ISSN 0147-6882, Scientific and Technical Information Processing, 2021, Vol. 48, No. 5, pp. 327–332. © Allerton Press, Inc., 2021.
Russian Text © The Author(s), 2020, published in Iskusstvennyi Intellekt i Prinyatie Reshenii, 2020, No. 2, pp. 86–93.

A System–Object Approach to Determinant Analysis of Complex Systems

S. I. Matorin[a], * and V. V. Mikhelev[b]

[a] *SoftConnect CJSC, Belgorod, 308023 Russia*

[b] *Belgorod State University National Research University, Belgorod, 308015 Russia*

**e-mail: matorin@softconnect.ru*

Received April 3, 2020

Abstract—This article describes the possibility of analyzing systems using the system–object approach. The concept of determinant analysis of systems based on this approach is introduced, which allows one to consider the detailed description of a system as the gradual refinement of its properties. It is proposed to use the calculus of Abadi–Cardeli objects and the *ALCHIO (D)* descriptive logic language to formalize the stages of determinant analysis, from class systems (external or conceptual) to phenomena systems (internal or material).

Keywords: system–object approach, determinant analysis, phenomena systems, class systems, formalization, calculus of objects, descriptive logic

DOI: 10.3103/S014768822105004X

INTRODUCTION

The systems analysis that emerged in the 1960s together with the invention of computers provides a logical and consistent approach to studying complex systems and their properties, which was shown by both the first attempts to apply it in military management tasks (choosing engineering and economic specifications of bomber planes and air bases) and its subsequent use in solving diverse problems.

Despite the obvious success of analytical activities under the umbrella of systems analysis, there is no clear understanding of systems analysis proper [1, p. 231]. Several definitions of the systems analysis have been suggested, for example, in [1–3]. There are also several systems analysis procedures that are designed as sums of various principles, approaches, and methods that provide a certain level of analytical efficiency but do not include systems analytics proper [1]. The analysis of diverse systems analysis procedures (Quaid's, Optner's, Chernyak's, Golubkov's, Young's, Tarasenko's, Kapitonov's, Plotnitskii's, and others) makes it clear that they are essentially different from each other, although do have some elements in common and are designed as fairly general guidelines whose methods are not specifically indicated. Most importantly, however, all of these procedures make essentially no use of the concept of a system, ignore the systemic impact, use principles of the systems approach quite superficially, and do not rely on general system regularities.

It must also be considered that there is currently no clear definition of the systems approach itself. In our opinion, there are three kinds of this approach, including system–structural (functional and process), object-oriented, and system–object [4]. Since the tools and means of the system–object approach allow one to consider system properties and relations and general system regularities more efficiently [4–8], it is proposed to use the system–object analysis procedure including systems analytics.

This article describes an efficient systems analysis tool defined as determinant systems analysis based on the system–object approach. Initially introduced by Mel'nikov [9], the concept of determinant analysis allows determination of the buildup stages of a system depending on its existing properties. The formalization of these stages using the calculus of Abadi–Cardeli objects and a descriptive logic language is proposed.

1. THE BASICS OF SYSTEM–OBJECT ANALYSIS

One integral stage of the conventional systems analysis of a loosely formalized object is the analysis of the reasons that the object (system) has certain properties. In the existing systems analysis procedures this stage corresponds to the stage of goal definition, identification, or formation. Systems analysis based on the system–object approach is exactly the type that makes it possible to identify the sources and reasons for the existence of the properties the system has. This is provided by an array of special concepts from the system–object approach, some of which are borrowed from Mel'nikov's work [9].

First of all, this approach considers a system as a phenomenon (material object) or a class (conceptual system) whose function or role are due to the function of a phenomenon or the role of a class from a higher level (that is, by a phenomenon suprasystem or a class suprasystem). Thus, the system–object approach considers the existence of not only phenomena systems (internal, according to Schrader [10], or material, according to Ackoff [11]) but of class systems as well (external, according to Schrader, or conceptual, according to Ackoff).

