

V.A. GORBATOV THEORY OF CHARACTERIZATION – PRINCIPLES AND EXAMPLES

Tadeusz KRUPA

Faculty of Management, Warsaw University of Technology, Warsaw, Poland
e-mail: t.krupa@wz.pw.edu.pl

Abstract: Characterization theory was developed in the 1970s by a member of the Russian Academy of Sciences, a prominent cybernetics V.A. Gorbатов, and became the nucleus of an international school. Many of today's academy graduates make a new generation of scholars, more than 150 doctors and assistant professors, developing a new descriptive theory of cybernetic complex systems, based on the canon of the so-called characterization principle, binding the sets of Ψ_a functioning models and Ψ_b structure models using the paradigm of monotone mappings of the considered system. This article presents an overview of the following problems: (1) the issues of functional–structural connection systems from the point of view of their design correctness; (2) the basic postulates of the characterization principle; (3) the nature of interaction between system objects; (4 and 5) mechanisms and functions for initiation of the operation; (6) the mechanisms of control and reaction functions; (7 and 8) the analytical form of initiation function and a network initialization function and (9) the axiom of extensionality (J), feasibility (R) and compatibility (Z).

Keywords: functional model, structure model, atomic predicate, characterization principle, operations and functions of initiation, control and response functions, axioms of clarity, feasibility and compatibility.

1 Introduction

The essence of the characterization principle is expressed in mutual interpretation of the Ψ_a functioning model of a certain object (resource) with the Ψ_b structure model.

Reciprocal interpretation of Ψ_a , Ψ_b models is achieved by the selection of universal rules of "proper" functioning model Ψ_a and expressed in the Ψ_b model structure. Rules of proper functioning are determined by the obligatory, forbidden and neutral figures of Ψ_a functioning model. Due to the graphic form of their structure, these figures are often called graph figures.

The idea of the proper functioning of a specific object [3, 4] can be expressed through mutual interpretation of Ψ_a Ψ_b models in the following way:

- object functions adequately to its structure described by Ψ_b model structure and
- object structure is adequate to its desired way of functioning described by Ψ_a functioning model.

A basic model of the characterization theory can be described as:

$$\langle \Psi_a, \Psi_b, P_o(\Psi_a, \Psi_b) \rangle \quad (1)$$

where $P_o(\Psi_a, \Psi_b)$ is an atomic predicate characterizing the interpretation possibility of Ψ_a functioning model in the categories of Ψ_b structure model.

Satisfactory interpretation is obtained if Ψ_a functioning model (e.g. solution model) is stored in the form of proper functioning axioms expressed in the form of obligatory, forbidden and neutral graph figures. Such an approach allows creating flawless (and also optimal) solution design tasks while avoiding a time-consuming procedure of testing directly the structure of (prototype) solution.

The difficulties in verifying the proper functioning of the object by applying tests lies in the fact that, in real terms, we are not able to see all the possible variants of the structure. The unlimited storage of the object in question or the memory of its environment (the environment in which it operates) is practically an obstacle.

Known methods of system design are the "trade" answers how to create structural network architecture solutions, how to build machinery and equipment and how to design processors and computer programs. Universal design method for all structures does not exist, although from system positions, "trade" struc-

tures become more and more similar to each other (homeomorphic).

The computer virtual space is already able to hold “almost everything” what is coming into existence from the imagination of engineers. In this context, it is reasonable to formulate universal reference examples or functional–structural paradigms, understood as special design theories. Characterization principle determines the general direction of theoretical works but Ψ_a , Ψ_b models and atomic predicates are constructed for a specific class of tasks and their resources.

2 Functioning and structure

A relationship between functioning and resource structure is considered as a necessary for the whole of analytical synthetic methods for decomposing and solving any design tasks.

The first theoretical solutions to the structural–functional relationships were developed by V.A. Gorbатов in his works on a partially ordered sets, particularly on the synthesis of logical structures and their imaginary drawing of Hasse diagrams [3,4]. Continuation of these studies led to the formulation and development of the characterization principle for the task of designing complex technical and organizational systems [5-7, 11, 16].

The nature of characterization principle is determined by the Ψ_a functioning model connected with the Ψ_b structure model using $P_o(\Psi_a, \Psi_b)$ predicate, characterizing the interpretation possibility of functioning model in the categories of Ψ_b structure model. In case of designing the structure of interested us system, the characteristic principle comes down to interpretation of the Ψ_a model (the assumed logical, temporary and priority connections) in the Ψ_b model categories of the resource structure of this system. Such an interpretation is achieved through axiomatic proper functioning and graph figures of Ψ_a model.

The necessary condition for the correctness of project is the presence and absence of homeomorphic graph figures adequately to obligatory and forbidden graph figures. The fundamental problem in designing constitutes synthesis and seeking such figures. In return, we are given the opportunity to avoid errors in the system design project without the need to time-consuming and costly testing of structures with the complexity of P expressed in the form:

$$P = \prod \kappa_i! \quad (2)$$

where κ_i is the number of elements of the storage device and the twig-Hasse (e.g. for seven 5-elements twigs of Hasse $P = 38.8 \times 10^{12}$).

In case of system resource interpretation, the theoretical interpretation of specific changes in the value of its input and output characteristics is determined by the structure and functioning of the resource. System resources will function properly if between their functioning rules and structure a mutually synonymous interpretation is determined including the following assumptions:

- system functions adequately to its structure, and
- resource structure is adequate to its desired way of functioning.

Examining functional–structural connections constitutes the basic paradigm of the theory and practice of the systems design. Software engineers develop appropriate structures to achieve the desired functioning of existing or hypothetical objects. Design theories compose to some extent “trade” answer how to create architectural solutions, how to build machinery and equipment, how to design processors and computer programs. There is no and it may not be a universal method for designing all structures, although from the point of system positions, the “trade” structures become more and more alike. Virtual computer space is already able to hold “almost all”, which comes from the imagination of engineers.

In this context, it is reasonable to formulate universal examples or functional–structural paradigms, understood as specific design methodologies. Characterization principle sets the general direction for methodological work; Ψ_a , Ψ_b models are constructed for a specific class of tasks.

