circ S_{17} \circ S_{20})$
R_2: *IF* $(S_1 \circ S_8 \circ S_{11} \circ S_{14})$ *THEN* $(S_{15} \circ S_{18} \circ S_{21})$
R_3: *IF* S_6 *THEN* $(S_{15} \circ S_{16})$
R_4: *IF* $(S_1 \circ S_2 \circ S_{14})$ *THEN* $(S_{15} \circ S_{18} \circ S_{21})$
R_5: *IF* $(S_1 \circ S_3 \circ S_{13})$ *THEN* $(S_{16} \circ S_{17} \circ S_{20})$

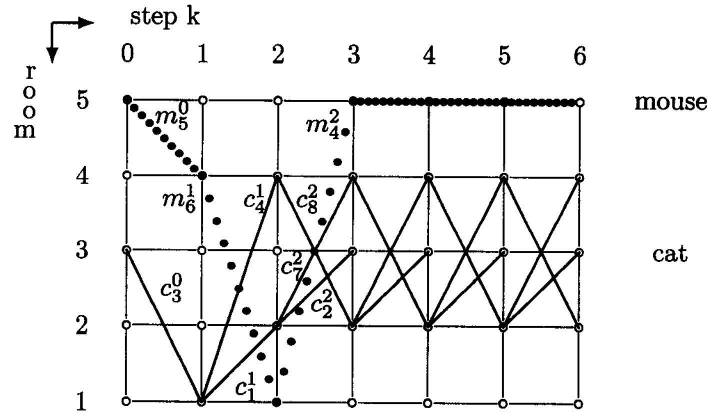

Figure 9: The graphical expression of the task solution.

R_6: *IF* $(S_1 \circ S_5 \circ S_{14})$ *THEN* $(S_{15} \circ S_{18} \circ S_{21})$
R_7: *IF* $(S_1 \circ S_4 \circ S_{13})$ *THEN* $(S_{16} \circ S_{17} \circ S_{20})$
R_8: *IF* $(S_4 \circ S_6 \circ S_{12} \circ S_{13})$ *THEN* $(S_{16} \circ S_{17} \circ S_{20})$
R_9: *IF* $(S_5 \circ S_6 \circ S_{11} \circ S_{14})$ *THEN*$(S_{15} \circ S_{18} \circ S_{21})$
R_{10}: *IF* $(S_2 \circ S_6)$ *THEN* $(S_{15} \circ S_{16})$
R_{11}: *IF* $(S_3 \circ S_6)$ *THEN* $(S_{15} \circ S_{16})$
R_{12}: *IF* $(S_1 \circ S_2 \circ S_{12})$ *THEN* S_{19}
R_{13}: *IF* $(S_1 \circ S_3 \circ S_{11})$ *THEN* S_{19}
R_{14}: *IF* $(S_1 \circ S_2 \circ S_5 \circ S_{11} \circ S_{12})$ *THEN* S_{19}
R_{15}: *IF* $(S_1 \circ S_3 \circ S_4 \circ S_{11} \circ S_{12})$ *THEN* S_{19}
R_{16}: *IF* $(S_1 \circ S_{11} \circ S_{12})$ *THEN* S_{19}
R_{17}: *IF* $(S_1 \circ S_7 \circ S_8 \circ S_{13} \circ S_{14})$ *THEN* S_{22}
R_{18}: *IF* $(S_4 \circ S_6 \circ S_{13})$ *THEN* S_{20}
R_{19}: *IF* $(S_5 \circ S_6 \circ S_{14})$ *THEN* S_{21}
R_{20}: *IF* $(S_1 \circ S_7 \circ S_8 \circ S_9 \circ S_{10} \circ S_{13} \circ S_{14})$ *THEN* S_{23}