Secondly, the functions of a system are due to the functions of a suprasystem. This phenomenon is considered to be a functional request from the suprasystem to a system with a certain function and considered to be the system's external determinant.

Thirdly, the performance of the system affected by an external determinant that directly shapes the internal properties of this system (properties of subsystems) is considered to be the system's internal determinant. There is the difference between the system's current internal determinant reflecting the system's performance at a current instant and the critical internal determinant, which must essentially be as close as possible to the external determinant.

In our opinion the external determinant of the system is the cause of its origin, the goal of its existence, and the main determiner of its structural, functional, and substantial properties. In the system–object approach this determinant is thus considered a universal backbone factor. The performance of the system according to the external determinant (that is, the compliance of the internal determinant to the external determinant) engages the system and the suprasystem in the relation of maintaining the functional capability of a greater whole.

The representation of the system is convenient to specify as a triune Node–Function–Object structure describing the system's structural, functional, and substantial characteristics [4]. In this case the node as the intersection of connections describes the functional request of the suprasystem to the system, that is, its external determinant.

According to [6–8], the phenomenon system **s** is formally represented as a special object of the calculus of Abadi–Cardeli objects, that is, as

$$\mathbf{s} = [(\mathbf{Ls}?,\ \mathbf{Ls}!); \mathbf{fs}(\mathbf{Ls}?)\mathbf{Ls}!; (\mathbf{Os}?, \mathbf{Os}!, \mathbf{Osf})], \quad (1)$$

where **(Ls?, Ls!)** is the field for describing node **us** or intersection of a finite set of input connections **Ls?** and output connections **Ls!** in the structure of the suprasystem; **fs(Ls?)Ls!** is the field for describing function **fs**, preset by the system, or the method that provides the functional compliance between output flows **Ls?** and input flows **Ls!** of the node in question; and **(Os?, Os!, Osf)** is the field for describing the substantial (object) (input, output, and transmissive) characteristics of the system.

The class system $\boldsymbol{S^i}$, whose role is determined by the role of the class system from a higher hierarchical level, is formally represented as a different special object of the calculus of Abadi–Cardeli objects using designations from descriptive logic [8]:

$$\forall \boldsymbol{S}^i \exists \boldsymbol{RS}^i \text{ and } \boldsymbol{S}^i = [\boldsymbol{S}^{i-1}; \boldsymbol{RS}^i \sqsubset \boldsymbol{RS}^{i-1}], \quad (2)$$

where $\boldsymbol{S^{i-1}}$ is the field for indicating the class system from a higher hierarchical level corresponding to node $\boldsymbol{US^i}$ of system $\boldsymbol{S^i}$; $\boldsymbol{RS^i \sqsubset RS^{i-1}}$ is the field for describing the method corresponding to role $\boldsymbol{RS^i}$ (function $\boldsymbol{FS^i}$) of system $\boldsymbol{S^i}$ embedded in role $\boldsymbol{RS^{i-1}}$ of suprasystem $\boldsymbol{S^{i-1}}$; and $\sqsubset$ symbolizes the nesting of role to role and symbol to symbol in the language of descriptive logic.

An efficient system analysis tool can be proposed that relies on the above introduced notions of the system–object approach. This tool is a modification and formalization of the determinant analysis proposed in [9] and allows describing the causes of the system's origin, the stages of its buildup, and its existing properties; this information must be known when analyzing existing and designing new systems.

According to Mel'nikov's concept, the determinant analysis must start from analyzing the internal determinant of an object (system). This analysis allows one to disclose the sources of the object's internal properties, because the functional property of the whole, which is supported by everybody and in relation to which all of the other properties of the whole and its components are merely indirectly functional and support this main property from the inside, becomes the determining property of the whole. It is noted that the knowledge of the internal determinant alone is not enough for analyzing and understanding the object's properties in general. This understanding makes it necessary to know the reason that the object has this determinant, that is, it is necessary to know what the system's determining internal property is determined by from the outside. This will allow one to identify the (external) determinant of the system's internal determinant. The results of this analysis allow one to get as close as possible to defining the essential properties of the system and correctly evaluating its current state and development prospects. In the system–object approach the essential property of the system is, first of all, the functional property for which the system was formed. Thus, on the one hand, the system's essence stems from the functional request from a higher-level system (suprasystem), that is, the external determinant; on the other hand, this essence is the internal source of the system's essential functional properties [9].

