NOTAS DE MATEMATICA

Nº 137

DISCRETE DETERMINABLE PROCESSES-COMPACT KNOWLEDGE-BASED MODEL

(DISCRETE DETERMINISTIC CONTROLLED PROCESSES- ALGEBRAIC LOGICAL MODEL)

BY

EWA DUDEK-DYDUCH

PRE-PRINT

UNIVERSIDAD DE LOS ANDES
FACULTAD DE CIENCIAS
DEPARTAMENTO DE MATEMATICA
MERIDA-VENEZUELA
1993

DISCRETE DETERMINABLE PROCESSES - COMPACT KNOWLEDGE-BASED MODEL (DISCRETE DETERMINISTIC CONTROLLED PROCESSES - ALGEBRAIC-LOGICAL MODEL)

Ewa Dudek-Dyduch
Institute of Automatics,
University of Mining and Metallurgy
al. Mickiewicza 30, 30-059 Krakow, Poland
tel\fax: + 48 12 34 15 68

DISCRETE DETERMINABLE PROCESSES - COMPACT KNOWLEDGE-BASED MODEL (DISCRETE DETERMINISTIC CONTROLLED PROCESSES - ALGEBRAIC-LOGICAL MODEL)

KEY WORDS: knowledge-based model, state-based models, control of discrete event processes, manufacturing processes, expert systems.

ABSTRACT

The paper presents the special compact knowledge-based model of some class of discrete event processes. The model is appropriated for control. The comparisons of the model with other models (AI knowledge-based model, state graph and Petri nets) are given. Application of the model is described.

INTRODUCTION

The paper deals with formal model of some class of discrete event processes, namely such processes that events are considered not to occur spontaneously but to be depended on the decisions taken previously. Moreover, the instants of particular events occurrences can be computed when sequence of decisions (control) is known. Such processes are named in this paper **as discrete determinable event processes (DDEP)**. The attribute "**determinable**" is just to stress that occurrences of events are determined by the decisions i.e. depend on control. The model is appropriated for control of the processes. Both events and decisions can have different interpretation. The examples of control of DDEP include control of discrete manufacturing processes (with no disturbances) and especially operational production scheduling, determining the optimal tour of salesman, scheduling pro- grams processed by computers, and many other combinatorial optimization problems.

According to the author, the name "**discrete deterministic controlled processes**" better reflects the nature of the processes under consideration.

The aim of the paper is 3-fold:

- to present a general, knowledge based model of a vast class of DDEP,
- to compare it with other models,
- to discuss the model applications, especially the application for discrete manufacturing processes.

Numerous approaches to the modelling of discrete event processes have appeared in literature (see (Cassandras 1990), (Cao 1990)). The variety of approaches reflects the diversity of areas in which the discrete event processes play an important role and different aims for which the models have been devised. Thus, let us present the purpose for which the proposed in this paper formalism has been worked out.

As it is known, variety of DDEP and their control algorithms are considered in literature (combinatorial problems of that type and their optimization algorithms). They are presented by means of different kinds of models and sometimes with use of verbal description only. Different algorithms were presented for the same problems but the lack of uniform formalization makes determination of formal differences between their conceptions impossible. Moreover, lack of uniform, formal theory makes difficult to create algorithms for novice engineers. The role of the proposed model can be characterized as follows:

- it enable us to present both problems and algorithms in the uniform, formal way,
- it is a basis for uniform language for defining properties of problem to be solved and to clarify relationships between problems properties and manners of solving,
- it is basis for formal comparison of different algorithms proposed for the same problem and for more detailed classifications of algorithms.

Additionally, there is the important role of the model for expert system for control of discrete manufacturing processes.

KNOWLEDGE BASED MODEL OF DISCRETE PROCESS

In the earlier paper of the author (Dudek 1988) a formal model of a vast class of discrete event processes has been presented. Now, let us present its slightly modified version.

Let us assume the following denotation:

X is a set named a set of proper states,

U is a set named a set of control decisions (control signals),

$\mathbf{S} = \mathbf{X} \times (\mathbf{R}^{+} \cup \{0\})$ is a set named set of generalized states, where $\mathbf{R}^{+} \cup \{0\}$ is a set of non negative real numbers representing the time instants and denoted as $\mathbf{T}$,

thus $\mathbf{S} = \mathbf{X} \times \mathbf{T}$.

