A NEW METHOD OF SYSTEMOLOGICAL ANALYSIS COORDINATED WITH THE PROCEDURE OF OBJECT-ORIENTED DESIGN. II[1]

S. I. Matorin

UDC 519.688 519.683.7

The paper presents the results of development of an object-oriented systemological method used to design complex systems. A formal system representation, as well as an axiomatics of the calculus of systems as functional flow-type objects based on a Node-Function-Object class hierarchy are proposed. A formalized NFO/UFO analysis algorithm and CASE tools used to support it are considered.

Keywords: *systemological analysis, object-oriented design, NFO/UFO analysis.*

The object-oriented approach to the analysis and design of information and organizational systems have given rise to a problem on the orthogonality of the well-known methods of systems and object-oriented analysis due to increase in their complexity and interest in the results and methods of systems studies. This problem was solved within the framework of a new scientific direction in the systemological analysis of complex dynamic objects developed in the Laboratory of Knowledge Acquisition of the Kharkov National University of Radio Electronics [1]. This solution is based on the use (in the analysis and design of systems) of the base hierarchy of classes and a formal-semantic adaptive normative system [2], which allows us to decompose the system being designed, which is structural, object, and functional simultaneously. Such possibilities of the base hierarchy of classes (and the normative system based on it) are determined by the fact that the given hierarchy is a classification of system components considered as functional objects. The classification is constructed through a systemological classification analysis, which results from the solution of the well-known problem of natural classification [3].

An object-oriented methodology of systemological analysis and design (OMSAD) for complex dynamic objects is presented in [2]. It is a result of a knowledge-oriented development of systems analysis, on the one hand, and a systemological development of the object approach, on the other hand. We will continue a purposeful development of the methodology. Let us consider an UFO-hierarchy of classes or NFO-hierarchy (Nodes-Functions-Objects), which is a part of the base hierarchy. It allows us to create a more formalized algorithm for object-oriented systemological analysis and an efficient CASE-tool for its support and simulation of organizational and intelligence systems.

MODIFICATION OF THE BASE HIERARCHY OF CLASSES

The base hierarchy of classes has some constraints imposed by the operation of abstraction-concretization (generalization-specialization) due to which taxonomy structures are created. As a result of modification of the base hierarchy, the UFO-hierarchy is selected, which allows us to remove these constraints. Note that considerating elements irrespective of links (operation without relations and parts without the whole) does not allow us to reflect fully the specificity of actual systems relations. Actually, there are neither elements without links, nor functionality that does not ensure any relations (interactions), nor parts without the whole. These aspects of reality may be integrally considered with regard for both generic-specific and partitive relations. If we consider not only the operation of system elements supporting a system

[1]For the first part, see Cybernetics and Systems Analysis, Vol. 37, No. 4, 2001.