The determinant analysis procedure involves classification analysis and is also considered a tool for identifying essential properties and essence of objects (systems) [12]. This is the reason that Mel'nikov recommends performing determinant analysis by successively making up several classifications, such as partitive (or measurenomic/metronomic), genetic (or multistage), and generic-specific (or taxonomic).

2. DETERMINANT SYSTEM–OBJECT ANALYSIS PROCEDURE

In our opinion, the main aim of any systems analysis, as well as the initial point for designing any new system, must be to define the backbone factor, that is, the reason the system exists or is designed for. Henceforth, the main aim of determinant system–object analysis (DSOA) is to identify the suprasystem of and the function request to the considered system, that is, the cause of its origin or creation (external determinant). As emphasized by Mel'nikov, the pioneer of determinant analysis, if the examination of the system has allowed one, first of all, to identify its external determinant, the internal determinant is derived from the external one by substantive reasonings about the system's formative stages.

This is the reason that we recommend to start DSOA by building a generic-specific classification of the systems of the analyzed subject domain; this classification must allow one to identify various classes of the considered systems and evaluate the typical invariant properties of the analyzed or designed system. In its respect, this evaluation favors the elaboration of ideas about the suprasystem's functional request that corresponds to the critical internal determinant of the system in question. This opportunity is due to the fact that essential properties of systems are defined, first of all, by a hierarchy of classes [4, 13].

When building the generic-specific classification, that is, the hierarchy of classes of the systems from the analyzed subject domain that describes a subject domain that corresponds to the analyzed or designed system, it is necessary to define the abstract class (class system) including this system, which is recorded as

$$\boldsymbol{S}^{i,n} = [\boldsymbol{S}^{i,n-1};\ \boldsymbol{RS}^{i,n} \sqsubset \boldsymbol{RS}^{i,n-1}], \qquad (3)$$

where the superscript $\boldsymbol{i}$ means that this abstract class includes the analyzed phenomenon system $\mathbf{s}^i$ and the index $\boldsymbol{n}$ is the number of a level in the hierarchy of class systems. The notations in expression (3) correspond to the designations in expression (2).

The next classifying step makes it necessary to switch from abstract class $\boldsymbol{S}^{i,n}$ to a concrete class of the lower level; this class describes the analyzed or designed system as a class system using node, function, and object classes as

$$\boldsymbol{S}^{i,n+1} = \Big[\big(\mathbf{L}\boldsymbol{S}?^{i,n+1}, \mathbf{L}\boldsymbol{S}!^{i,n+1}\big); \\ \boldsymbol{FS}^{i,n+1}\big(\mathbf{L}\boldsymbol{S}?^{i,n+1}\big)\mathbf{L}\boldsymbol{S}\,!^{i,n+1}; \\ \big(\boldsymbol{OS}?^{i,n+1}, \boldsymbol{OS}!^{i,n+1}, \boldsymbol{OS}\mathbf{f}^{i,n+1}\big)\Big].$$

Thus, building the generic-specific classification of the analyzed subject domain allows one to conceptually (at the class level) define the external determinant of the class system or the reasons that the system has particular structural, functional, and object characteristics or the possible sources of their creation.