Functional–structural paradigm is the characteristic for the project activity, which is looking for a structural response to a specific functional problem. This paradigm is in balance of structural–functional paradigm, where the functional interpretations (interpretations in the field of functional) are derived from the original structure. Structural–functional paradigm focuses equally on archeology, astronomy and art. In many practical situations, we cannot separate or indicate what is primary: functioning or structure?

The functioning of any object including, for example, any engineering system, is determined by the appropriateness which takes place between its structure (struc-

tural–static characteristics) and functioning (dynamic characteristics). It is possible to examine this suitability at different levels, in particular:

- at the level of abstract models and theories,
- at the level of program or physical simulation models,
- on the level of testing models and prototypes, and
- during the observation of working systems.

The practical importance of this type of analytical work includes giving directions for reducing workload and functionally optimized systems search. Such a feature is particularly important in the process of computer-aided design work applied to the methods of designing engineering systems. These are the methods that allow “browsing” through many simplified indirect solutions until the final versions in the form of complete projects have been generated.

Functional–structural paradigm raises the hypothesis of a constructive proving the unity of structural-functional systems, understood as a theoretical ability to design functionally and structurally correct systems. For the purposes of this hypothesis:

- functional validity of the system should be understood as the ability to define a finite set of primary functional rules of the system for which will be created such a practical possibility of their structural realization that the system will implement only primary functional rules,
- structural validity of the system should be understood as the ability to define a finite set of secondary rules of this model in the form of functional obligatory, forbidden and neutral graph figures that determine the functional validity of the system in terms of its set of primary functional rules.

Structural accuracy of the system is ensured if the model of functioning will meet these requirements:

- there will be no forbidden figures,
- obligatory figures occur,
- neutral figures will be used in the equivalent transformations to simplify the system structure without compromising its functionality.

3 Characterization principle in a specific project

Efforts to find the optimal variant of solution implies the need to generate and evaluate the set of all possible

solutions that could potentially arise in a particular problem situation (in a given objective area) and the applied method. In the most practical design problems, the number of solution set elements grows in a combinatorial way, thus, any practical chances of finding “optimum solution variants” by scanning each of them fail. Therefore, it is necessary to use appropriate “simplifications” narrowing down the search space and accelerate the evaluation process. This becomes especially important in a situation where the complexity of objects grows to the order of magnitude of hundreds or thousands of items or if there is a need for a “smooth” change of the level of detail in analyzed situations. In such cases, we may be helpless against the multidimensionality of objective area.

Universal rules of proper functioning are expressed mostly by the so-called obligatory, forbidden and neutral figures operational model. These figures are also called graph figures because of their graphical form.

Obligatory figures of the functioning model are abstract constructions, which as homeomorphisms should occur in the functioning model under the “threat” of their incorrectness. Forbidden figures of the functioning model should be easily identifiable objects. Isolation or dispersion (in the functioning model) assures the functional correctness of the object.

Neutral figures are used to make simplification transformations of the functioning model, which do not result in forbidden and obligatory figures. Neutral figures play the role of “system builder” in this proceeding that can be removed or added without affecting the validity operation of the entire object. Elimination of neutral figures to ensure functional correctness leads to an effective simplification of the object model and, consequently, to reduction of the combinatorial computations.

In case of any object (resource), the theoretical realization of certain changes in the value of its input and output channels (features) is determined by its structure (hierarchical memory) and its functioning. An object will operate correctly if there is a mutually synonymous interpretation expressed between functioning rules and structure, which might be determined on the following assumptions:

- object (resource) functions adequately to its structure and
- object structure (resource) is adequate to its desired way of functioning.

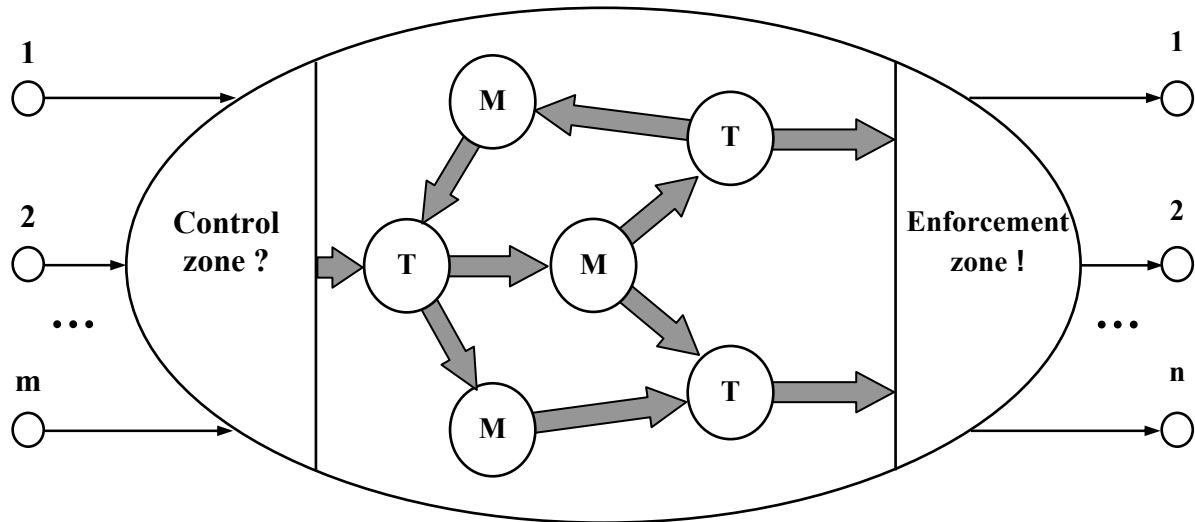


Figure 1. Object as a finite state machine

The study of functional–structural connections constitutes a basic paradigm of interests in management sciences (formerly the organization and management sciences) born of engineering activities. It is essentially a domain of theorist and designer engineers developing appropriate structures to achieve the intended operation of existing or hypothetical objects.

From the methodological point of view, the application of characterization principle for solving design problems makes a very promising approach that can be successfully used, provided that for a given problem area will be created an adequate characterization theory in the form of a basic model of the theory (1).

