P. CONSTANTINESCU
G. PĂUN
Ştefan Gheorghiu Academy

The MIS as a Two Loops Cybernetic System

Using operation-research considerations and a mathematical-linguistic approach of the production process, we prove that is necessary to describe the MIS as a cybernetic system with two loops, and, that this structure is as well sufficient to model the management by planning and the managed system. Such a MIS is said to be a MIS of the class S_2.

In the last years, the mathematical-linguistics is proving more and more its utility as a tool for the formalisation of an increasing number of phenomena which apparently have no connection with neither the natural nor the programming languages. A great number of examples of fields in which the mathematical-linguistics is applied are given in [7], [9]. Particularly, the automata theory, implicitly, the formal languages theory has a great theoretical and practical significance as a tool for modelling.

Following Kornai Janos [6], by such a modelling "we succeed in describing and analysing the systems by the help of general theses from economic cybernetics, the mathematical system theory and automata theory", and, especially, we can simulate on the computer the modelled system. "It is known that as much we can describe the behavior of a system (string of events) by automata theory, as more we can simulate the system on the computer. Automata theory and the simulation language on the computer are strongly related".

1. INFORMATION SYSTEMS WITH TWO LOOPS ([1], [2], [3])

We shall consider further the MIS as a model of the management and the managed process. This is an accurate and profitable approach for the purpose of formalisation, describing and implementing MIS.

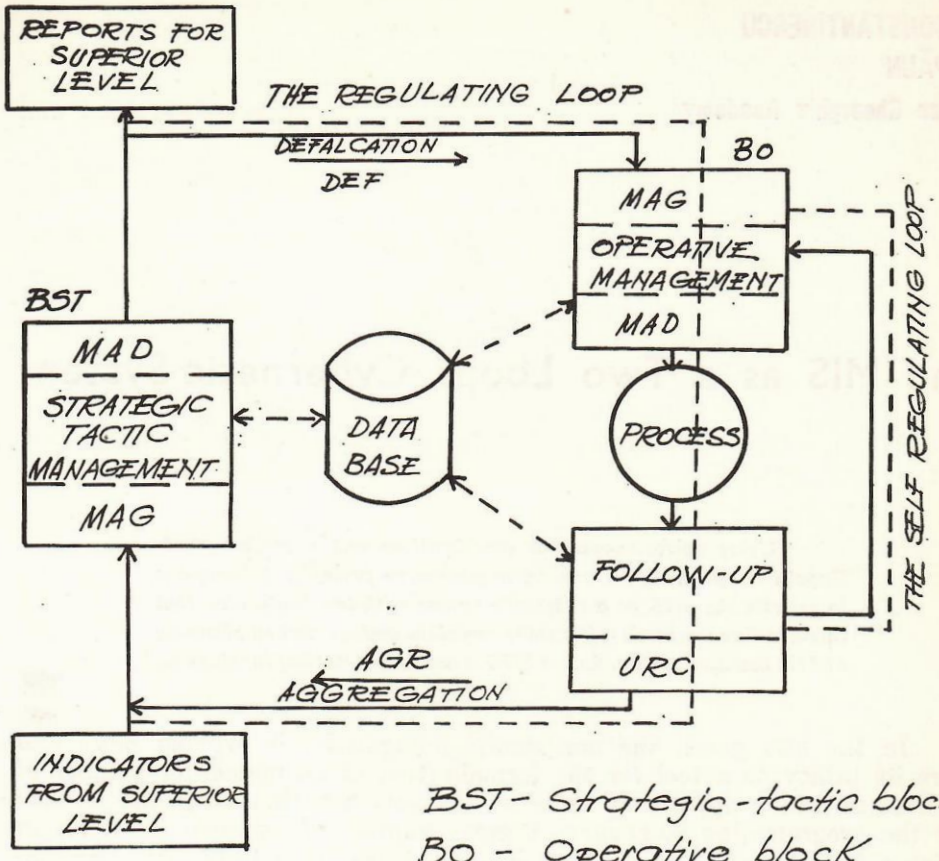

REPORTS FOR SUPERIOR LEVEL
THE REGULATING LOOP
DEFALCATION
DEF
Bo
MAG
OPERATIVE MANAGEMENT
MAD
THE SELF REGULATING LOOP
BST
MAD
STRATEGIC TACTIC MANAGEMENT
MAG
DATA BASE
PROCESS
FOLLOW-UP
URC
AGR
AGGREGATION
INDICATORS FROM SUPERIOR LEVEL
BST- Strategic-tactic block
Bo - Operative block