Further classifying results in building the genetic classification of the analyzed class system. This classification identifies the phases in which the required phenomenon system forms from the class system. The building of the genetic classification as a continuation of the generic-specific classification allows distinguishing or forming the node ($\mathbf{L}\boldsymbol{S}?^{i,n+2}$, $\mathbf{L}\boldsymbol{S}!^{i,n+2}$) from the node class ($\boldsymbol{LS}?^{i,n+1}$, $\boldsymbol{LS}!^{i,n+1}$). That node is the intersection of concrete input and output flows (like the functional request from the phenomenon suprasystem represents the external determinant of the analyzed or required phenomenon system), which make more specific class system $\boldsymbol{S}^{i,n+1}$:

$$\boldsymbol{S}^{i,n+2} = \Big[\big(\mathbf{L}\boldsymbol{S}?^{i,n+2},\ \mathbf{L}\boldsymbol{S}!^{i,n+2}\big); \\ \boldsymbol{FS}^{i,n+2}\big(\mathbf{L}\boldsymbol{S}?^{i,n+2}\big)\mathbf{L}\boldsymbol{S}!^{i,n+2}; \\ \big(\boldsymbol{OS}?^{i,n+2}, \boldsymbol{OS}!^{i,n+2}, \boldsymbol{OS}\mathbf{f}^{i,n+2}\big)\Big].$$

In the next step of classification the concrete function $\mathbf{f}\boldsymbol{S}^{i,n+3}(\mathbf{L}\boldsymbol{S}?^{i,n+3})\mathbf{L}\boldsymbol{S}!$ is additionally singled out or formed from function class $\boldsymbol{FS}^{i,n+2}(\mathbf{L}\boldsymbol{S}?^{i,n+2})\mathbf{L}\boldsymbol{S}!^{i,n+2}$ or concrete function $\mathbf{f}\boldsymbol{S}^{i,n+3}(\mathbf{L}\boldsymbol{S}?^{i,n+3})\mathbf{L}\boldsymbol{S}!^{i,n+3}$ forms; henceforth, class system $\boldsymbol{S}^{i,n+2}$ becomes even more concrete:

$$\boldsymbol{S}^{i,n+3} = \Big[\big(\mathbf{L}\boldsymbol{S}\,?^{i,n+3},\ \mathbf{L}\boldsymbol{S}\,!^{i,n+3}\big); \\ \mathbf{f}\boldsymbol{S}^{i,n+3}\big(\mathbf{L}\boldsymbol{S}\,?^{i,n+3}\big)\mathbf{L}\boldsymbol{S}\,!^{i,n+3}; \\ \big(\boldsymbol{OS}?^{i,n+3}, \boldsymbol{OS}\,!^{i,n+3}, \boldsymbol{OS}\mathbf{f}^{i,n+3}\big)\Big].$$

Finally, the singling out or formation of concrete substantial properties ($\mathbf{Os?}^i$, $\mathbf{Os!}^i$, $\mathbf{Osf}^i$) from object characteristics class ultimately distinguishes or forms phenomenon system $\mathbf{s}^i$ from class system $\boldsymbol{S}^{i,n+3}$ (that is, from $\boldsymbol{S}^{i,n}$):

$$\mathbf{s}^i = \Big[\big(\mathbf{Ls}\,?^i, \mathbf{Ls}\,!^i\big);\ \mathbf{fs}^i\big(\mathbf{Ls}\,?^i\big)\mathbf{Ls}\,!^i; \big(\mathbf{Os}\,?^i, \mathbf{Os}\,!^i, \mathbf{Osf}^i\big)\Big].$$

Further classification results in building the partitive classification of the analyzed phenomenon system. This classification is intended to identify the system's internal (sustaining) properties that formed as a result of adaptation to the request. This identification is made by analyzing the system's parts, components, and elements, that is, its subsystems. These properties are the system's partial functions that sustain its current cohesive functional properties. Therefore, their analysis will allow one to evaluate the current internal determinant of the considered system. For the cohesive representation of the DSOA stage, see Table 1.