4 Objects, channels and channel operations

The theory of given physical reality sets out the principles of this reality existence through the system of semiotic signs being a synthesis (model) of the operation specifications.

The term “object” is used for modeling a relatively constant structure of interacting facts. Object in the sense of external behavior similarity can also be interpreted as a finite state machine (Fig. 1). The way of combining objects into the configurations of objects (Fig. 2) and the internal mechanism of interaction processes of the object make the main differences. On the basis of the inspection results, operation carried out in the control zone at input channels in the object, appropriate operations of forcing its output channels are taking place in the excitation zone. It is possible

to distinguish in the object m input channels and n output channels and certain conventional procedural interaction zones. Each channel is related to a finite alphabet of the channel states. The zones of procedural impacts:

- control zone states of the input channels (designation: ?)
- enforcement zone states on the output channels (designation: !)
- transformation zones of the interaction results (processing zone) connecting the two aforementioned zones (designation: T) and
- object memory zones (designation: M).

Object configuration creates a parent object. In Figure 2, such an object is O_5 whose input and output channels (channels $K_{1,0}$ and $K_{1,5}$) connect internal objects (objects O_2, O_4) with the environment. Any output object channels can be an input channel of one or more configuration objects or its output channel. Object channels are capable of autonomous memorizing their own state. Object interpretation of reality can be carried out in such a way that the entire internal memory of any object will be expressed in the form of its output channels and its output channels of internal objects.

A subset of the output channels KW_Y of object O_i will be called characteristic channels of this object. Each channel can be a characteristic channel of only one object and at the same time can be commutated with input channels of many other objects (output channel object can be used at the same time as its input channel). This situation is illustrated in Fig. 2.

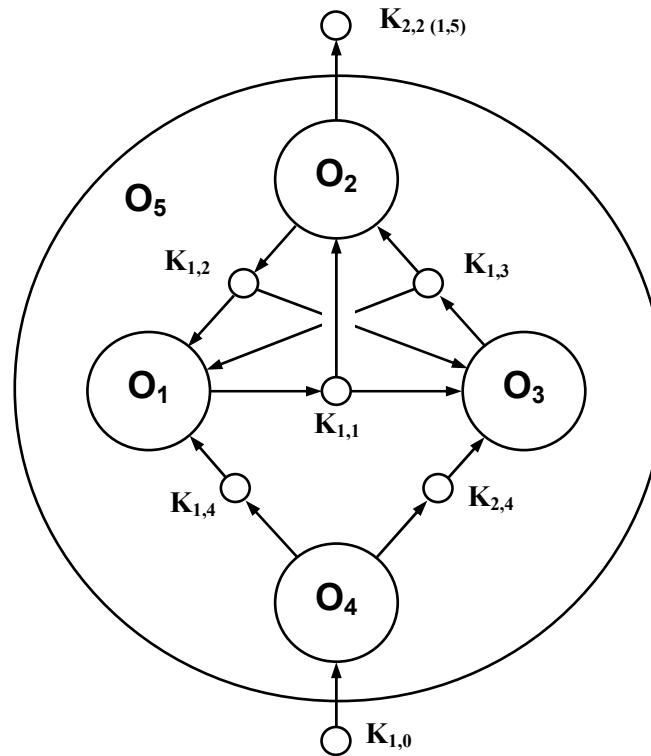


Figure 2. Object configuration

Each characteristic channel of object O_i is associated with repertoire of states S determining the all possible states of the channel. The characteristic channel of object O_i may be at any point of time only in the S_i state belonging to the repertoire of the channel states.

The status of the characteristic channel object O_i is defined by the enforcement procedure carrying out the impact of this object on its environment.

The internal processing of the object is directed (described) by the operation procedure of the object, which in general it may assume a stochastic (random) character. The reading-out of input channels states and excitation in accordance with its model of functioning and the states correctness of output channels should be ensured by operating procedure for the object at any opportune moment. The functioning procedure for operation of object should accomplish the constitutive goal of the object.

5 The beginning of operation

The implementation of discreet process is determined by the starting (initiation) of appropriate enforcement operations and control of events of this process. The purpose of initiation is only the “recovery” of procedure, which executes on the channel object the en-

forcement of event (change of state) or the control of its occurrence (changes observation status).

Initiation of operations on channels is carried out by the initialization function network (Section 7). This approach is achieved by:

- separation excitation and control processes from internal computing processes of the function initialization system,
- the possibility of constructing multi-level (hierarchical) control systems,
- the possibility of using the modeling apparatus of concurrent processing for the design of distributed control systems for objects of distinguished sets of input and output channels on which there are implemented allowable sequences of events.

Because of the separation of procedures implementing excitation and control operations from the computational processes of initiation function system, it is possible to use the multiprocessor solutions (distributed or implemented in a single computer, e.g. in a parallel architecture). In this case, each enforcement or control operation may correspond to one physical processor controlling all parameters of the operation. This type of solution also allows easily changing the time parameters and priorities of events during the initiated operation.

7 Control and response functions

The result of the operation of enforcement procedure Q_u^1 or control procedure P_u^2 is the control function model of C_u . The result marked as $[Q_u^1]$ of enforcement procedure Q_u^1 indicates a positive ($C_u = 1$) or a negative result ($C_u = 0$) of the enforcement operation (the events on channel object occurred or not occurred in accord-

ance with reference sequence). The result marked as $[P_u^2]$ of control procedure P_u^2 indicates events compatibility on the channels with their expected sequence ($C_u = 1$) or the lack of such compatibility ($C_u = 0$).

Interpretation of the control function of C_u for enforces procedures is as follows:

- 0 – if enforced sequence of events is different than assumed in the pattern specified in the function of control parameters C_u
- 1 – if enforced sequence of events has been implemented in accordance with the pattern
- 2 – the required time for carrying out the sequence of events determined in the pattern has not yet passed or the realization of events has not started yet.