Fig. 1

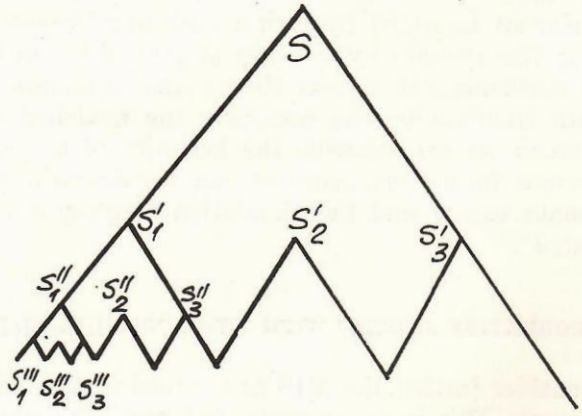

S
S'1
S'2
S'3
S''1
S''2
S''3
S'''1 S'''2 S'''3

Fig. 2

Describing the MIS as a cybernetic system with two regulating loops, with the structure of models from the two decision blocks and the control block given in Fig. 1, we can prove ([1], [2], [3]) that, in this way, we can realize the management by planning at the level of the whole pyramide and of every subpyramide of the pyramide (Fig. 2).

We shall illustrate the management by plan by the activities described in Fig. 3, which represents the foundamentation, elaboration and the execution of the yearly plan; we shall notice that one prepares two large categories of decisions.

a) Decisions concerning the yearly-plan, that is, foundamentation (determination of S_f) prelimination of the closing of a plan-period (determination of S_0) and the elaboration of the trajectory for the new year;

b) Decisions concerning the basic year for the defalcation of the plan and the correction of the deviations from the designed trajectory and eventually the modification of the trajectory between S_0 and S_f.

These decisions are concerning the continuous process of harmonisation resources and objectives, that is, the elaboration of the optimal trajectory between S_0 and S_f.

For both categories of decisions, this harmonisation has to be conceived both statical (global) and dynamic. The global harmonisation is essentially the correlation of the resources and objectives on a given global interval of

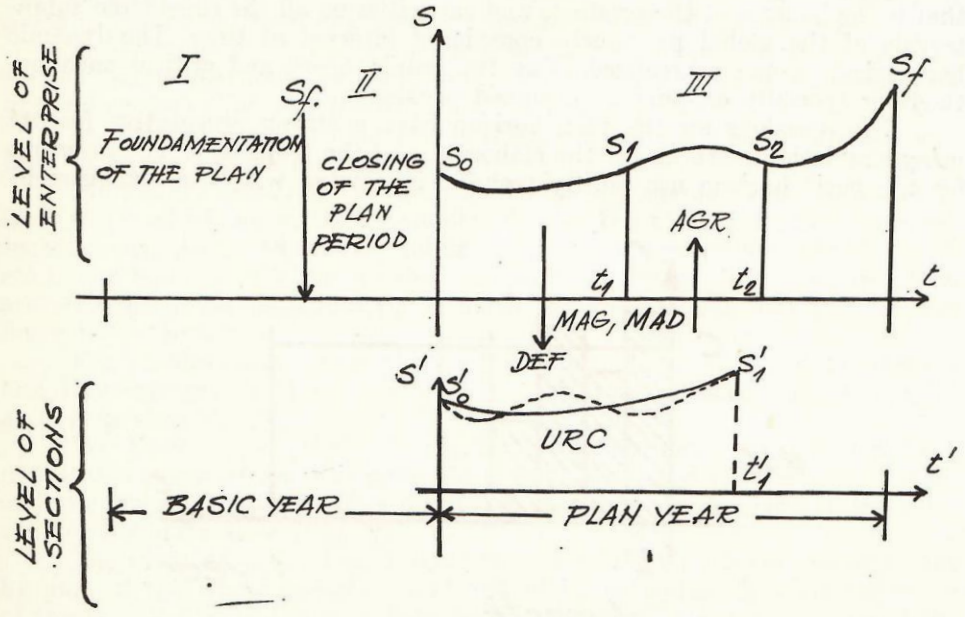


Fig. 3

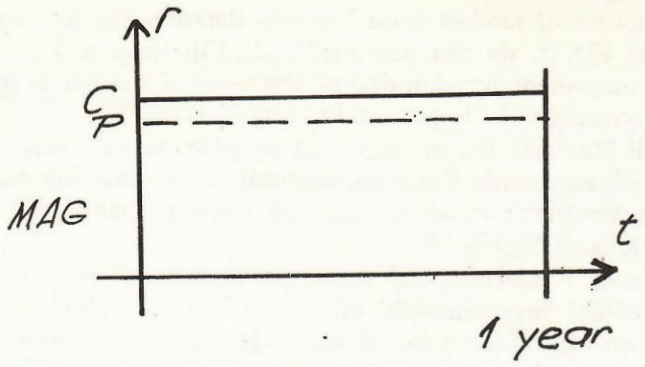


Fig. 4

time; we shall illustrate the balance of products (P) which shall be manufactured in one year using an average yearly capacity (C) for a single resource r in Fig. 4.