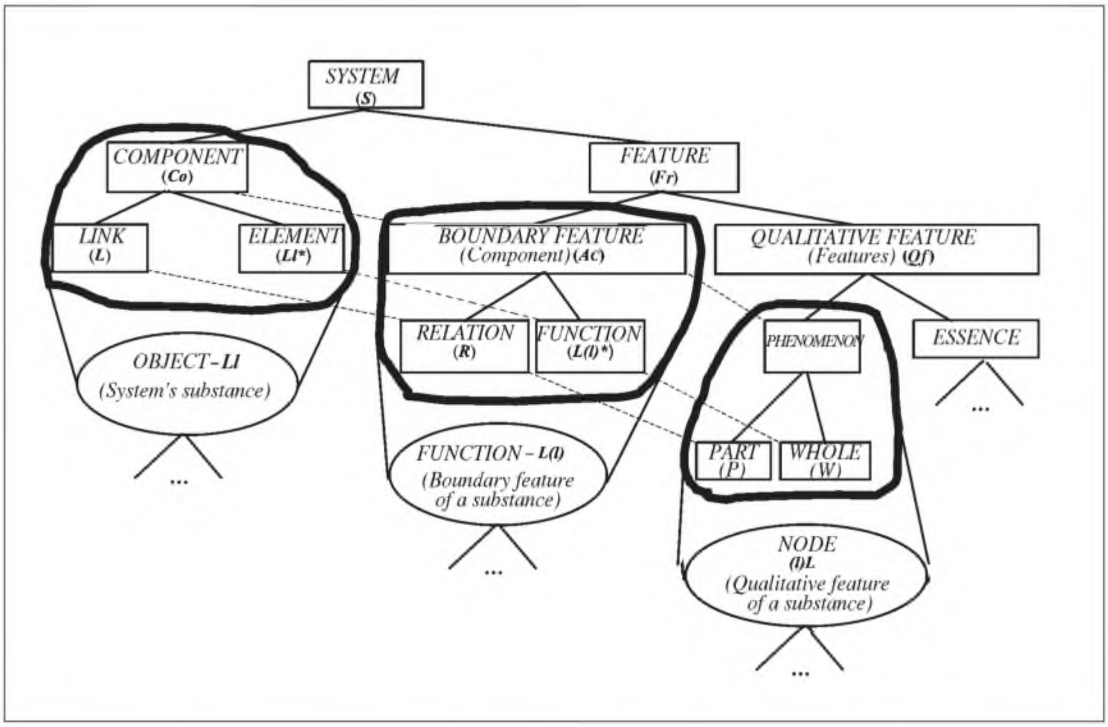

Fig. 1. Obtaining hierarchy of classes Nodes–Functions–Objects from the base hierarchy.

but also their role in the structure of links of this system as flow-type nodes, then we can modify the above hierarchy of classes as is shown in Fig. 1. Thus, the UFO-hierarchy is a conceptual classification model of elements considered as objects functioning at flow-type nodes of the system. In Fig. 1, L or l designates a class of elements of any nature from some deep layer of the systems being considered; the systems exchange these elements.

As a result, the sense of concepts, i.e., the interpretation of classes of the base hierarchy (AN ELEMENT, A FUNCTION, A PHENOMENON) is determined in the UFO-hierarchy (AN OBJECT, A FUNCTION, A NODE) with regard for the obligatory presence of links for elements-objects, and the relations for functions (in a sense wider than mathematical). A flow-type NODE is considered as an integrity existing necessarily as a structural part of greater integrity. Such an approach (to a greater extent than that realized in [2] at a level of base hierarchy) corresponds to the conceptual representations of functional systemology. According to these concepts [4–6], the features of systems are considered as the intrinsic abilities of these systems to support links of some form and (or) to prevent the realization of links of other form, i.e., they are characterized by links with other systems. Any link between systems is the process of mutual exchange of elements of definite deep layers of connected systems. Thus, a feature of a system is understood as manifestation of its activity to be included into links, into exchange flows with other systems in the hypersystem structure. Therefore, the initial (qualitative) features of a system should include its ability to be a node in some structure, and the secondary (quantitative, formal) features should include its ability to function at this node, ensuring balance of the inflow and outflow of input and output links. Thus, the analysis of features of a system as a complete object by means of functional systemology "is based, first of all, on detecting the flows in which it is included as an element of hyperobject, i.e., as a flow-through element in the network of closed exchange flows of the hyperobject. Clearly, detection of these features will be both a complete characteristic of functions of this object and reflection of its integrity, because the balance of inflowing and outflowing flows will not fail to manifest itself in qualitative characteristics in this case [4, 43].

The presented hierarchy, as well as the base hierarchy of classes, described in [2], allows us to generate an adaptive set of alphabetic symbols (a formal-semantic alphabet), uniquely interpreted by their definitions (features) by both the user and the computer. The alphabet is a collection of specific classes (pages) of classification scheme (in this case, nodes, functions, and objects) and also a collection of respective specimens of these classes, from which the object model of the system is created in the course of object-oriented systemological analysis and design.

FORMAL PRESENTATION OF A SYSTEM AS A FUNCTIONAL FLOW-TYPE OBJECT

Let us consider a formalization of the systemological understanding of a system using a logical-mathematical interpretation of components and the features taken into account in the above hierarchies of classes. Clearly as distinct from the result presented in [2] based only on the base hierarchy, some refinement of the formal system representation with regard for NFO-hierarchy will take place in this case. Here, we will italicize classes and designate samples (phenomena) by an ordinary font according to Fig. 1.

As shown in [2], we can formalize the representation of a specific system (phenomenon) as a functional object using an ordered set of classes as follows: $\mathrm{S} = \langle L, \mathrm{R}, Ll^*, \mathrm{L}\,(\mathrm{l})^* \rangle$, where the class L represents a specific class of LINKS, in which a system-phenomenon S participates (i.e., a respective set of elementary components, generating connecting flows), and R is a specific RELATION (as a phenomenon) defined on a set of elementary components, generating links (connecting flows), Ll^* is a specific ELEMENT class (a transformer or a transmitter [2]) to which the system-phenomenon S belongs, and $\mathrm{L}\,(\mathrm{l})^*$ is a specific FUNCTION as a phenomenon (conversion or mapping [2]) given on a set of elementary components flowing through the system S along the input and output links.