Definition

The individual discrete process P is a process that is defined by the six-tuple $P = (\mathbf{U}, \mathbf{X}, s_0, f, \mathbf{S}_N, \mathbf{S}_G)$ where:

U, **X**, **S** are determined former,

$f\colon \mathbf{U} \times \mathbf{S} \rightarrow \mathbf{S}$ is a partial function called a transition function, (it has not to be determined for all elements of the set $\mathbf{U} \times \mathbf{S}$), and defined by means of two functions: $f = (f_x, f_t)$ where

$f_x : \mathbf{U} \times \mathbf{X} \times \mathbf{T} \rightarrow \mathbf{X}$ determines the next proper state and

$f_t : \mathbf{U} \times \mathbf{X} \times \mathbf{T} \rightarrow \mathbf{T}$ determines the next time instant; there is assumed that the difference $\Delta t = f_t(u,x,t) - t$ has a value which is both finite and positive,

$s_0 = (x_0, t_0)$, $\mathbf{S_N} \subset \mathbf{S}$, $\mathbf{S_G} \subset \mathbf{S}$ are respectively: an initial generalized state, a set of not admissible generalized states, and a set of goal generalized states, i.e. the states in which we want the process to take place at the end.

The model takes into account only these instants of time at which any event occurs. Function f was defined as a partial function. This is so in order to deal with all the limitations concerning the control decisions in the current state in a convenient way. This is done by means of the so-called sets of possible decisions in state s, denoted as $\mathbf{U_p}(s)$ and defined as:

$$\mathbf{U_p}(s) = \{ u \in \mathbf{U} : (u,s) \in \mathrm{Dom}\, f \}.$$

At the same time an individual process P is represented by a set of its trajectories i.e. finite or infinite sequences of generalized states, that are determined by means of the transition function. A trajectory starts from the distinguished initial state s_0 and it is assumed to end (if it is a finite one) in one of the following ways:

- it can end in the goal state set $\mathbf{S_G}$,
- it can end in the not admissible state set $\mathbf{S_N}$,
- it can end in any state for which the set of possible decisions $\mathbf{U_p}(s)$ is empty.

It is assumed that no state of a trajectory, apart from the last one, may belong to the set $\mathbf{S_N}$ or have empty set of possible decisions. A trajectory that ends in the set of the goal states $\mathbf{S_G}$ is an admissible one and the decision sequence determining it is an **admissible** decision sequence. In other cases, a trajectory is said to be a not admissible one.

The **task** of optimization of control of DDEP lies in finding of such an admissible decisions sequence $\tilde{u}$,that minimizes a certain criterion Q. The optimization task is uniquely determined by the pair (P,Q) where process P represents all the task limitations.

A defined **problem** of control optimization is such a set of optimization tasks that have a commonly determined set of data (the same type of data fulfilling the commonly determined properties), common (parametric) definition of set of individual discrete processes and common (parametric) definition of criterion functions. Such determined set of processes will be also called a process but will be denoted **P** . The problem will be denoted **(P,Q)**.

In the most general case sets **U** and **X** may be presented as a Cartesian product

$\mathbf{U} = \mathbf{U}^1 \times \mathbf{U}^2 \times ... \mathbf{U}^m$, $\mathbf{X} = \mathbf{X}^1 \times \mathbf{X}^2 \times ... \mathbf{X}^n$. Thus a decision u as well as a state x is represented by finite sequence of values that belong to succeeding sets. There are no limitations imposed on the sets; in particular the sets need not be numerical ones. We will use the notions "state vector" and "control decision vector" by analogy to the classical control theory:

$$u = (u^1, u^2, .. u^m) \quad x = (x^1, x^2, .. x^n).$$

Thus values of particular coordinates of a state as well as a control decision may be names of elements as well as some objects (e.g. finite set, sequence, etc.).

Particular coordinates of the vector $u = (u^1, u^2, .. u^m)$ represent a separate decisions that refer to different matters (different objects or activities). One should point out that the moment when the decision is taken and the moment when the realization of the decision starts need not be the same for the decision processes. In the considered model, particular coordinates of the decision vector represent the decisions that must or may be taken at the same time. It is assumed for the sake of being concise (for minimal number of distinguished states). Obviously, the fact that a decision is taken earlier than it begins to be realized is taken into account in the transition function algorithm.

It is assumed for the model, that at least one set $\mathbf{U}^i$ or $\mathbf{X}^i$ is not numerical one. As a consequence, the sets $\mathbf{S}_N$, $\mathbf{S}_G$ and $\mathbf{U}_p$ are formally defined with use of logical formulae. Therefore, the complete model constitutes a specialized form of knowledge-based model (algebraic-logical model) and for the use of this paper it will be named compact knowledge-based model (CKBM).