we can obtain the structural matrices $\mathbf{\Psi}$, $\mathbf{\Gamma}$ $(n_1 = 23, m_1 = 20)$ of the analytical model of the PN-based knowledge representation. The nonzero elements of the matrix $\mathbf{\Psi} = \{\psi_{11}, \psi_{12}, \psi_{14}, \psi_{15}, \psi_{16}, \psi_{17}, \psi_{1,12}, \psi_{1,13}, \psi_{1,14}, \psi_{1,15}, \psi_{1,16}, \psi_{1,17}, \psi_{1,20}, \psi_{24}, \psi_{2,10}, \psi_{2,12}, \psi_{2,14}, \psi_{35}, \psi_{3,11}, \psi_{3,13}, \psi_{3,15}, \psi_{47}, \psi_{4,8}, \psi_{4,15}, \psi_{4,18}, \psi_{56}, \psi_{59}, \psi_{5,14}, \psi_{5,19}, \psi_{63}, \psi_{68}, \psi_{69}, \psi_{6,10}, \psi_{6,11}, \psi_{6,18}, \psi_{6,19}, \psi_{71}, \psi_{7,17}, \psi_{7,20}, \psi_{82}, \psi_{8,17}, \psi_{8,20}, \psi_{9,20}, \psi_{10,20}, \psi_{11,2}, \psi_{11,9}, \psi_{11,13}, \psi_{11,14}, \psi_{11,15}, \psi_{11,16}, \psi_{12,1}, \psi_{12,8}, \psi_{12,12}, \psi_{12,14}, \psi_{12,15}, \psi_{12,16}, \psi_{13,1}, \psi_{13,5}, \psi_{13,7}, \psi_{13,8}, \psi_{13,17}, \psi_{13,18}, \psi_{13,20}, \psi_{14,2}, \psi_{14,4}, \psi_{14,6}, \psi_{14,9}, \psi_{14,17}, \psi_{14,19}, \psi_{14,20}\}$ and the nonzero elements of the matrix $\mathbf{\Gamma} = \{\gamma_{1,16}, \gamma_{1,17}, \gamma_{1,20}, \gamma_{2,15}, \gamma_{2,18}, \gamma_{2,21}, \gamma_{3,15}, \gamma_{3,16}, \gamma_{4,15}, \gamma_{4,18}, \gamma_{4,21}, \gamma_{5,16}, \gamma_{5,17}, \gamma_{5,20}, \gamma_{6,15}, \gamma_{6,18}, \gamma_{6,21}, \gamma_{7,16}, \gamma_{7,17}, \gamma_{7,20}, \gamma_{8,16}, \gamma_{8,17}, \gamma_{8,20}, \gamma_{9,15}, \gamma_{9,18}, \gamma_{9,21}, \gamma_{10,15}, \gamma_{10,16}, \gamma_{11,15}, \gamma_{11,16}, \gamma_{12,19}, \gamma_{13,19}, \gamma_{14,19}, \gamma_{15,19}, \gamma_{16,19}, \gamma_{17,22}, \gamma_{18,20}, \gamma_{19,21}, \gamma_{20,23}\}$. The KB obtained on this way helps to find automatically the solution of the control task - i.e. to obtain the automatic synthesis of the control actions. E.g. when in the step $k = 2$ of the DES development the situation is that the input statements S_1, S_2, S_5, S_8, S_9, S_{13}, and S_{14} are true the KB will start from the initial state $\mathbf{x}_K$, $K = 0$ given as follows

$$\mathbf{\Phi}_0 = (11001001100011 \| 000000000) \tag{30}$$

Hence, the rules R_4 and R_6 can be evaluated as it can be seen in the following vector given by the inference mechanism presented e.g. in [3]

$$\mathbf{\Omega}_0 = (00010100000000000000) \tag{31}$$

Consequently, the output statements S_{15}, S_{18}, and S_{21} will be true as it can be seen from the state vector of the KB in the step $K = 1$ of the KB dynamics development