The functional links of a system in the structure of a hypersystem (a node) characterize an exterior determinant (D^{ex}) of a system as a phenomenon, and its functionality as an interior determinant (D^{in}) of the given system-phenomenon.

The structure and features of the hierarchies of classes (see Fig. 1) allow us to maintain that $L = \langle Co, R \rangle$, $\mathrm{R} = \langle R, \mathrm{P} \rangle$, $Ll^* = \langle Co, L\,(l)^* \rangle$, and $\mathrm{L}\,(\mathrm{l})^* = \langle L\,(l)^*, \mathrm{W} \rangle$, where R is a specific RELATION class, P is a phenomenon PART defining a specific manifestation of the relation R; $L\,(l)^*$ is a specific FUNCTION class, I is the a phenomenon the WHOLE defining a specific function $\mathrm{L}\,(\mathrm{l})^*$ as a phenomenon. Thus, $\mathrm{S} = \langle Co, R, \mathrm{P}, L\,(l)^*, \mathrm{W} \rangle$.

With regard for partitive relations between classes (NFO-hierarchy), we may maintain the following:

— $\mathrm{P} \vee \mathrm{W} = (\mathrm{l})\,\mathrm{L}$ is the NODE phenomenon corresponding to D^{ex} of the system-phenomenon S and defining its connecting flows (of class L) in the structure of a hypersystem-phenomenon,

— $R \vee L\,(l)^* = L\,(l)$ is a specific FUNCTION class corresponding to D^{in} of the system S. In this case, the domain of definition Dom $L\,(l)$ is a set of elements of input links of the system S, and the domain of variation Im $L\,(l)$ is that of output links,

— $Co = Ll$ is a specific OBJECT class whose element is the system S.

Therefore, $\mathrm{S} = \langle Ll, L\,(l), (\mathrm{l})\,\mathrm{L} \rangle$, i.e., a system, as a functional flow-type object, is in essence characterized by a node in hypersystem structure, a set of functions balancing the node, and a set of objects realizing these functions.

Note that such a representation of a system is in good agreement, for example, with the concept of a generatrix in the Grenader theory of patterns [7], where by a generatrix is meant an object having some attributes and input and output links (which, in turn, are characterized by some indices). In this case, we can consider Ll and $L\,(l)$ as attributes of a generatrix object, and (l)L as links whose indices are types L.

CALCULUS OF FUNCTIONAL FLOW-TYPE OBJECTS

To represent informative conceptual propositions of functional systemology as a deductive theory, it will not suffice to have a formal definition of a system (as a functional flow-type object), initial statements, i.e., axioms are necessary. Let us consider one of the possible axioms of such theory. It is obtained by a logical and mathematical description of the NFO-hierarchy and formalization of the informative axioms presented in [8] (we will speak here only about classes and use Roman type).

The NFO-hierarchy can be presented, for example, by the following pair of logical and mathematical expressions:

$$\exists!\,\mathrm{S} = \{\mathrm{Ll}, \mathrm{Fr}\} : \mathrm{Fr} = \{\mathrm{L}\,(\mathrm{l}), \mathrm{Qf}\} \rightarrow \mathrm{Ll} = \langle \mathrm{S}, \mathrm{L}\,(\mathrm{l}) \rangle \wedge \mathrm{Fr} = \langle \mathrm{S}, \mathrm{Qf} \rangle;$$

$$(\mathrm{l})\mathrm{L} \subset \mathrm{Qf} \rightarrow \mathrm{L}\,(\mathrm{l}) = \langle \mathrm{Fr}, (\mathrm{l})\mathrm{L} \rangle,$$

where S is a system, Ll is a functional object, Fr is a feature, L (l) is a function, Qf is a qualitative feature, and (l)L is a flow-type node.

These logical and mathematical expressions can be considered as axioms of the theory of functional flow-type objects or respective calculi of systems. They allow us to specify formally logical definitions of signs for system elements and their features, i.e., to define their semantics algorithmically (constructively).

It can be seen from the presented expressions that $Ll = < S, L\,(l) >$, i.e., an OBJECT is a SYSTEM characterized by a FUNCTION, $L\,(l) = < Pr, (l)L >$, i.e., a FUNCTION is the FEATURE to be a NODE. Thus, we can say that $Ll = < S, Pr, (l)L >$. However, in this case $Pr = < S, Qp >$. Therefore, we can maintain that $Ll = < S, Qp, (l)L >$, i.e., a functional flow-type object can be described by concepts that cannot be defined in the given formal system: a SYSTEM, QUALITY, and a NODE. This means that all types of the class OBJECT-Ll introduced for a specific data domain may be defined formally and logically with the help of a finite sequence beginning with the above finite sequence consisting of undefined classes and a chain of types of the class NODE-(l)L inserted one in another.