Interpretation of the value of the control function of C_u for the control procedure is as follows:

- 0 – controlled sequence of events is not consistent with the pattern stored in the function of control parameters C_u
- 1 – controlled sequence of events has been implemented in accordance with the pattern;
- 2 – control of events has not yet been completed or not started at all

Figure 4 shows a timing diagram of permissible changes of C_u control functions in case of the enforcement procedure Q_u^1 , and Figure 5 presents the case of control procedure of P_u^2 . Changes as a result of the completion

of various phases of the enforcement and control operation are taking place, respectively (after suspension of t_z on a particular channel operation and after the time of enforcement t_w or after the control of t_k).

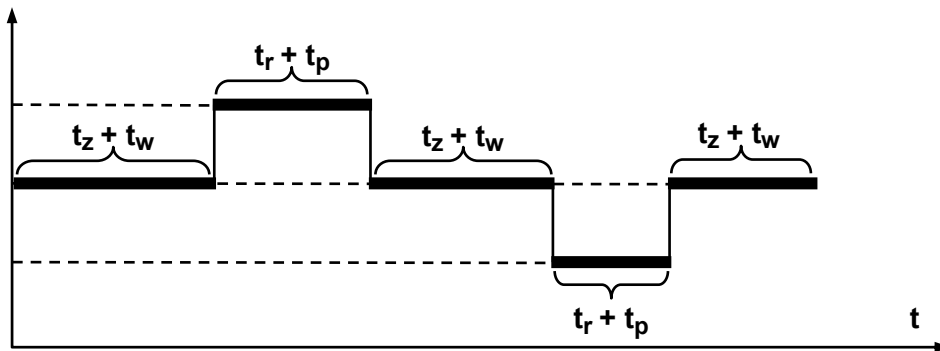


Figure 4. Timing diagram of changes in C_u control function for the enforcement procedure Q_u^1

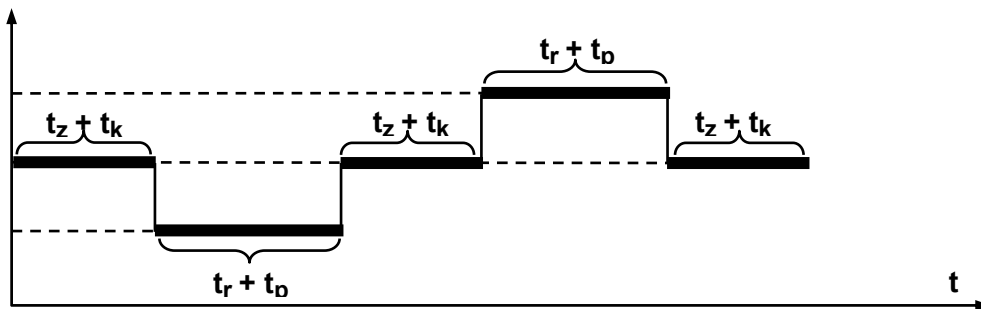


Figure 5. Timing diagram of changes in C_u control function for the control procedure P_u^2

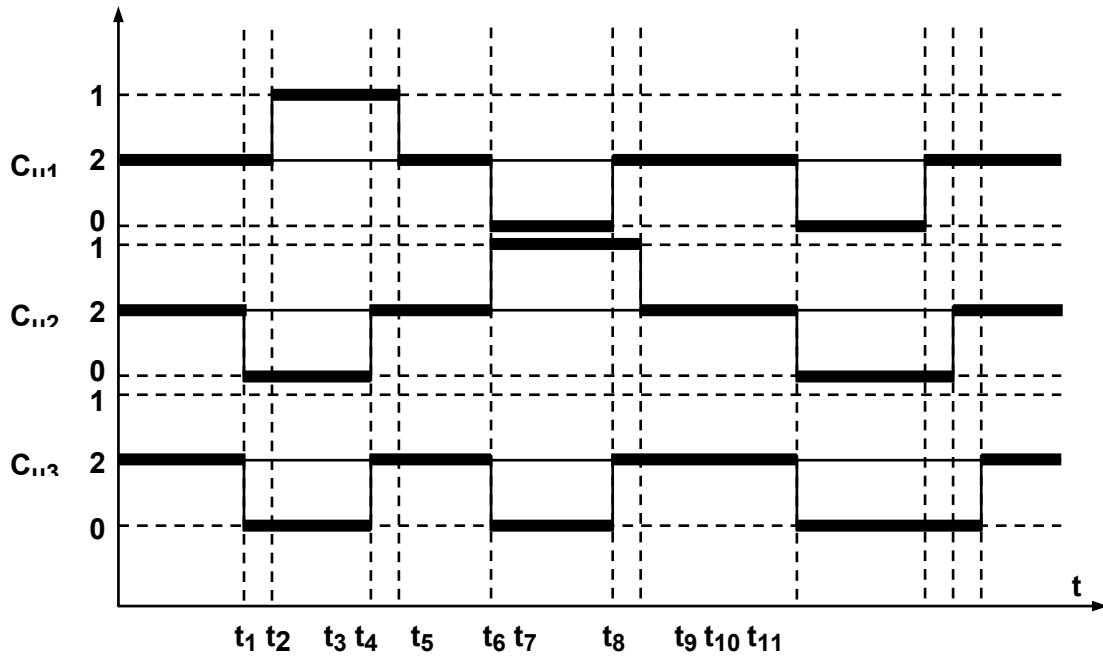


Figure 6. Timing diagrams of the value changes in the sequence $\langle C_{u1}, C_{u2}, C_{u3} \rangle$ of elementary control functions

Values 0, 1 of control function are returning to a value 2 when the response function corresponding to a specific control function returns to a value 2 (after reaction time t_r and propagation delay t_p of the value of reaction function). In case of the enforcement procedure, a value 0 may indicate the emergence of higher priority sequences of enforcement events on a specific channel than the priority sequence of events currently being implemented, which results in interruption of the enforcement procedure. The appearance of the 0 function control, accompanying the enforcement procedure, can be also determined by the inability to start the operation of events sequence (e.g. due to the emergency disconnect from the channel control system).

Each control procedure $P_u^?$ has a sequence of at least two alternative patterns, which control the analysis of events on a channel. Each enforcement procedure $Q_u^!$ has only one pattern, which determines the implementation of the events on a channel.