In the same way, we procede for all the resources, having as purpose their best use. The global harmonisation (MAG) can be realized, for instance, using linear programming.

However, in most real cases, on various intervals of the global considered period, the flow of the manufacturing may lead to botle — necks or subuses of the capacities (Fig. 5).

On this ground, it is needed also the dynamic harmonisation (MAD) that is the balance of the products and capacities on all the considered subintervals of the global previously considered interval of time. The dynamic harmonisation can be realized using the graph-theory and critical path-methods or specially on purpose designed packages.

The decisions for the plan horizon have a strong prospective feature using simulation methods for the elaboration of the trajectory. The decisions for the basic horizon use mostly technics associated with the management

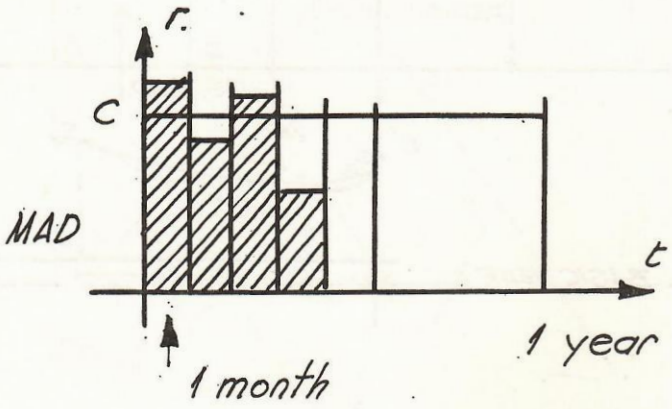


Fig. 5

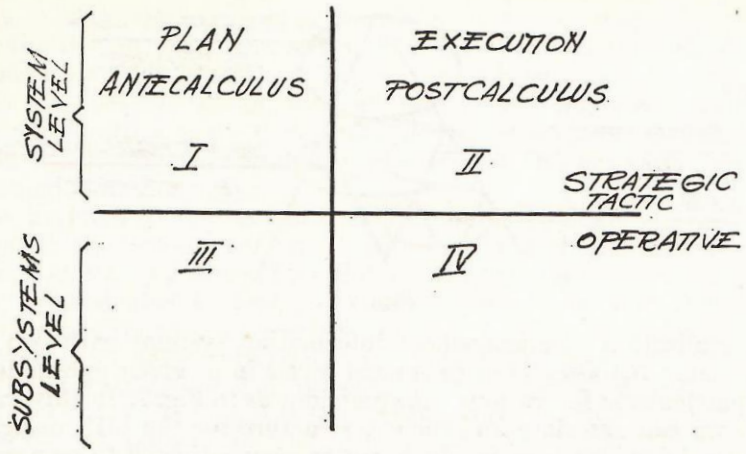


Fig. 6

by exceptions, which lead to the correction of the deviations. The elaboration of the trajectory is made with the aid of MAG and MAD; the follow-up uses the model URC. The flow of information (decisions, indicators, realizations, etc.) between the two levels of management is realized by the help of the models DEF and AGR. We shall notice that the five models are sufficient in order to prepare the long, medium and short term planning decisions and for the corrections of deviations for the pyramide in Fig. 2 and any of its subpyramides, that is, are invariant for any pairs of sequential horizons associated with any pair of sequential levels of management.

In order to prove the sufficiency of the two loops structure of MIS we shall show also that the structure of the data base can provide the data and information for the five models.

In order to give an example of this kind, we shall structure the data base in four zones as in Fig. 6.

The zones I and II are associated with the aggregated data (the states S) from the level of the whole pyramide, whereas the zones III and IV are associated with the data corresponding to all the subpyramides (S', S'', S''', etc.). For foundamentation we shall use particularly the first zone in which are structured the data and information for antecalculus and plan at the level of the pyramide.

For prelimination we shall use the zones I and II, for defalcation I and III, aggregation II and IV, and for deviations' reports the pairs I, II and, respectively, III and IV.

We shall notice that the reports (foundamentation, prelimination, defalcation, aggregation and deviations) imply the models'structure of MIS, and the reports and the models imply the structure of the data and information in the data base (Fig. 6).

The structure from Fig. 1 is as follows sufficient for the management by plan of the whole pyramide and any of its subpyramides, the structure of the models and of the data base being invariant for any pair of plan horizons and hierarchical levels.