The introduced axioms realize the so-called epistemological approach in systems studies within which "all possible systems are considered, irrespective of whether they actually exist. Then the laws concerning a subset of systems are postulated" [9, p. 83]. These axioms are an analytical expression of the conceptual model of data domain of extremely high level of abstraction and, therefore, are a base for a formally semantic alphabet of systems calculus as functional flow-type objects, which describes an arbitrary data domain.

To apply the theory to a specific data domain, it is necessary to have a possibility to form alphabetic symbols possessing not abstract but specific semantics, which corresponds to the given data domain. As to practical application of the formal axiomatic theory, it is clear that a possibility must exist of specializing an abstract conceptual model (axioms) and constructing a model of a specific data domain.

Let us consider the algorithm for constructing the analytical conceptual model of a specific data domain. Such an algorithm is based on consideration of the types of linking elements — subclasses of the class L (we compare subclasses of the class B of links in [2]), providing interaction of the elements in the system under consideration.

To specialize an abstract conceptual model to a conceptual model defining the semantics of signs for system elements in whose structure two types of links (material M and information I) are taken into account, it is necessary to apply the following rule of specializing classes in NFO-hierarchy:

$$Yx \subset Ll,\ Yx = < Ll, Y\,(x) > \leftrightarrow Y(x) \subset L(l),\ Y(x) = < L\,(l), (x)Y > \leftrightarrow (x)\,Y \subset (l)L.$$

Shortly,

$$Yx^{i} = < Ll, (x)Y^{1}, (x)Y^{2}, \ldots, (x)Y^{i} >,$$

$$\text{where } (x)Y^{1} \subset (l)L, (x)Y^{2} \subset (x)Y^{1}, \ldots, (x)Y^{i} \subset (x)Y^{i\text{-}1}.$$

According to the rule introduced, we will have the following model for systems with material links for $Y = M$ and $X = M$:

$$Mm \subset Ll,\ Mm = < Ll, M\,(m) > \leftrightarrow M\,(m) \subset L\,(l),\ M\,(m) = < L\,(l),\ (m)M > \leftrightarrow (m)M \subset (l)L.$$

For $Y = M$ and $X = I$, we have the following model for systems with output material links and input information links:

$$Mi \subset Ll,\ Mi = < Ll, M\,(i) > \leftrightarrow M\,(i) \subset L\,(l),\ M\,(i) = < L\,(l), (i)\,M > \leftrightarrow (i)\,M \subset (l)\,L.$$

For $Y = I$ and $X = M$, we have the following model for systems with output information links and input material links:

$$Im \subset Ll,\ Im = < Ll, I(m) > \leftrightarrow I\,(m) \subset L\,(l),\ I(m) = < L\,(l), (m)\,I > \leftrightarrow (m)\,I \subset (l)\,L.$$

For $Y = I$ and $X = I$, we have the following model for systems with information links:

$$Ii \subset Ll,\ Ii = < Ll, I\,(i) > \leftrightarrow I\,(i) \subset L\,(l),\ I\,(i) = < L\,(l), (i)\,I > \leftrightarrow (i)\,I \subset (l)\,L.$$

Shortly,

$$Mm = < Ll, (m)\,M >;\ Mi = < Ll, (i)\,M >;\ Im = < Ll, (m)\,I >;\ Ii = < Ll, (i)\,I >.$$

To specialize further the conceptual model (hierarchy of classes) and obtain models defining the semantics of signs for systems elements in whose structure more specific types of links are taken into account (i.e., types of material links: real