The determinant system–object analysis whose stages are provided in Table 1 above is the procedure of defining the system's essence preset by the system's

Table 1. The stages of determinant system–object analysis

NFO approach	Valuable interpretation	DSOA stages
Node	Wish or Reason for requirement (Task)	Generic-specific classification: $S^{i,n} = [S^{i,n-1}; RS^{i,n} \sqsubset RS^{i,n-1}]$ $S^{i,n+1} = [(LS?^{i,n+1}, LS!^{i,n+1}); S^{i,n+1}(LS?^{i,n+1})LS!^{i,n+1}; (OS?^{i,n+1}, OS!^{i,n+1}, OSf^{i,n+1})]$
Function	Possibility Condition Designing	Genetic classification: $S^{i,n+1} = [(LS?^{i,n+1}, LS!^{i,n+1}); S^{i,n+1}(LS?^{i,n+1})LS!^{i,n+1}; (OS?^{i,n+1}, OS!^{i,n+1}, OSf^{i,n+1})]$ $S^{i,n+2} = [(LS?^{i,n+2}, LS!^{i,n+2}); FS^{i,n+2}(LS?^{i,n+2})LS!^{i,n+2}; (OS?^{i,n+2}, OS!^{i,n+2}, OSf^{i,n+2})]$ $S^{i,n+3} = [(LS?^{i,n+3}, LS!^{i,n+3}); fS^{i,n+3}(LS?^{i,n+3})LS!^{i,n+3}; (OS?^{i,n+3}, OS!^{i,n+3}, OSf^{i,n+3})]$ $s^i = [(Ls?^i, Ls!^i); fs^i (Ls?^i)Ls!^i; (Os?^i, Os!^i, Osf^i)]$
Object	Actuality Consequence Implementation	Partitive classification: $s^i = [(Ls?^i, Ls!^i); fs^i (Ls?^i)Ls!^i; (Os?^i, Os!^i, Osf^i)]$ $s_1^i = [(Ls?_1^i, Ls!_1^i); fs_1^i (Ls?_1^i)Ls!_1^i; (Os?_1^i, Os!_1^i, Osf_1^i)]$ $s_2^i = [(Ls?_2^i, Ls!_2^i); fs_2^i (Ls?_2^i)Ls!_2^i; (Os?_2^i, Os!_2^i, Osf_2^i)]$ … $s_j^i = [(Ls?_j^i\ Ls!_j^i); fs_j^i (Ls?_j^i)Ls!_j^i; (Os?_j^i, Os!_j^i, Osf_j^i)]$

suprasystem. DSOA allows mutual adjustment of the described classifications of an analyzed or designed system, from the class system level and up to the phenomenon system level, and thus allows one to come as close as possible to understanding the suprasystem's functional request to the considered system and the direction of its evolution as well as its current state. In its respect, this allows correctly formulating the system's essential properties and the extent to which the system corresponds to these. The knowledge about the object obtained in this manner allows more efficient evaluation of the current situation, predicting changes in the object's states, and elaboration of guidelines for further control and decisions. Thus, DSOA provides the researcher or designer with an efficient set of tools with diverse capabilities for analyzing or designing complex hard-to-formalize systems.

3. FORMALIZING DSOA BY DESCRIPTIVE LOGIC

The stages of DSOA are described above using the theory of the calculus of Abadi–Cardeli objects. DSOA can also be formalized by descriptive logic.

Descriptive logic (DL) is a language for representing knowledge about subject domains in a formalized manner using the primitive concept and role concepts. Concepts describe classes and roles describe interconcept relationships, which allows using them for describing concepts and their properties. One of the basic descriptive logics is DL *ALC* [14]. Its syntax is represented in short as

$$\{\top; \bot; A; A \sqsubseteq C; \neg C; C \sqcap D; C \sqcup D; \exists R.C; \forall R.C\},$$

where $\top$ and $\bot$ are the respective concepts of truth and falsehood; A is the primitive concept; C, D are the random concepts; and R is the atomary role.

The syntax of DL describes what expressions (concepts, roles, axioms, and others) it considers to be correctly composed. The semantics of DL shows how to interpret these expressions. Descriptive logic expresses general knowledge about concepts and their relations using general assertions, that is, the set *Tbox* of terminological axioms. On the other hand, the knowledge about individual objects, their properties and connections with other objects is the set *ABox* of assertions about the relationships and properties of individuals. Together, they form a knowledge base or ontology recorded as $K = TBox \cup Abox$.