The proper state x is assumed to be so-called minimal state i.e. the state is presented with use of minimal information. In models of some problems, the coordinate t depends on the proper state, i.e. t_i is a function of x_i . In models of others problems, however, the coordinate t plays the same role as an additional coordinate of a state, i.e. the pair (x,t) constitutes the minimal state. In order to present the two kinds of problems uniformly, we use the notion "**generalized state**".

Limitations referring to the set of not admissible states $\mathbf{S}_N$ may be used to define the so-called sets of admissible decisions in state s , denoted as $\mathbf{U}_d(s)$.

$$\mathbf{U}_d(s) = \{ u \in \mathbf{U}(s) : f(u,s) \notin \mathbf{S}_N \}$$

Let us point out that a lot of discrete manufacturing processes can be presented by means of the above model (the symbols f,U,X can be interpreted by means of manufacture terms). In fact, the CKBM has been devised just for control of discrete manufacturing processes.

DISCRETE MANUFACTURING PROCESSES - EXAMPLES

Control of discrete manufacturing processes (DMP) lies in determining a manner of performing some set of jobs under restrictions referring to machines, resources, energy, time, transportation possibilities, order of operations performing and others. Variety of manufacturing processes and and control algorithms are considered in literature. They are presented by means of different kind of models and some- times by means of verbal description only. Thus, although we have a large knowledge related to this area, there is no formal theory of these processes. It makes diffi-cult to create algorithms for novice engineers.

The question arises: can one create the expert system that would assist to novices in designing algorithms for particular problems? In order to do it, the formal, general model of the class of problems under consideration must be worked out. The CKBM is proposed to be just such a model.

Now, let us present models of some exemplary manufacturing processes (the most simple ones). The model of really complicated process can be find in (Dudek 1987)

Before we give the examples, let us pay our attention to some notification matters. Because no restrictions referring to values of state coordinates and decision coordinates have been assumed, thus different objects may be the values. Particularly, they may be names of some elements. In order to differentiate the name of variable from its value, the proper notification should be used e.g. "w(x)" to indicate a value of variable x ," $w(x) \in A$" - value of x belongs to set A , or "$w(x) \subset B$" - value of x is contained in set B . We will use, however, the simplified notifica-tion, i.e. $x \in A$, $y \subset B$ to denote that value of x belongs to A or value of y are contained in B respectively. Moreover, in order to emphasize that some function (or other object), formally defined as a function of several arguments, does not depend on part of them in a special case, we will omit needless arguments (e.g. writing "f(x)" instead of "f(u,x,t)").

Example 1.

There is given the finite set of jobs that are represented by natural numbers $\mathbf{Z} = \{1,2...n\}$ and the function $\tau : \mathbf{Z} \rightarrow \mathbf{R}^+$ that determines the processing times. The jobs are to be performed by means of one machine. We have to find an order of the jobs performing so that any regular performance index be minimal (e.g. the weighted sum of the particular jobs completion times be minimal).

Let us define the elements specifying the process $\mathbf{P} = (\mathbf{U}, \mathbf{X}, s_0, f, S_N, S_G)$. The proper state x is determined by the set of completed jobs and the decision consists in determining the next job to be performed. Thus, the decision value is the name (number) of the chosen job. Formally:

the state set $\mathbf{X}$ is a set of all subsets of the jobs $\mathbf{X} = 2^{\mathbf{Z}}$,

the decision set $\mathbf{U}$ is the same as the set of jobs $\mathbf{U} = \mathbf{Z}$, there are no not admissible states, thus $S_N = \varnothing$;

the set of goal states $S_G = \{(x,t) \in \mathbf{S} : x = Z\}$ (the process is in the goal state if all the jobs are completed);

the initial state $s_0 = (\varnothing,0)$ as we assume $t_0 = 0$,

the transition function $f : \mathbf{U} \times \mathbf{X} \times \mathbf{T} \to \mathbf{X} \times \mathbf{T}$ is determined as follows:

$\mathbf{U_p}(x,t) = \mathbf{Z} \setminus x$ thus $\mathbf{U_p}(s_0) = \mathbf{Z}$

$f_x(u,x,t) = x \cup \{u\}$, $f_t(u,x,t) = t + \tau(u)$.

The transition function has the following properties.