In the case of complex logical expressions (trivalent logic), a value of the control function C_u may be modified by logical operator E which depends on its own state of $[E] \in \{0, 1, 2\}$ and executes one of the three transformations $[E]C_u$:

- inversion ($[E] = 0$),
- identity ($[E] = 1$),
- or background ($[E] = 2$).

The following join dependencies should be fulfilled:

$$\begin{aligned} [0] 0 = 1 \quad [1] 0 = 0 \quad [2] 0 = 2 \\ [0] 1 = 0 \quad [1] 1 = 1 \quad [2] 1 = 2 \\ [0] 2 = 2 \quad [1] 2 = 2 \quad [2] 2 = 2 \end{aligned} \quad (7)$$

Control function C_u for control procedure $P_u^?$ is denoted as elementary sequences of control function (C_{ui}, \dots).

$$(C_{u1}, C_{u2}, \dots, C_{un}) \quad (8)$$

Control function C_u for control procedure $Q_u^!$ is denoted as a single element of the sequence:

$$(C_{u1}) \quad (9)$$

The following restrictions are applied on the control function of C_u , defined as a sequence:

- any element C_{ui} can adopt only the values from the set $\{0, 1, 2\}$; it also refers to the sequence of a single-element function (C_{u1}); the change of the items value C_{ui} may occur only in accordance with the diagram,
- if at a particular moment one of the elements $C_{ui} = 1$, then the remaining elements of the sequence can achieve only the value 0,
- if at a particular moment one of the elements $C_{ui} = 0$, thus all other elements require a value 0, or one and only one element achieves a value 1, and the other elements a value 0,
- element C_{ui} returns to a value 2 when the corresponding reaction function R_x assumes a value 2.

The response function R_x is denoted by expression:

$$R_x(I_i, [E]C_u) \quad (10)$$

and assumes values compatible with the following indication:

$$\begin{aligned} 0 & - \text{if } I_i = 1 \text{ and } [E]C_u = 0 \text{ or if } I_i = 0 \\ 1 & - \text{if } I_i = 1 \text{ and } [E]C_u = 1 \\ 2 & - \text{if } I_i = 1 \text{ and } [E]C_u = 2 \text{ or if } I_i = 2 \end{aligned} \quad (11)$$

A value 2 for the function from the sets I, C, and R is treated as a background for changes in the value 0, 1 (2). It is considered arbitrarily that any of these functions may change its value only in accordance with the diagram limit changes $0 \leftrightarrow 2 \leftrightarrow 1$.

Logical operator in the considerations may be omitted (e.g. by setting $E = 1$ for all instances of E).

Each enforcement procedure Q_u^1 has only one pattern to follow and in accordance with it implements the events on the channel to which it is assigned.

The control procedure P_u^2 has a sequence of at least two alternative patterns, according to which the events control on a channel is analyzed.

The reaction function $R_x \in R$ is interpreted as a result of the work of control procedure Q_u^1 or enforcement procedure P_u^2 . Reaction function R_x is a model, which describes the results of initiation procedures. To determine the value of a single-response function, the following are necessary:

- initiation function value,
- control function value, and
- (in some applications) the value of logical operator associated with the control function.

Because the control function C_u is defined as a sequence of elementary control functions (C_{ui}, \dots), which correspond to the checked patterns of events, thus, the corresponding sequences of the reaction function can be assigned to this sequence

$$(R_{x1}, R_{x2}, \dots, R_{xn}) \quad (12)$$

Elementary reaction functions R_{x1} of the above sequence are used to build an analytical initiation function $I_i = \nabla(R, \dots)$.

The sequence of reaction function changes its value according to the diagram $0 \leftrightarrow 2 \leftrightarrow 1$.

Any pair of elementary reaction function R_{xi}, R_{xj} differs from others in their initiation functions I_i, I_j in the sequence of reaction function.

A value of (R_x, \dots) sequence is computed in the following way:

- 0 – if all $R_{xj} \in (R_x, \dots)$ have a value other than 2, and at least one of them is equal to 0
 - 1 – if all $R_{xj} \in (R_x, \dots)$ have a value equal to 1
 - 2 – if even one $R_{xj} \in (R_x, \dots)$ is equal to 2
- $$(13)$$

In computing the sequence of reaction function returns to the background value (and thus also return to its background value all the elementary functions of the reaction) at a time when the associated initialization function takes the value 1 or 0.

8 Analytic form of initiation function

The evaluation of initiation function value is based on the analytical form (formula) of this function

$$I_i = \nabla(R_x, \dots)_s \quad (14)$$

where ∇ is the symbol of alternative trivalent disjunction, which semantics are described by previously presented expressions, R_x is the elementary reaction function, $(R_x, \dots)_s$ represents the sequence of reaction functions.

A value of initiation function is determined as follows:

- 0 – if all sequences $(R_x, \dots)_s$ have a value 0
 - 1 – if one and only one sequence $(R_x, \dots)_s$ has a value 1, and other sequences have a value 0
 - 2 – if even one sequence $(R_x, \dots)_s$ has a value 2
- $$(15)$$

The hypothesis that there is at most only one such sequence $(R_x, \dots)_s$ which can assume a value 1, makes an important limitation adopted in determining the value of initialization function $I_i = 1$. This assumption is fundamental to the whole nature of the computational process, which takes place in determining the function value of sets I, C and R.

The assumption of the existence of at most only one sequence $(R_x, \dots)_s$, where 1 results from to the correct construction of valid computational processes in the initialization function networks. A value as an element of set $\{0, 1, 2\}$ can be assigned only to sequence $(R_x, \dots)_s$.

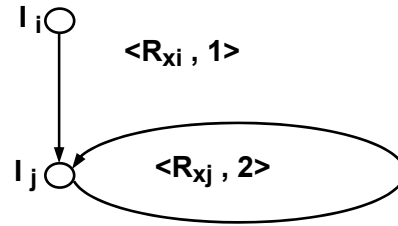


Figure 7. Cycle computations of initiation function

The value computations of the individual control functions, reaction function and initiation function must take place simultaneously (in parallel) for all functions. Hence, it is necessary to synchronize the processes of computation, which is particularly evident for the cycle computations where the initiation function I_i occurs as an immediate or indirect operand for itself. Fig. 7 shows such a type of situation in the network of initialization function.