There are various extensions for basic DL. As an example, *ALC* logic uses formal descriptions for representing the knowledge about a subject domain at the conceptual and abstract levels. In determinant system–object analysis, however, the genetic and partitive classification of the analyzed or designed system makes it necessary to describe phenomena systems with concrete properties that have numerical, time, or spatial characteristics, such as price, wages, and temperature. The logic *ALCHOIQ(D)* [15], which is an extension of *ALC*, has additional DL framework tools for formalizing all of the DSOA stages. These tools are

• *hierarchy of roles* (H), including supraroles R_1 and subroles R_2, in which case $R_2 \sqsubseteq R_1$. The set of these roles is called *Rbox;*

• *nominals* (O) defined as concepts formed from concrete individuals. The syntax of this concept is as follows: if a is an individual, then $\{a\}$ is a concept.

Nominals are full-scale concepts. We can write $A \sqcup \{a\}$ or $\exists R.\{a\}$;

• *reverse roles* (I). If R is a role, then R^{-} is a *reverse role*.

• *role restriction* (Q) set restrictions with syntax $\exists >$ $2R.C$.

• *concrete domain* (D). The pair (D, Φ), where D is a random nonempty set and Φ is the set of predicates in set D, allows one to preset set PN of predicate symbols. Each symbol $P \in PN$ has valence n; and Φ juxtaposes it with n-place relation $P^D \subseteq D^n$.

As an example, we assume that concrete domain $Order_N = (N, \Phi)$ is set of natural numbers N and predicate family Φ consists of binary predicates $<, \leq, =$ (and their negation: $\neq, \geq, >$) and monadic predicates $<_n, \leq_n, =_n$ (and their negation: $\neq_n, \geq_n, >_n$). We assume that *hasAge* is the concrete attribute (with values in D) indicating a man's age. Then, the set of adult people of at least 18 years of age presets the concept $Human \sqcap \exists hasAge \geq_{18}$.

The descriptive logic *ALCHOIQ(D)* has the following syntax, according to [15]:

$$\{T; \perp; A; A \sqsubseteq C; \neg C; C \sqcap D; C \sqcup D;$$
$$\exists R.C; \forall R.C; \geq nR.C; \{a\}; \exists[u_1, \ldots, u_n].P\}.]$$

Let us formalize the stages of DSOA by DL tools.

Generic-Specific Classification

In [13] *ALCHOIQ(D)* logic is used to describe the concept of a class system as the following concept:

$$S_{i,j}^{l} = S_{i-1,j}^{n} \sqcap \exists R.S_{i+1,pj}^{lj}, \quad (4)$$

where $S_{i,j}^{l}$, $S_{i-1,j}^{n}$, $R.S_{i+1,pj}^{lj}$ are the abstract classes (concepts), where $i = 0, \ldots, N$, i is the hierarchical level number, and l, j, lj, pj are the numbers within one level. The distinguishing of these classes is the first stage of determinant system–object analysis.

When switching from abstract classes to concrete, the system–object approach is used to represent the system as a class of nodes, class of functions, and class of objects (NFO element). For this purpose, the following concepts from expression (1) are introduced that correspond to NFO elements:

• *node* as the intersection of set of inputs $L?$ and set of outputs $L!$: $\boldsymbol{U} = L? \sqcup L!$;

• *function* that converts the set of inputs to the set of outputs as $\boldsymbol{F} = L! \sqcap \exists R.L?$. Expression $L! \sqcap \exists R.L?$ indicates set of outputs $L!$, connected through role R with set of inputs $L?$; R is the functional role showing the correspondence between the concepts. The role *R, hasCorrespondence,* can be introduced, which can refine the definition of function as $\boldsymbol{F} = L! \sqcap \exists\ hasCorrespondence\ L?$;

• *object* that implements the functions, has substantial characteristics, and is recorded as $\boldsymbol{O} = OS? \sqcup OS! \sqcup Osf$.

In terms of descriptive logic the system with concrete classes is represented as a sum of the above specified NFO elements:

$S = [U; F; O] = [(L? \sqcup L!);\ L! \sqcap \exists hasCorrespondenceL?; OS? \sqcup OS! \sqcup OSf]$.