1. The set of possible decisions depends on proper state only i.e. $\mathbf{U_p}(s) = \mathbf{U_p}(x)$.

2. The set of possible decisions in the state x_{i+1} may be computed by means of simply modification of the previous set: $\mathbf{U_p}(x_{i+1}) = \mathbf{U_p}(x_i) \setminus \{u_i\}$.

3. For each state s , the set of admissible decision $\mathbf{U_d}(s)$ is equal to the set of possible decision $\mathbf{U_p}(s)$; thus any permutation may be an admissible solution.

4. The function f_t does not depend on state x directly, but it depends on decision only.

The same task may be formalized in another way. Namely, the decisions set can be assumed as $\mathbf{U} = \mathbf{Z} \times \mathbf{R}^+$ where

$u = (u^1, u^2) = (i, \Delta t)$; u^1 denotes the number of task and u^2 - the assigned time such that $\Delta t > t(i)$. The transition function is determined by the formulas:

$f_x(u,x) = x \cup u^1$, $f_t(u,x,t) = t + \Delta t$.

Now, the sets $\mathbf{U}$, $\mathbf{U_p}(s)$ are infinite and uncountable. Since $S_N = \varnothing$, there exist infinitely many admissible trajectories. However, the simple analysis, that takes into account the properties of criterion, enable us to replace this model by the previous one.

Let us notice that a definition given with use of set theory terms is equivalent to some logical formula. The set theory definitions are used here in order to have concise form of the model.

Example 2.

Let us modify the example 1 by introduce the additional restriction, namely we assume that due dates must be observed. Let the function $d: \mathbf{Z} \to \mathbf{R}^+$ determine the due dates for the jobs.

The additional restriction changes only the definition of the set of not admissible states S_N , whereas the other elements $\mathbf{X}$, $\mathbf{U}$, f, s_0 ,S_G are the same. The set S_N consists of all the pairs (x,t) such that the time t is greater or equal to the due date of some job and the job is not completed yet:

$$S_N = \{ (x,t) \in S : \exists\, i \in \mathbf{Z} ,\ i \notin x ,\ d(i) \leq t \}$$

Let us notice that in this case not each permutation of jobs creates an admissible control sequence $u^{\sim}$.

The CKBM takes into account the following properties of the real manufacturing processes :

- both influences on the process (control) and the observable change of state take place in strictly defined time moments and the intervals between these moments need not be equal; in a general case next moment depends on the state, decision and current time;

- both physical signals and decision may be controls; no limitations are imposed on the structure of the set of controls and states (does not need be defined in a numerical space);

- production technology limitations determine the decisions which are possible in given situations (e.g. limitations referring to the order of operations); on the other hand, not all decisions defined formally make sense in certain conditions; all this is taken into account when defining the set $\mathbf{U}_P(s)$; so the calculation of the transition function is realized in two stages: verifying if the decision belongs to the set $\mathbf{U}_P(s)$ and determination of next state;

- in the process there additionally exist limitations concerning both proper states and time (due dates); they are taken into account by means of definition of S_N .

COMPARISON WITH OTHER MODELS

The CKBM belongs to the class of state-based models. Let us sum up its characteristic features.

1. Model is appropriated not only for particular tasks (individual processes) but also for problems (i.e. process **P** can be defined with use of parameters that may be different objects as function, relations etc.)

2. The state set as well as the decision set may be infinite and uncountable.

3. The state set as well as the decision set is defined as a Cartesian product of sets which need not be the number ones (they may be of any type). Thanks to it, we may compare different structures of states and decisions.

4. The transition function is defined as a partial function. It enables us to formally separate the restriction relating to decisions and relating to states.

5. There is a special numerical coordinate monotonously increasing along each trajectory. This coordinate was given time interpretation.

6. The time increase for the successive states of any trajectory depends both on the state and the decisions, and it has a finite value.

AI knowledge-based model

The presented model is generalization of the classical model of discrete dynamic systems used in control theory: $x_{i+1} = F(u_i, x_i, t_i)$. The generalization lies in the fact that both the state and decision coordinates need not be numerical ones and that the definition of the sets $\mathbf{U_p}$, S_N, S_G are given by means of logical formulae.