$$\mathbf{\Phi}_1 = (11001001100011 \| 100100100) \tag{32}$$

Cat Behaviour	Mouse Behaviour
Step k = 0	
${}^{c}\mathbf{x}_0 = (0\,0\,1\,0\,0)^T$	${}^{m}\mathbf{x}_0 = (0\,0\,0\,0\,1)^T$
${}^{c}\overline{\mathbf{x}}_0 = (1\,1\,0\,1\,1)^T$	${}^{m}\overline{\mathbf{x}}_0 = (1\,1\,1\,1\,0)^T$
${}^{c}\mathbf{w}_0 = (0\,0\,1\,0\,0\,0\,0\,0)^T$	${}^{m}\mathbf{w}_0 = (0\,0\,0\,0\,1\,0)^T$
${}^{c}\mathbf{u}_0 = {}^{c}\mathbf{w}_0$	${}^{m}\mathbf{u}_0 = {}^{m}\mathbf{w}_0$
${}^{c}\mathbf{x}_1 = {}^{c}\mathbf{x}_0 + \mathbf{B}_c.{}^{c}\mathbf{u}_0$	${}^{m}\mathbf{x}_1 = {}^{m}\mathbf{x}_0 + \mathbf{B}_m.{}^{m}\mathbf{u}_0$
${}^{c}\mathbf{x}_1 = (1\,0\,0\,0\,0)^T$	${}^{m}\mathbf{x}_1 = (0\,0\,0\,1\,0)^T$
${}^{c}\mathbf{x}_1 \neq {}^{m}\mathbf{x}_1$	
Step k = 1	
${}^{c}\mathbf{w}_1 = (1\,0\,0\,1\,0\,0\,0\,0)^T$	${}^{m}\mathbf{w}_1 = (0\,0\,0\,0\,0\,1)^T$
the control possibilities are: $\{\underline{c_1}, \underline{c_4}, \underline{m_6}\}$ all of them are possible	
${}^{c}\mathbf{u}_1^1 = (1\,0\,0\,0\,0\,0\,0\,0)^T$	${}^{m}\mathbf{u}_1 = {}^{m}\mathbf{w}_1$
${}^{c}\mathbf{x}_2^1 = (0\,1\,0\,0\,0)^T$	${}^{m}\mathbf{x}_2 = (1\,0\,0\,0\,0)^T$
${}^{c}\mathbf{u}_1^2 = (0\,0\,0\,1\,0\,0\,0\,0)^T$	= n o n e =
${}^{c}\mathbf{x}_2^2 = (0\,0\,0\,1\,0)^T$	= n o n e =
${}^{c}\mathbf{x}_2^1 \neq {}^{m}\mathbf{x}_2$; ${}^{c}\mathbf{x}_2^2 \neq {}^{m}\mathbf{x}_2$	
Step k = 2	
${}^{c}\mathbf{w}_2^1 = (0\,1\,0\,0\,0\,0\,1\,0)^T$	${}^{m}\mathbf{w}_2 = (1\,0\,0\,1\,0\,0)^T$
${}^{c}\mathbf{w}_2^2 = (0\,0\,0\,0\,1\,0\,0\,1)^T$	= n o n e =
the control possibilities are: $\{\underline{c_2}, c_7, \not{m}_1, \underline{m_4}\}$; c_2 has priority to m_1 $\{\not{c}_5, c_8, \not{m}_1, \underline{m_4}\}$; m_4 has priority to c_5	
${}^{c}\mathbf{u}_2^{11} = (0\,1\,0\,0\,0\,0\,0\,0)^T$	${}^{m}\mathbf{u}_2 = (0\,0\,0\,1\,0\,0)^T$
${}^{c}\mathbf{u}_2^{12} = (0\,0\,0\,0\,0\,0\,1\,0)^T$	= n o n e =
${}^{c}\mathbf{u}_2^2 = (0\,0\,0\,0\,0\,0\,0\,1)^T$	= n o n e =
${}^{c}\mathbf{x}_3^{11} = (0\,0\,1\,0\,0)^T$	${}^{m}\mathbf{x}_3 = (0\,0\,0\,0\,1)^T$
${}^{c}\mathbf{x}_3^{11} = {}^{c}\mathbf{x}_0$	${}^{m}\mathbf{x}_3 = {}^{m}\mathbf{x}_0$
${}^{c}\mathbf{x}_3^{12} = (0\,0\,0\,1\,0)^T$	= n o n e =
${}^{c}\mathbf{x}_3^2 = (0\,1\,0\,0\,0)^T$	=n o n e =

Table 1: The results of the control synthesis

References

[1] Čapkovič, F.: Using Fuzzy Logic for Knowledge Representation at Control Synthesis. *BUSEFAL*, **63**, 1995, pp. 4-9.

[2] Čapkovic, F.: Knowledge-Based Control of DEDS. In: Proc. of the 13th IFAC World Congress 1996, San Francisco, USA, June 30-July 5, 1996, Vol. J, Compact Disc, Elsevier Science Ltd., Pergamon, 1996, paper J-3c-02.6, pp. 347-352.

[3] Čapkovič, F.: Petri nets and oriented graphs in fuzzy knowledge representation for DEDS control purposes. *BUSEFAL*, **69**, 1997, pp. 21-30.

[4] Čapkovič F.: Fuzzy knowledge in DEDS control synthesis. In: M. Mareš, R. Mesiar, V. Novák, J. Ramik, A. Stupňanová (Eds.): Proceedings of the 7th International Fuzzy Systems Association World Congress - IFSA'97, Prague, Czech Republic, June 25-29, 1997, pp. 550-554, Academia, Prague, Czech Republic, 1997, pp. 550-554.

[5] Ramadge, P.J.G., Wonham, W.M.: The control of discrete event systems. Proceedings of the IEEE (Ed. Y.CH. Ho), **Vol. 77**, No 1, 1989, pp. 81-98.