In accordance with the analytic form of initiation function

$$I_j = (R_{xi}) \nabla (R_{xj}) \quad (16)$$

where

$$R_{xi} = R_{xi}(I_i, [E]C_{ui}) \text{ and } R_{xj} = R_{xj}(I_j, [E]C_{uj}).$$

In the considered example (Fig. 7), function I_i is acting for itself as the immediate operand.

9 Network initialization functions

The network initialization function is a model setting formulas and computing order of initiation function. The network initialization function N is defined as a report:

$$\langle I, R, V, U, H_v, H_u, B^+, B^-, r, s \rangle \quad (17)$$

where:

- I – set of initiation function,
- R – set of reaction function,
- V – set of nodes of the network graph,
- U – set of edges of the network graph,
- H_v – is the function applying mapping of $V \leftrightarrow I$,
- H_u – is the function applying mapping of $U \leftrightarrow \langle R_x, r, s \rangle$,
- B^+ – is the function applying inductive coupling 0,
- B^- – is the function applying coupling quench 0,
- r – index of edge output (relative to the graph node),
- s – index of edge input (relative to the graph node).

All sets are finite. Graph comprising a set V of nodes together with a set U is a carrier network N . Graph network N contains two highlighted nodes V_a and V_b that indicate a path for any $V_j \neq V_a$ from V_a to V_j and for any $V_i \neq V_b$, there is a path from V_i to V_b ; $V_a, V_b, V_i, V_j \in V$. A node V_a is called an initial node and V_b node is called a final node of graph network. There is only one initial and only one final graph node in the network.

Function H_v weights graph node of the network N using one function from the set I , and the function H_u weights edge graph of network N by one function R_x from a set R . The function H_v applies monotone mapping of $V \leftrightarrow I$, and the function H_u orders monotone mapping of $U \leftrightarrow \langle R_x, r, s \rangle$.

Function H_v weights each network node graph N using one initiation function in such a way that the different nodes correspond to different functions of initiation. Function H_u weights every edge graph of network N using a set $\langle R_x, r, s \rangle$ in such a way that the different input edges correspond to different sets. An edge directed from node V_i to node V_j (weighted by initiation functions I_i and I_j) is weighted by compilation $\langle R_x, r, s \rangle$, where the reaction function R_x is determined analytically as $R_x(I_i, [E]C_{ui})$.

Index r is used for numbering and identification of outgoing edges of the nodes. A node numbering is carried out from 'left to right' counter-clockwise; if the index r is repeated in several combinations $\langle R_x, r, s \rangle$ of outgoing edges of the same node, it means that they have a common elementary reaction feature (to the accuracy of the copy). Particular reaction function R_x in formula is treated as an element of one and only one sequence $(R_x, \dots)_s$ of index s , which is equal to the index of the compilation $\langle R_x, r, s \rangle$. In case when it is necessary to repeat function R_x in different sequences (combinations $\langle R_x, r, s \rangle$), due to the synchronization of computation its copy are made: R_x', R_x'', \dots

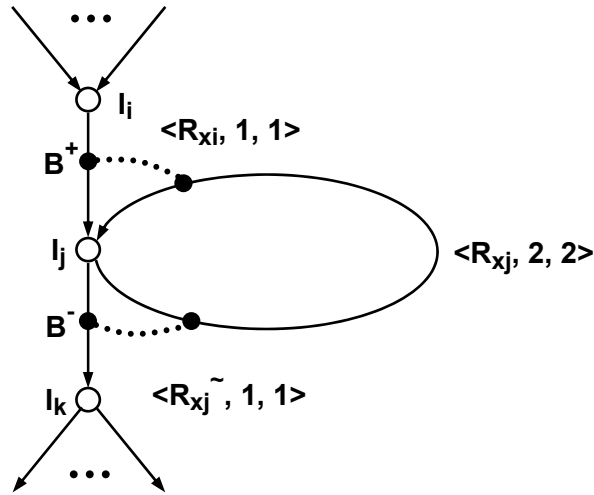


Figure 8. Example of a network fragment with sequences and couplings

B^+ is a function assigning inductive coupling of 0 and its main task is to induce a value of zero in one of the reaction functions covered by inductive coupling if the second function of the coupling reaction is equal to 1.

B^- is a function assigning the dampening coupling 0 and its task is to suppress a value 0 (by changing to a value 2) for one of the reaction functions covered by dampening coupling if the second function of coupling reaction has a value 1.

An example of application functions B^+ , B^- is illustrated in Fig. 8. The corresponding calculation formulas have the form as follows:

$$\begin{aligned}
 I_j &= (R_i / R_j^+)_1 \nabla (R_j / R_i^+ / R_j^-)_2 \\
 I_k &= (R_j^- / R_j^-)_1 \\
 R_i &= R_i(I_i, [E] C_u) \\
 R_j &= R_j(I_j, [E] C_u)
 \end{aligned}
 \tag{18}$$

where:

R_i / R_j^+ denotes inductive coupling 0 for function R_i , if $R_j = 1 \Rightarrow (R_i = 0)$,

$R_j / R_i^+ / R_j^-$ denotes inductive coupling 0 for function R_i , if $R_j = 1$ and dampening coupling 0 for function R_j , if $R_j^- = 1$
 $(R_i = 1 \Rightarrow R_j = 0, R_j^- = 1 \Rightarrow R_j = 2)$

Edges weighed by combinations $\langle R_x, r, s \rangle$ and directed to the node V_i , which weighed by initialization function I_i , assign a subset of the reaction function R' onto which a function is determined. In turn, a subset of R' is distrib-

uted into disjoint subsets (sequences) $(R_x, \dots)_s$ each of which corresponds to one index s and different indexes r . Indexes s assigned to the sets $\langle R_x, r, s \rangle$ and relating to the sequence $(R_x, \dots)_s$ of one initiation function I_i (one vertex) form a set S_i . The number of different sequences is equal to the numerousness of a set S_i ; indexes s assume values 1, 2, ..., S_i (from left to right, clockwise rotation). Thus, the sequence $(R_x, \dots)_s$ constitutes a set of these $R_x \in R'$ assigned by function H_u to graph edges with a common end vertex and identical indexes 1 in the sets of $\langle R_x, r, s \rangle$.