Each concept can consist of subconcepts. As an example, R in a concept function can consist from subroles: $\boldsymbol{F} = L! \sqcap \exists(R_1 \sqcap \ldots \sqcap R_n).L?$. In its respect, this is also true for node $\boldsymbol{U} = ((L?_1 \sqcup \ldots \sqcup L?_n) \sqcup (L!_1 \sqcup \ldots \sqcup L!_n))$ and object $\boldsymbol{O} = (OS?_1 \sqcup \ldots \sqcup OS?_n) \sqcup (OS!_1 \sqcup \ldots \sqcup OS!_n) \sqcup (OSf_1 \sqcup \ldots \sqcup OSf_n)$.

Genetic Classification

This stage consists in the transition from class systems to phenomena systems, for which concepts from DL *ALCHOIQ(D)* are used, such as *nominal* and *concrete domain.* In this case, the inputs and outputs are concrete individuals, the function is preset by a concrete role, and the object is refined by concrete domains. The steps are given by

– $[(\{L?\} \sqcup \{L!\});\ \{L!\} \sqcap \exists R.\{L?\};\ OS? \sqcup OS! \sqcup OSf]$;

– $[(\{L?\} \sqcup \{L!\});\ \{L!\} \sqcap \exists hasCorrespondence.\{L?\};\ OS? \sqcup OS! \sqcup OSf]$;

– $[(\{L?\} \sqcup \{L!\});\ \{L!\} \sqcap \exists hasCorrespondence.\{L?\};\ \exists hasOS?.=_{n1} \sqcup \exists hasOS!.=_{n2} \sqcup \exists hasOSf.=_{n3}]$; n_1, n_2, n_3 are the values of the object's respective fields (attributes).

Partitive Classification

This stage consists in the formation of a hierarchy of phenomena systems rooted in the part/whole foundation:

– $[(\{L?_1\} \sqcup \{L!_1\});\ \{L!_1\} \sqcap \exists hasCorrespondence_1.\{L?_1\};\ \exists hasOS?_1.=_{n1,1} \sqcup \exists hasOS!_1.=_{n1,2} \sqcup \exists hasOSf_1.=_{n1,3}]$;

– $[(\{L?_2\} \sqcup \{L!_2\});\ \{L!_2\} \sqcap \exists hasCorrespondence_2.\{L?_2\};\ \exists hasOS?_2.=_{n2,1} \sqcup \exists hasOS!_2.=_{n2,2} \sqcup \exists hasOSf_2.=_{n2,3}]$;

…

– $[(\{L?_j\} \sqcup \{L!_j\});\ \{L!_j\} \sqcap \exists hasCorrespondence_j.\{L?_j\};\ \exists hasOS?_j.=_{nj,1} \sqcup \exists hasOS!_j.=_{nj,2} \sqcup \exists hasOSf_j.=_{nj,3}]$.

Thus, the successive building of the three classifications described above allows identification of the cause of the system's existence or the requirements on the system in design, the conditions for its existence or possibilities of its creation, as well as the structural and performance features of the analyzed or designed system.

CONCLUSIONS

The conceptual DSOA framework proposed in this article provides the researcher or designer with an efficient set of tools with diverse options for analyzing and representing complex, hard-to-formalize systems.

First, the identification of a class, to which the analyzed or designed system belongs, provides the analyst or developer with some idea about the system's purpose. This identification also allows one to refine the requirements on the system by describing concrete classes of its inputs and outputs, concrete input-to-output conversion class, and concrete classes of object characteristics. This allows one to clearly define and preset the system's external determinant.

Second, the tracking of the stages of the system's buildup (or creation) allows one to outline more specific requirements on the system to the level of describing concrete input and output flows as phenomena, concrete functional requirements, and concrete object characteristics. On the other hand, this tracking allows one to clearly define and set the system's internal determinant.

Third, the decomposition of the requirements on the system as a phenomenon (or its internal determinant) provides the analyst or developer with the concept of the methods that provide the compliance of the subsystems of the analyzed or designed system with its internal determinant, that is, of the system's performance or building methods.