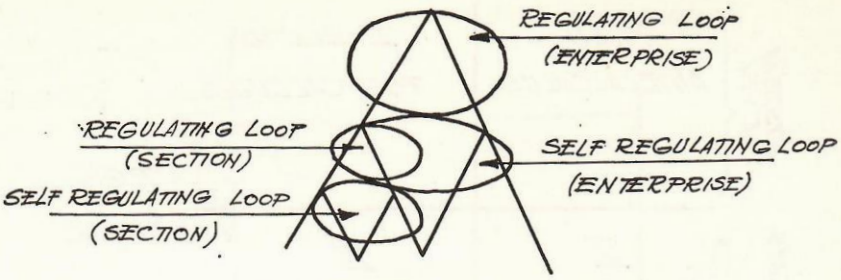


Fig. 7

The realization of management information systems with two regulating loops (class S_2) associated to several levels in a given pyramide can be obtained particularly for various subpyramides as in Fig. 7. In this way, from one hand, we can associate an efficient structure for the MIS using "prefabricates" (modular components which can be easy adjusted to be used again) as packages and data structures associated to the various levels of aggregation and from the other hand, we can associate MIS to more and more complex pyramides.

2. A MATHEMATICAL-LINGUISTIC MODEL OF THE PRODUCTION PROCESS

Let us consider the MIS with two regulating loops in Fig. 1 at the level of the enterprise. Both the first regulating loop — closed at the level of the sections — as well as the second one — closed at the level of the working places (technological lines or groups of interchangeable machines) — are of the same type, namely, the cybernetic loop described in Fig. 8. By analogy with automatics, we shall notice that the main loop is a regulating loop dealing mainly with external deviations, whereas the second loop is a self-regulating loop which deals mainly with internal deviations. Because the two loops are closed at different levels, their working time intervals are different; anyways the first one is a multiple of the second one.

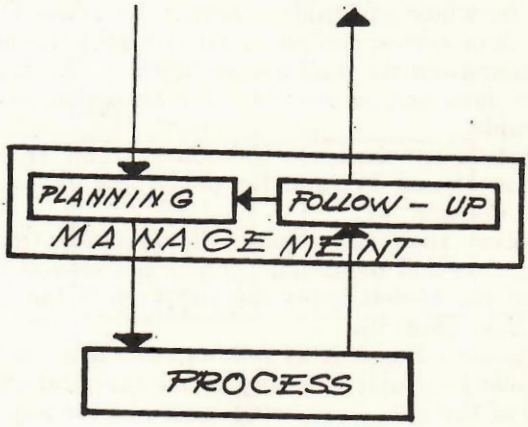


Fig. 8

The level (departments, subassemblies, the follow-up time,etc.)where we close the first loop and the second one depends from one hand on the existent conditions (equipments, follow-up procedures, etc.) for the monthly follow-up (or decades, weeks, etc.) at least of the manpower and material expences; from other hand, it depends on the level of aggregation of the information which the decision-maker can use for the semantic validation of consumption values.

The problem of the semantic validation of consumption values, taking into account the actual differences between antecalculus and postcalculus can be solved only by the supervision of the decision-maker, using the mechanism "antecalculus-decision-postcalculus"; in respect with the aggregated values, checked by the mechanism, we can validate the elementary data which by aggregation have to be in coincidence with the decided "key" values.

In what follows, we shall develop the approach of the production process by the help of the formal languages from [11]. In this way we can model the management activity by the help of automata. More exactly, we shall investigate the place in the Chomsky hierarchy of the automata which realize the *planning* of the evolution of the process and the activity of *following-up* the trajectory in order to eliminate the deviations. In this way, a new field — the economical cybernetics — can take advantage from the mathematical linguistics. (For another practical application of the formal languages theory see [10]). On this base we shall infer the necessity of describing MIS as system with at least two cybernetical loops.

Definition. We call a *production system* the 7-touple,

$$\sigma = < O, R, M, A, T, I, S >$$

with O, R, M, A, T, I, S, defined as follows:

a) $O = \{a_1, a_2, \ldots, a_p\}$ is the set of finite products;

b) R is the set of resources

$$R = \{r_1, r_2, \ldots, r_n\};$$

c) In any unit of time (the time unit corresponds to a choosed level of aggregation, associated to a hierarchical level of management) the available amount of the resource r_i is m_i, $m_i \in \mathrm{R} \cup \{\infty\}$.
Then $M = \{m_1, m_2, \ldots, m_n\}$;

d) The production process transforms resources in R to finite products in O, in concordance with some technological prescriptions. Let us denote by 1, 2,..., q_i; the activities with the duration of one unit of time, which are performed to produce a product a_i from O. Let U_i be the set of the pairs of the form (k, j), $1 \leqslant k, j \leqslant q_i$, representing activities which must be performed succesively in this order in the manufacturing of a_i. Then, the technological receipt of a_i can be written as a graph

$$\Gamma_i = (\{1, 2, \ldots, q_i\}, U_i).$$

We suppose that q_i is allways the last activity, therefore when q_i is performed, an item of a_i is finished. Among the activities we can consider vertices in the graph describing the preparation of manufacturing from designing and technology point of view as well as from the materials'supply or colaborations.