On the other hand, the model is a special kind of a knowledge based model of dynamic system. In (Fikes 1971), (Ligeza 1992), (Nilsson 1980) the dynamic system is specified by means of state formulae and transformation rules. Any state $s \in \mathbf{S}$ has a unique corresponding state formula. For intuition, a state formula corresponding to some state s is a conjunction of all the facts that are true in this state and are possible to be expressed in the assumed language. For practical reasons mostly systems having finite state formulae are considered. The information determining a state can be redundant and usually it is redundant. The possible changes of states are described by means of transformations rules. Any of such rules contains several parts, i.e. a part defining when it is possible to apply the particular rule (preconditions), a part specifying the action to be taken (action), and the part specifying the changes in the state description. The changes of current state formula can be executed by retracting the facts which after the actions specified by the rule are no longer true (delete_results), and the ones which become true as a result of the action (add_results). The basic form of any transformation rule is as follows:

```
if          preconditions (s)
then
      do      action(s)
      retract delete_result(s)
      assert  add_result(s)
```

The interpretation of this scheme is straightforward: if precondition(s) is true (i.e. satisfied with regard to the current state s) then the specified action is to be executed, the knowledge base constituting the current state formula is to be modified (updated) by retracting delete_results(s) (the ones which are no longer true) and asserting add_result(s) (the ones which become true). This form is known as the AI (artificial intelligence) knowledge-based model.

In opposite to it, the introduced CKBM is specified by means of six elements $(\mathbf{U}, \mathbf{X}, s_0, f, S_N, S_G)$, where the definition of the transition function $f: \mathbf{U} \times \mathbf{S} \rightarrow \mathbf{S}$ consists of definition of its domain (sets $\mathbf{U_p}(s)$) and the algorithm determining the next state. Let us describe the correspondence and the differences between these formalisms.

The definition of sets $\mathbf{U_p}(s)$ determines all possible actions (decisions related to actions) in any state $s \in \mathbf{S}$. The transition functions f corresponds to all the transformation rules that constitutes a particular AI model. Note that existence of the transformation function enables us to define different properties of processes (especially the properties that are satisfied by all the transformation rules), as it has been shown in (Dudek 1988, 1989 1992).

The crucial difference between the models consists in the representation of states. In the CKBM a state is defined by means of minimal information and it is presented in the form of vector. Thus, the pieces of information (facts) are presented in some ordered way. It is necessary for one to be able to compare different processes. It is also convenient for analysing the changes of particular facts, that occur in course of process and, as a result of the analysis, for defining the process properties. We should point out, however, that the AI model is mostly used for reasoning, for which the minimal state representation would be inconvenient.

The concise form of the model proposed in this paper justifies its name as "compact knowledge-based model". It is easy to see that each AI knowledge-based model can be presented in form of CKBM.

State graph model

Let us consider an individual process P such that its decisions set **U** is a countable one. We can define the state graph of the process in the following way.

The graph $G = (S^P, R)$ is the state graph of individual process P if the set of nodes S^P is a set of all states belonging to the process trajectories and relation $R \subseteq S \times S$ is defined as follows:

$$(s_i, s_j) \in R \text{ iff there exists } u \in \mathbf{U}_p(s_i) \text{ such that } s_j = f(u, s_i).$$

The state graph model is widely used for presenting and analysing some class of algorithms especially within the frame of artificial intelligence (Pearl 1984). Let us notice that the CKBM is more general than the state graph model because the set of decisions (actions) need not be a countable one. Practically, the graph model is useful for the so-called local finite processes, i.e. the ones for which the sets of possible decisions $\mathbf{U}_p(s)$ are finite, and it is used to present limited class of algorithms (exhaustive or implicit search algorithms and some heuristic ones). Thus a lot of processes cannot be modelled with use of state graph (e.g. scheduling under resource constrains) Moreover, the CKBM enable us to define more properties than the graph (e.g. one can analyse a structure of a state and decision and changes of particular coordinates values).

It should be also stressed that both the states graph model and the AI knowledge-based model are appropriated only for individual process (task) while the CKBM can be applied for both the task and the problem (i.e. for finite or infinite set of tasks). Thus CKBM corresponds to a family of state graphs or a family of AI knowledge-based models.

Petri net model

Now, let us compare the CKBM with Petri net model. The simple Petri net (PN) view of a system concentrates on two concepts, namely, transitions and conditions. Transitions are actions that take place in the system, and the occurrence of these transitions is controlled by the state of the system. The state of the system may be described as a set of conditions that have logical descriptions.

For a transition to occur, it may be necessary for certain conditions to hold, and these are termed the preconditions of the transition. The occurrence of the transition may cause the precondition to cease to hold and may cause other conditions, called postconditions, to become true.