The described way of analyzing and representing systems by determinant system–object analysis will be useful and promising in designing hard-to-formalize organization systems with people as their integral constituent.

FUNDING

This work was financially supported by the Russian Foundation for Basic Research, project nos.18-07-00355a, 19-07-00290a, 19-07-00111a, and 19-29-01047mk

REFERENCES

1. Kachala, V.V., *Obshhaja teorija sistem i sistemnyj analiz* (General Systems Theory and Systems Analysis). Moscow: Goryachaya Liniya–Telekom, 2017.
2. Antonov, A.V., *Sistemnyi analiz* (Systems Analysis), Moscow: Vysshaya Shkola, 2004.
3. Volkova, V.N., *Teoriya sistem i sistemnyi analiz v upravlenii organizaciyami. Spravochnik. Uchebnoe posobie* (Systems Theory and Systems Analysis in the Management of Organizations: Reference Guide and Handbook), Volkova, V.N. and Emel'yanova, A.A., Eds., Moscow: Finansy i Statistika, 2006.
4. Matorin, S.I., Zhikharev, A.G., Zimovets, O.A., et al., *Teoriya sistem i sistemnyj analiz* (Systems Theory and Systems Analysis), Matorin, S.I., Ed., Moscow: Direktmedia Pablishing, 2020. http://biblioclub.ru/index.php?page=book&id=574641. Cited March 12, 2020.
5. Matorin, S.I., Zimovets, O.A., and Zhikharev, A.G., System-wide principles in terms of the system-object approach "Unit-Function-Object"), *Tr. Inst. Sist. Anal. Ross. Akad. Nauk*, 2016, no. 1, pp. 10–17.
6. Matorin, S.I., Zhikharev, A.G., and Zimovets, O.A., Calculus of objects in the system-object method of knowledge representation, *Iskusstv. Intellekt Prinyatie Reshenij*, 2017, no. 3, pp. 95–106.
7. Matorin, S.I. and Zhikharev, A.G., Accounting for regularities in system-object modeling of organizational knowledge, *Iskusstv. Intellekt Prinyatie Reshenij*, 2018, no. 3, pp. 115–126.
8. Matorin, S. I., Zhikharev, A. G., and Mikhelev V.V., Accounting for system-wide patterns in the conceptual modeling of conceptual knowledge, *Iskusstv. Intellekt Prinyatie Reshenij*, 2019, no. 3, pp. 12–23.
9. Mel'nikov, G.P., *Sistemologiya i yazykovye aspekty kibernetiki* (Systems Science and Language Aspects of Cybernetics), Moscow: Sov. Radio, 1978.
10. Shreider, Yu.A. and Sharov, A.A., *Sistemy i modeli* (Systems and Models), Moscow: Radio i svyaz', 1982.
11. Ackoff, R. L. General system theory and systems research: Contrasting conceptions of systems science. *View on General Systems Theory: Proc. of the Second Systems Symposium at Case Institute of Technology*, Mesarović, M.D., Ed., New York: John Wiley & Sons, 1964, pp. 51–60.
12. Brekhovskikh, S.M., *Osnovy funktsional'noi sistemologii material'nykh ob"ektov* (Fundamentals of Functional Systems Theory of Material Objects), Moscow: Nauka, 1986.
13. Matorin, S. I. and Mikhelev, V.V., An analysis of the role and structure of information (conceptual) systems, *Autom. Doc. Math. Linguist.*, 2020, vol. 54, no. 2, pp. 105–112. https://doi.org/10.3103/S0005105520020077
14. *The Description Logic Handbook: Theory, Implementation, and Applications*, Baader, F., Calvanese, D., McGuinness, L., Nardi, D., and Patel-Schneider, P. F., Eds., Cambridge: Cambridge Univ. Press, 2003.
15. Baader, F. and Sattler, U., Expressive number restrictions in description logics, *J. Logic Comput.*, 1999, vol. 9, no. 3, pp. 319–350. https://doi.org/10.1093/logcom/9.3.319

Translated by S. Kuznetsov