We consider that the activities represented in Γ_i are performed in an exact unit of time. Hence, two activities, k, j, such that $(k, j) \in U_i$, may not be performed in the same unit of time.

For any vertes j in Γ_i, a n-touple is considered

$$L_{i,j} = (L_{i,j,1}, \ L_{i,j,2}, \ ..., \ L_{i,j,n}),$$

where $L_{i,j,k}$ represents the amount of r_k consumed in the activity j from the manufacturing cycle of a_i. Let

$$L_i = (L_{j,1}, \ L_{i,2}, ..., \ L_{i,q_i})$$

be the sequence of the n-touples associated to the graph Γ_i. We denote the set

$$\{(\Gamma_1, \ L_1), ..., \ (\Gamma_p, \ L_p)\} \text{ by } A;$$

e) For any $a_i \in O$ a strictly increasing sequence $T_i = (t_1^i, \ t_2^i, ...,)$ of positive integers is considered. At the moments t_j^i, $j = 1, 2, ...,$ one item of the article a_i must be finished. Then $T = \{T_1, \ T_2, ..., \ T_p\}$ is the set which describes the delivery plan;

f) In order to describe the activity of management by planning of the production process, we must define the *state* of the system. A state will describe the production system in one unit of time, therefore a state must identify the products on which the system is acting, and also the activities from the graph corresponding to these products performed in this interval of time. These elements determine the resource consumptions and the values of indicators.

A state will be represented by a 2-rows matrix $s(2 \times m)$ where

1) $s(1, i) \in \{1, 2, ..., p\}$, is an article from O,

2) $s(2, i) \in \{1, 2, ..., q_{s(1,i)})\}$, is an activity from $\Gamma_{s(1,i)}$ performed at this moment.

Let us denote by S the set of all the states. Considering that in every unit of time the system is acting on at most m_0 articles, for a fixed m_0, it follows that S is a finite set.

Any string is S^* (S^* denotes the free monoid generated by S under the operation of concatenation and with the identity element λ) is called *manufacturing process*. Obviously S^* is an infinite set.

g) The manufacturing activity must be developed in such a way that the graphs in A, the capacity limitations of M and the delivery plan of T to be accomplished. Moreover, the management must ensure a continue harmonisation of the objectives and the resources. On the manufacturing process act consequently restrictions of the form of some indicators.

$$f : S^* \to \mathrm{R}.$$

I represents the set of these indicators,

$$I = \{f_1, f_2, ..., f_s\}.$$

The evolution of the system σ can be described by strings of states. A correct evolution of σ determines some strings for which the conditions imposed by A, M, T hold. The values of indicators depends on the form of

the manufacturing processes, therefore I represents the semantic constituent of the system. In conclusion, we are interested in the syntactic investigation of the sets of manufacturing processes, in the determination of the generative complexity of these sets. In this way this problem of management becomes a linguistical problem.

3. THE MANAGEMENT OF THE SYSTEM

From the activities which have to be performed in the process of management we focuse on two: *the planning* and the *follow-up*. The former is a prospective activity and is dealing with the determination of the trajectory to be followed by the process. The later is a retrospective activity; it controls whether the system evolution is a planned one. The planning determines a set of strings in S^*, therefore it is a generative activity in the sense of formal languages theory. The follow up is a parsing activity; it must decide whether a given string of states (the real evolution of system) is a string (or an initial substring of some string) in the planned set.

Let V be a vocabulary. We define, for any $x \in V^*$, the set

$$init(x) = \{z \mid z \in V^*,\ x = zy \text{ for some } y \in V^*\}.$$

For $L \subset V^*$ we put

$$init\ (L) = \bigcup_{x \in L} init(x).$$

Consequently, the follow-up procedure must decide whether or not a given string belongs to a language of the form $init(L)$.

The data base of the MIS for the management of the production pocess must contain in the files the sets O, R, M, A, T, distinguished trajectories from S^*, and, among the working algorithms, the procedures for computing the functions from I. Because the indicators are defined for the states in S and also for strings of states of arbitrary lengths, among the working algorithms of the data base we must have also procedures to pass from $f(s_1),..., f(s_k)$ to $f(s_1,..., s_k)$ hence to compute some functions $h_1,..., h_s$, h_i: $R_k \to R$ for which $f_i(s_1,..., s_k) = h_i(f_i(s_1),..., f_i(s_k))$. The algorithms for the computation of h_i can be more or less complicated according to the kind of indicators. Many of these functions are linear. For instance, the consumption of a resource r_u in a state s_i is

$$c_u(s_i) = \sum_{j=1}^{m} L_{s_{i(1,j)},\, s_{i(2,j)},\, u}$$

and, obviously,

$$c_u(s_1,..., s_k) = \sum_{i=1}^{k} c_u(s_i).$$

We say that a state s is *feasible* if and only if $c_u(s) \leqslant m_u$ for any r_u.