A Petri net needs four entities for description: places, transitions, inputs and outputs. Places are used to represent conditions and transitions are used to represent events. The inputs and outputs are mappings from transitions to places and from places to transitions respectively. Markings are used to assign tokens to places. Tokens move through a Petri net and are used to define the execution of the net. A Petri net executes by firing transitions. As a result of firing of a transition, the distribution of tokens is changed. The dynamic properties of system are represented by changes of net markings. The original Petri net has been extended and modified. A class of Petri net with inhibitor arcs, coloured Petri nets, predicate/transition Petri nets and timed Petri nets have been worked out.

It is easy to notice the correspondence between PN and state space model. A current state of the system is determined by marking (tokens in places). Decisions correspond to transitions (if all the events are controlled) and the transition function is represented by changes of marking.

Petri net model captures wider class of discrete event processes that the presented CKBM (the process need not be embedded in time) but its application is different. The practical application of Petri net model consists in designing and analysing of systems. The model is appropriated for solving such problems as concurrence and parallelism but the model is not convenient to create and compare control optimization algorithms for the processes considered in the paper.

It should be also underlined that PN model is created for the system that structure is fixed to some extent. If for example we model the manufacturing system with use of Petri net, then the number of machines or buffers must be fixed, while these numbers may be parameters in a CKBM.

CONCLUSION

The paper presents the compact form of knowledge-based model appropriated for the class of discrete determinable processes. Applications of the model can be characterized as follows. The model enables us:

1) to present uniformly different problems and define their properties; the properties are base for formal classification and comparison of problems,

2) to present formally, discuss and classify different control algorithms, especially, to analyse them using formally defined properties and, as a result, to give rules for synthesis of new algorithms,

3) to discuss heuristic algorithms; since the problems and algorithms can be presented formally, heuristic algorithms can be considered as simplified ones or as the exact ones but optimizing the simplified problem; a kind of simplification can be defined and its consequences can be analysed.

There is a special role of the model for an expert system for discrete manufacturing processes control. The role is twofold:

a) it provides general paradigm for representation knowledge on processes and at the same time it is a basis for simulation of them.

b) it is basis for uniform language for defining properties of problems to be solved, for presenting algorithms and to clarify relationships between problems properties and manners of solving; briefly speaking it is necessary for knowledge acquisition.

REFERENCES

1. Banaszak Z. (ed) (1991): Modelling and control of FMS. Petri Net Approach. Wroclaw TU Press, Wroclaw.
2. Bubnicki Z.(1990): Introduction to expert systems, PWN, Warsaw.
3. Cao X.R., Ho Y.C.(1990): Models of discrete event dynamic systems, IEEE Control Systems, vol.10, No 4, 69- 76.
4. Cassandras C.G., Ramadge P.J.(1990): Toward a control theory for discrete event systems, IEEE Control Systems, vol.10, No 4, 66-68.
5. Dudek-Dyduch E.(1987): Simulation of some class of discrete manufacturing processes, Proc.of Europ.Congress on Simulation, Prague .

6. Dudek-Dyduch E., Dyduch T.(1988): Scheduling some class of discrete processes, Proc.of 12th IMACS World Congress, Paris .

7. Dudek-Dyduch E., (1989): Formal bases of Classification of Discrete Production Processes Control Problems, Zesz. Nauk. AGH Nr 1313, Automatyka z.49, 8, pp 189-200.

8. Dudek-Dyduch E. (1992): Control of discrete event processes - Branch and bound method, Preprints of IFAC/IFORS/IMACS/ Symposium Large Scale Systems, Beijing, 573-579.

9. Fikes R.E., Nilsson N.J. (1971): STRIPS: a new approach to the application of theorem prooving to problem solving. Artificial Intelligence 2, 189-208

10. Kanal L., Kumar V (ed)(1988): Search in artificial intelligence, Springer Verlag New York Inc. .

11. Ligeza A. (1992): Logical foundations for knowledge-based control systems Raport LAAS No 92269.

12. Nilsson N.J. (1980): Principles of Artificial Intelligence. Tioga Publishing Co., Palo Alto, California.

13. Pearl J.(1984): Heuristics. Intelligent search strategies for computer problem solving. Addison-Wesley Comp. Menlo Park.

14. Ramadge P.J., Wonham W.M. (1985): Supervisory control of a class of discrete event processes, Systeme Control Group Report #8515, University of Toronto.

The research was supported by the project No 8-8528-9102 "Computer Systems for Control and Decision Making" of KBN (State Committee on Scientific Research).