A part of the network corresponding to the sequences $(R_x, \dots)_s$ is shown in Fig. 9. The initiation function I_p is determined by the following formula:

$$I_p = (R_1, R_2)_1 \nabla (R_1', R_3)_2 \tag{19}$$

The network of function N constructed in accordance with these principles allows to perform calculations and analysis of their correctness if their implementation will go in accordance with the appropriate axiom of Clarity, Feasibility and Compatibility (axiom system CFC). The calculation of the initiation function, the reaction function and control functions is carried out simultaneously across the network initialization function. The correctness of the calculations (according to the characterization principle: a functioning model Ψ_a) is determined by the correctness network topology (structural model Ψ_b) in the sense of axiom system CFC.

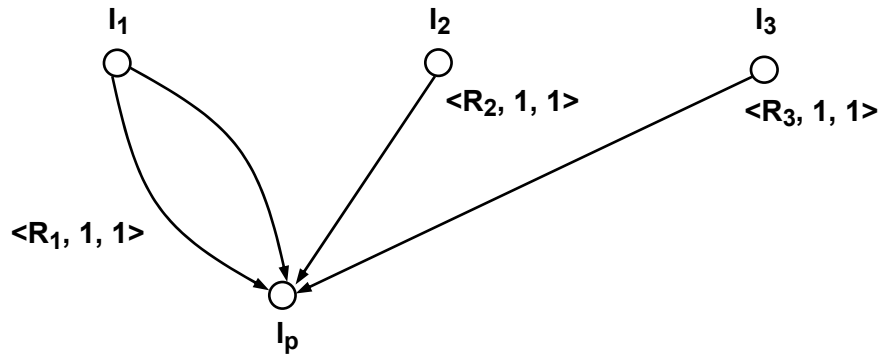


Figure 9. Example of the network fragment with the multiple edges and two sequences $\langle R_1, 1, 1 \rangle$ and $\langle R_1', 1, 2 \rangle$

10 Summary – the axioms of Clarity, Feasibility and Compatibility

Organizational–technical systems design can be carried out in many ways. This work presents considerations indicating the possibility of designing complex systems combining with simultaneous analysis of their functioning, carried out at the level of logical structure.

The principle of characterization belongs to the methodological apparatus of modern systems theory [4]. The main epistemological postulates of the characterization principle are as follows:

- characteristics of the solutions rather than the solutions themselves should be sought;
- solution characteristics should relate to the created class representatives (invariants) of equivalent solutions; and
- equivalent solutions class is created as a result of input data interpretation of the solved task group of the problem area in the representative solution characteristics categories.

Usually, there is less equivalent class solutions than the solutions themselves and the analysis of the characteristics of solutions can be carried out without their direct (objective) generation. Characterization theory consists of formal elaboration and methodological verification in the selected characterization theory objective area, which includes the basic idea of the mutual interpretability of functioning model to the model structure of certain object with the model of its structure. Mutual interpretability of models is achieved through:

- selection of universal rules of “proper” functioning (expressed in the functioning model) and
- structural (technical) interpretation of the functioning model.

The defining of axioms (and not just the rules) of the correct functioning (action) is the sine qua non of a satisfactory project design and the implementation of a system based on the project. The axioms allow constructing theories and theorems, and proving their completeness and consistency. Formulating rules may resemble, at the most, “patching” of the operating system or another system. It is impossible to effectively “patch up” loophole in the law or other legal act by using only rules, which are almost daily media reports.

The beginning of the road towards the correct project and its implementation is the correctness of the formal model. It is not possible to get the correct project if its model assumptions (functional and/or structural) are erroneous. The correctness of model, of course, does not imply automatically the correctness of project, but if the project has the characteristics of a formal model (and nothing else) and the producing operations are technologically correct, thus, in addition to a certain idealization, it is difficult to deny that approach. Hasse diagrams are examples of formally correct structures that can be automatically transferred and implemented in a physical medium. It is also important that the testing of the correctness of the Hasse logical structures is not necessary, as long as they have been obtained by using the theory of characterization.

The computation correctness on the \mathbf{N} network depends on the structure of the network. Calculations on the network relate to the value of initiation, response and control functions.

The calculation of these functions will be called logically valid (valid in a logical sense) if they take place in accordance with the logical correctness of the axiomatic system specific for the network of \mathbf{N} functions.

Axiomatic logical correctness is formed by:

- axiom of Clarity (C),
- axiom of Feasibility (F), and
- axiom of Compatibility (C).

Presented below axiom of extensionality, feasibility and compatibility (axiomatic CFC) is the result of the analytical work on multiple models of concurrent discrete processes, such as regular expression language [1, 11], machine scheme and operators [1, 3], a Petri net, transforming networks [10, 13, 14] and others.

10.1 Axiom of Clarity

Axiom of Clarity is formulated as follows:

If there is in the formula $\nabla (R, \dots)$ more than one sequence $(R_x, \dots)_s$, the value 2 of any of these sequences can be converted into a value 1 only in the calculation for one sequence and other sequences should assume a value 0; time sequence of the changes in the sequence is not important. (20)

Axiom of Clarity establishes the need for natural uniqueness (within the network of structure N , for example, damping and inductive coupling or artificial synchronization (using additional elements outside of the network) of the changes in a value of 0 and 1 onto 2. Thus, the possibility of the so-called "race" performance of computing and related race of initiation operation and control-enforcement operations are blocked.

10.2 Axiom of Feasibility

Feasibility axiom is formulated as follows:

Formulas $\nabla (R_x, \dots)$ must be constructed in such a way that during the calculation, starting from the initiation function $I_a = 1$ can be obtained for any one value $I_i \in I$ including the final operation initiation function I_b . (21)

This axiom requires that configurations and functions of the initiation sequence $(R_x, \dots)_s$ should not occur in the structure of the network initialization function N because they could affect the suspension of computing operations, for example, due to the lack of changes in the value of individual response functions in accordance with the diagram $2 \leftrightarrow 0$ or $2 \leftrightarrow 1$, and at the

same time the initiation functions for which the reaction functions are arguments. Feasibility axiom shows the need to ensure the achievement of value 1 by any function of the reaction occurring on the network N .