A manufacturing process ϕ is said to be *correct with respect to A* if and only if there is a sequence of graphs in A such that:

1) for any column, $s(1, j)$, $s(2, j)$, in a state of ϕ there exists a vertex $s(2, j)$ in a graph $\Gamma_{s(1,j)}$ from the sequence;

2) this correspondence — let us denote them by γ — is a bijective function;

3) if $(k, j) \in U_i$, then there are two states s and s' appearring in this order in the sequence, and there are r and t such that $s(1, r) = s'(1, t) = i$, $s(2, r) = k, s'(2, t) = j$, and $\gamma(s(1, r), s(2, r)) = (k, \Gamma_i), \gamma(s'(1, t), s'(2,t)) = (j, \Gamma_i)$.

Let $P_1(\sigma)$ be the set of all the manufacturing processes correct in respect with A. We consider also the languages $P_2(\sigma)$ being the set of the manufacturing processes belonging to $P_1(\sigma)$ constructed from feasible states, $P_3(\sigma)$ being the set of the manufacturing processes from $P_1(\sigma)$ for which the conditions imposed by T hold, and $P_4(\sigma)$ being the intersection of the two last languages. In this way we defined languages which contain strings of states accomplishing the conditions represented by A, M, T. As we allready saw, the indicators are acting on these strings as restrictions. They depend on the form of the manufacturing process. Moreover, the generation of a language and the parsing essentially depend on the place of the language in Chomsky hierarchy. Consequently, in what follows, we are concerned only with the syntax of the previous languages. We shall investigate the place of these languages in Chomsky hierarchy, hence the type of automata which generate them.

4. THE NECESSITY OF DESCRIBING THE MIS AS A CYBERNETIC SYSTEM WITH AT LEAST TWO REGULATING LOOPS

In the sequel we shall consider that the notions concerning the formal languages theory from [5] and [12] are known. The denotations are also those of [12]. We specify only that we describe by $\mathcal{L}_i$, $i = 0, 1, 2, 3$ the four families of languages in Chomsky hierarchy: $\mathcal{L}_0$ is the family of recursively enumerable languages (generated by Turing machines), $\mathcal{L}_1$ is the family of context — sensitive languages (generated by linear bounded automata $\mathcal{L}_2$ is the set of context-free languages (generated by push-down automata) and $\mathcal{L}_3$ is the family of regular languages (generated by finite automata). The following proper inclusions are known

$$\mathcal{L}_3 \subset \mathcal{L}_2 \subset \mathcal{L}_1 \subset \mathcal{L}_0.$$

In [11] are proved the following theorems.

Theorem 1. For any production system we have $P_1(\sigma) \in \mathcal{L}_1$.

Theorem 2. For any σ, the language $P_2(\sigma)$ is of the same type as $P_1(\sigma)$. If any T_i can be written as the union of a finite set and of a finite family of arithmetical progressions, then the language $P_3(\sigma)$ is of the same type as $P_1(\sigma)$.

The last condition describes the requirement that the deliveries have to be rhytmically.

Therefore, the type of $P_1(\sigma)$ determines the type of the other languages. The indicators also do not modify the type of the language $P_1(\sigma)$.

Let us suppose that an indicator f_i is controlled after each unit of time. Denoting by S_i the set of states for which the condition imposed by f_i holds, then we obtain the language $P_1(\sigma) \cap S_i^*$. Because S_i^* is a regular language, we obtain a language of the same type as $P_1(\sigma)$.

Suppose now, that f_i is controlled after k units of time. We consider, in this case the set S_i of the strings $x \in S^*$, $|x| = k$ for which the condition imposed by f_i holds. The language $P_1(\sigma) \cap S_i^* \; init(S_i)$ is again of the same type as $P_1(\sigma)$.

Another kind of indicators can be the following: any substring of length v of a string of state has a certain property. In the aim to clarify this situation, let us define the following operation: let $L \subset V^*$ be a finite language. We define the language L^∞ as the smallest language $L' \subset V^*$ for which

i) $L \subset L'$

ii) if $x \in L'$ and $x = x_1x_2$, $x_2a \in L$ for some $x_1, x_2 \in V^*$, then $xa \in L'$.

Theorem 1 in [10] proved that the language L^∞ is regular for any L.

Let us consider the set S_v of those strings $x \in S^* \mid x| = v$, for which the considered condition holds. We obtain the language $P_1(\sigma) \cap (S_v \cup init\ (S_v)$ of the same type as $P_1(\sigma)$.

As an example of such indicators, let us consider "the index of resources utilization". For a resource r_i for which $m_i < \infty$, let $v_i \in \mathbf{N}$ and $u_i \in \mathbf{R}$ and let S_{v_i,u_i} be the set of states $s_1s_2...s_{v_i}$ for which

$$m_iv_i - \sum_{j=1}^{v_i} c_i(s_j) \leqslant u_i.$$

The utilization of the resource r_i can be controlled by means of the constants u_i, v_i We obtain the language $P_1(\sigma) \cap S_{v_i,u_i}^\infty$ which is of the same type as $P_1(\sigma)$.

Let us notice that all the previously considered restrictions imposed to the manufacturing processes — the restrictions determinated by M, T, I, — can be expressed by the help of the intersection of the language $P_1(\sigma)$ with a regular language. The family $\mathcal{L}_3$ is closed under intersection, therefore, by imposing two or more simultaneous conditions, we obtain a language of the same type as $P_1(\sigma)$.

Theorem 3. If one of the following conditions holds, then $P_1(\sigma)$ is not a context-free language.

1) there is at least one product a_i such that Γ_i contains at least three vertices;

2) there are at least two products, the corresponding graphs containing at least two vertices.

The theorem can be completed by the following remark. If any Γ_i in A contains exactly one vertex, then $\Gamma_i = (\{1\}, \emptyset)$, and, therefore, $P_1(\sigma) = S^*$. Consequently, in this case $P_1(\sigma)$ is regular. This situation occurs at a high level of aggregation, when the time unit is larger than the manufacturing duration of any article. This is the case, for instance, of the yearly plan of the enterprise.

Theorem 4. If in any moment the production system is acting on at most α articles — for a fixed α — then the language $P_1(\sigma)$ is regular.

The hypothesis of theorem 4 expresses the condition that the semi-finished stock to be bounded.

Therefore MIS can be generally simulated only by the help of linear bounded automata. But, it is known that there are many algorithmicaly unsolvable problems and many unsolved questions about the linear bounded automata (context — sensitive languages). Moreover, the handling of these automata implies a high degree of dificulty.

In this way we have obtained a theoretical proof of the complexity of MIS, that is, of its designing and implementation.

The type of $P_1(\sigma)$, $P_2(\sigma)$, $P_3(\sigma)$, $P_4(\sigma)$ depends only on the complexity of the production process at the considered level of aggregation (the size of the time unit): as the level of aggregation is higher (the time unit is larger) as the previous languages are simpler.

The context-sensitive languages are recursive. The follow-up activity decides whether a given string is in a language of the form $init(L)$, generally with $L \in \mathcal{L}_1$.

Theorem 5. It is an algorithmicaly unsolvable question whether or not a string $x \in V^*$, belongs to a language of the form $init(L)$ with $L \in \mathcal{L}_1$.

Proof. In [5] is proved the following theorem. Let $L_1 \in \mathcal{L}_0$, $L_1 \subset V^*$ and let a, b be two symbols which are not in V. There exists L_2 in $\mathcal{L}_1$ such that:

i) $L_2 \subset \{w \mid w = a^i bx,\ i \geqslant 0 \text{ and } x \in L_1\}$,

ii) for any $x \in L_1$ there is some $i \in \mathbf{N}$ such that $a^i bx \in L_2$.

Let us denote by $mi(x)$ the mirror image of the string x, $x \in V^*$. Because the families $\mathcal{L}_0$, $\mathcal{L}_1$ are closed under the operation mi it follows that $mi(L_1) \in \mathcal{L}_0$ and $mi(L_2) \in \mathcal{L}_1$ (we denoted by $mi\ (L)$ the set $\{mi(x) \mid x \in L\}$).

Let us suppose that $init(mi(L_2))$ is a recursive language. The family of recursive languages is closed under intersection and the family $\mathcal{L}_3$ is contained in this family. Consequently

$$L_3 = init(mi(L_2)) \cap V^*\{b\}$$

is a recursive language.

But $L_3 = mi(L_1)\ \{b\}$, hence $mi(L_1)\ \{b\}$ is a recursive language. Therefore $mi(L_1)$ is a recursive language, thus L_1 is a recursive language too. This contradicts the fact that there are recursively enumerable lanuages which are not recursive.

Hence the follow-up can not be made with a single algorithm (does not exist a general procedure) excepting the case if we overpass the level of context-sensitive languages. If we recall, for instance, that the emptiness and the finiteness problems are not algorithmic solvable for the context-sensitive grammars, clearly results the necessity of structuring MIS by means of automata classes more restrictive than linear bounded automata.

A possible way to realize this is suggested by theorem 4: imposing some restrictions on the production process in order to simplify the management process. However, this is a particular solution; we are interested to find a general MIS structure such that any production system can be managed.

According to theorems 1, 2, 3 and the previous remarks, this can not be made unless we proceed to the segmentation of the process (defalcation), therefore by the cleavage of the graphs in A and the management of each such subprocess separately, approaching the process from the largest aggregation level to the execution. The follow-up, for instance, must be monthly realized at the superior level and at the level of 5-days, week, decade, etc. must be correspondingly implemented by segmentation at inferior levels.