This axiom implies a potential need for the adoption of a value 1 by each initiation function, however, in a particular course of calculation from $I_a = 1$ to $I_b = 1$, not all the functions of initiation must be set to value 1.

10.3 Axiom of Compatibility

Axiom of Compatibility reads as follows:

None of the functions $I_i \in I$, $R_x \in R$, $C_u \in C$ cannot change its value in other way than assigned by diagram $0 \leftrightarrow 2 \leftrightarrow 1$ (22)

Axiom of Compatibility blocks the formation of queue arguments and computing operations of initiation and response functions. The fulfillment of this axiom allows, consequently, observing the proper functioning of the network N of initiation function.

Axiom of Compatibility precludes the existence of value queues different from the background (which could arise, for example, as a result of the occurrence of cyclic zones in the network) and, to some extent, "organizes" (synchronizes) the arguments and computational operations of function I_i .

The synchronization effect of value changes is determined by appropriately defined computational rules that under certain conditions induce functions I , R and C , which take a value 2. Synchronization allows the alignment of "relative speed" changes in the reaction function R_x which are arguments of one initiation function.

It is also assumed that the spread of a value 0 across network elements is at the "speed of light", i.e. immediately, regardless of the type of function (including the initiation and reaction function).

For practical reasons, the design of network N (the process) should weaken the functional limitations (expressed in a model Ψ_a) and structural constraints (expressed in a model Ψ_b). This issue will become more apparent after the presentation of some special properties of the N network (relationships on the network function). An axiom system CFC allows diagnosis of the so-called local and global correctness of the computation operations as well as the topologically error-free design of network function N .

The problem of the weakening of functional limitations and structural constraints requires a separate treatment because the solution is associated with the argumentation of many theorems about the completeness and consistency of primary assumptions of the original axiom CFC.

11 Bibliography

- [1] Bromirski J. - *Teoria automatów*. WNT, Warszawa 1970.
- [2] Dietrych J. - *System i konstrukcja*. WNT, Warszawa 1985.
- [3] Gorbatov V.A. - *Schemy uprawlenija CVM i grafy*. Izd. Energija, Moskva 1971.
- [4] Gorbatov V.A. - *Semanticeskaja terija projektirovanija avtomatov*. Izd. Energija, Moskva 1979.
- [5] Krupa T. - *Zasada charakteryzacji w projektowaniu systemów organizacyjno- technicznych*. Zeszyty Naukowe IOZ PW, Nr 1, Warszawa 1983, pp. 33-52.
- [6] Krupa T., Ostrowska T. - *The Logical Aspects of Tasks and Resource Design*. Int. Conf. "Cybernetics and Systems'90". World Scientific, University of Vienna, Vienna 1990, pp. 267-274.
- [7] Krupa T. - *Koncepcja przemiany organizacyjnej*. [in] Krupa T. i inni - *Aktualne problemy naukowo-dydaktyczne w organizacji produkcji, zarządzaniu i technologii*. Wyd. IOSP PW, Warszawa 1997, pp. 27-51.
- [8] Krupa T. (ed.) - *New Challenges and Old Problems in Enterprise Management*. WNT, Warszawa 2002.
- [9] Krupa T. - *O pewnej specyfikacji procesu decyzyjnego* [w] *Problemy Zarządzania*. Wydawnictwo Naukowe WZ UW, Nr 1 2005, pp. 79-88.
- [10] Krupa T. - *Sieciowe modele procesów zdarzeń* [in] *Komputerowo zintegrowane zarządzanie* (red. R. Knosala). Oficyna Wydawnicza PTZP, Opole 2006, pp. 55-66.
- [11] Krupa T. - *Elementy organizacji, zasoby i zadania*. WNT, Warszawa 2006.
- [12] Krupa T., Ostrowska T. - *Multilayer Decision Support Model for Value and Cost Analysis of IT Solutions – Hierarchical Approach* [in] *Managing Worldwide Operations and Communications with Information Technology*. IGI Publishing, IRMA Int. Conf., Vancouver Canada 2007, pp. 86-90.
- [13] Krupa T. - *Events and Events Processes* [in] *Foundations of Management - International Journal*, Vol. 1, No. 2, 2009, pp. 143-158.
- [14] Krupa T. - *Transforming Nets* [in] *Foundations of Management - International Journal*, Vol. 2, No. 1, 2010, pp. 21-40.
- [15] Krupa T., Ostrowska T. - *Decision Making in the Flat and Hierarchical Decision Problems* [in] *Foundations of Management - International Journal*, Vol. 4, No. 2, 2012, pp. 23-36.
- [16] Kulińska E. - *Aksjologiczny wymiar zarządzania ryzykiem procesów logistycznych. Modele i eksperymenty ekonomiczne*, Oficyna Wyd. Politechniki Opolskiej, Opole 2011.
- [17] Ostrowska T. - *The Resource Hazards Model for the Critical Infrastructure of the State Emergency Management Process* [in] *Foundations of Management - International Journal*, Vol. 5, No. 3, 2013, pp. 49-60.
- [18] Prokopczuk R., Krupa T. - *Modelowanie procesów biznesowych z IBM WebSphere Business Modeler* [in] *Uwarunkowania i analizy w systemach organizacyjnych przedsiębiorstw* (red. T. Krupa). Oficyna Wydawnicza PTZP, Opole 2008, s. 185-198.
- [19] Prokopowicz T., Krupa T. - *Modelowanie procesu zarządzania przedsiębiorstwem w sytuacji zagrożenia upadłością w oparciu o zasadę charakteryzacji* [in] *Uwarunkowania i analizy w systemach organizacyjnych przedsiębiorstw* (red. T. Krupa). Oficyna Wydawnicza PTZP, Opole 2008, pp. 199-216.