In this way occur at least two levels of management: at least one for the complete production process having a great time unit and, consequently, simplified graphs, and at least one for the graphs obtained by defalcation of the process, having smaller time units.

In conclusion, at least two management levels are necessary between which an aggregation — defalcation mecanism must be realized. At every management level we must have a cybernetic loop in order to realize the follow-up activity. Therefore, the MIS for modelling the management by planning must be a cybernetic system with at least two regulating loops.

From the existence of *exactly two* regulating loops does not necessary follows that the activities of MIS can be accomplished by push-down automata or by finite automata. However, by segmentation we obtain advantages in using this king of automata by decreasing the number of states, the number of moving rules, and, particularly, by standardization easier to be generalized in designing and implementation, having associated packages and standard data structures.

In this way have shown that the MIS structure described in Fig. 1 is also necessary and sufficient. Obviously, the existence of two decision blocks do not necessarily implies the existance of exactly two levels of management. However, the simplification of the hierarchical structure of management levels, and, respectively, the bringing-up of the management to the process is realized by the closing of the regulating loop nearer to the process that is using smaller time units (decade, week, etc.), by the help of more complex equipments; in this way the self-regulating loop can become a robot-loop which manages the process in real time (Fig. 7).

REFERENCES

1. CONSTANTINESCU P: *Informatica, Bazele metodologice ale proiectării și implementării sistemelor informatice.* Academia "Ștefan Gheorghiu", București, 1974.
2. CONSTANTINESCU P.: *Sistemul informatic pentru conducere.* Academia "Ștefan Gheorgiu", București, 1976.
3. CONSTANTINESCU P., NEGOIȚĂ C. V.: *Sistemul informatic-model al conducerii și procesului condus.* Editura tehnică, București 1975.
4. CONSTANTINESCU P., PĂUN G.: *The Cybernetic Structure of MIS Described by Automata Theory,* The 3-th International Congress of Cybernetic and System Theory, București, 1975.
5. CREANGĂ I., REISHER C., SIMOVICI D.: *Introducere algebrică în informatică. Limbaje formale.* Editura Junimea, Iași, 1974.

6. KORNAI J.: *Anti-equilibrium.* Editura ştiinţifică, Bucureşti, 1974.
7. MALITA M., MARCUS S.: *Matematica şi abordarea sistematică în ştiinţele omului.* Studii şi cercetări de calcul economic şi cibernetică economică, no. 3, 1975.
8. MALITA M., ZIDAROIU C.: *Matematica organizării,* Editura tehnică, Bucureşti, 1974
9. MARCUS S.: *Linguistics as an pillot science. Current trends in linguistics.* Edited by Th.Sebeok, vol. 12, Mouton, Paris, 1974.
10. PĂUN G.: *O problemă de planificare a itinerariilor unui punct mobil într-un graf şi rezolvarea ei cu ajutorul teoriei limbajelor formale.* Studii şi cercetări matematice, 25.5. 1976
11. PĂUN G.: *A formal linguistic approach of the production process.* To appear in Foundation of Control Engineering, Poznan, Poland.
12. SALOMAA A.: *Formal languages.* Academic Press, New York and London, 1973.

SYSTEME INFORMATIQUE EN TANT QUE SYSTEME CYBERNETIQUE A DEUX BOUCLES DE REGULATION

Utilisant les modalités de la recherche opérationnelle et linguistique-mathématique concernant les processus de production, nous allons démontrer le nécessité d'établir la structure des systèmes informatiques pour la direction, en tant que système cybernétique ayant au moins deux boucles de régulation. Aussi, nous allons prouver que la structure de deux boucles est suffisante pour mettre en modèle la direction au moyen du planning et le procesus dirigé. De ce fait, elle s'avère suffisante pour diriger une piramide et toute sous-piramide de la piramide-section, entreprise, centrale, branche, système territorial, etc. Ces systèmes informatiques ont été nommés dans [1], [2], [3] systèmes de classe S .

ИНОФОРМАЦИОННАЯ СИСТЕМА КАК КИБЕРНЕТИЧЕСКАЯ СИСТЕМА С ДВУМЯ СКОБАМИ РЕГУЛИРОВАНИЯ

Используя соображения операционных и лингвистическо-математических исследований относительно производственных процессов, мы продемонстрируем необходимость структурации информационных систем для управления как кибернетические системы по крайней мере с двумя скобами регулирования и тот факт, что двухскобовая структура достаточна для моделирования руководства управляемым планом и процессом. Итак, достаточно управлять пирамидой и любой подпирамидой из пирамида цех, предпритяие, об"единение, отрасль, территориальная система и т.д. Эти информационные системы были названы в 1], [2], [3] классовой системы S_2.

27. VI. 1976