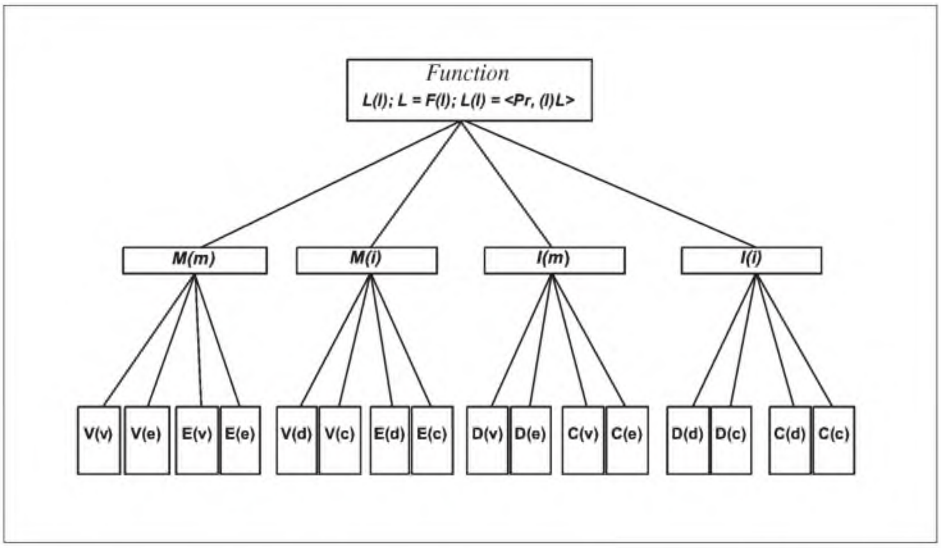

Fig. 2. Classification of the functions of systems as functional flow-type objects (isomorphic hierarchy of nodes and objects).

V and power E and types of information links: data D and control C), it is necessary to replace in the expression the rules of class specialization Ll, L (l) and (l)L by Mm, M (m), and (*m*) *M*, respectively, and to obtain models similar to those shown above for systems with real and power links. Then we should replace Ll, L (l), and (l)L by Mi, M (i), and (*i*) *M*, respectively, and to obtain similar models for systems with all types of material and information links. Further, we should replace Ll, L (l), and (l)L by Im, I(m), and (m)I, respectively and obtain again models for systems with all types of material and information links. Then we should replace Ll , L(l) , and (*l*) *L* by Ii, I(i), and (i)I, respectively, and obtain models for systems with links for data and control.

Shortly,

1) $Vv = < Ll, (m)\,M, (v)V >$, $Ve = < Ll, (m)\,M, (e)V >$, $Ev = < Ll, (m)\,M, (v)\,E >$, $Ee = < Ll, (m)\,M, (e)\,E >$;
2) $Vc = < Ll, (i)\,M, (c)V >$, $Vd = < Ll, (i)\,M, (d)V >$, $Ec = < Ll, (i)\,M, (c)\,E >$, $Ed = < Ll, (i)\,M, (d)E >$;
3) $Cv = < Ll, (m)I, (v)C >$, $Ce = < Ll, (m)I, (e)C >$, $Dv = < Ll, (m)I, (v)D >$, $De = < Ll, (m)\,I, (e)D >$;
4) $Cc = < Ll, (i)I, (c)C >$, $Cd = < Ll, (i)I, (d)C >$, $Dc = < Ll, (i)\,I, (c)\,D >$, $Dd = < Ll, (i)I, (d)D >$.

Hereinafter, we can continue to specialize classes with respect to each newly introduced type of object (a function and a node). Figure 2 gives an example of a hierarchy of functions, an isomorphic (according to the laws of natural classification) hierarchy of classes of nodes and objects, allowing for types of links introduced in [2].

The rules of manipulation with an alphabet (i.e., the rules of constructing expressions syntactically correct and semantically interpreted from the point of view of the given formal system) are an integral part of any formal system, including axiomatic one. Let us consider rules for manipulating symbols of formally semantic alphabet for the calculus of systems as functional flow-type objects, i.e., rules for constructing object models of systems in terms of given calculus.

The rule of balance: $\forall\,(x_i)\,Y^j\ \exists\,Y^j(x_i)$, where $i = 1, \ldots, n$; $j = 1, \ldots, m$, i.e., for any D^{ex} an appropriate D^{in} exists.

This rule means that the qualitative and quantitative balance of the inflow and outflow of input and output functional links must be observed at nodes of the system structure.

The rule of association: $\forall\,Y^j(x_i)\,\exists\,Y^j x_i$, where $i = 1, \ldots, n$ and $j = 1, \ldots, m$, i.e., elements should be linked together according to the qualitative and quantitative characteristics of links inherent in them.

This rule means that a functional element is used according to its interior determinant D^{in}, which is presented by its function.

The above rules are similar to those of systems decomposition considered in [2]. However, in this case, they are a formal corollary of the axiomatics introduced in the present paper. Indeed, from the equality $Y\,(x) = < L\,(l), (x)Y >$ the rule of balance follows, and from the equality $Yx = < Ll, Y\,(x) >$ the rule of association follows (see the rule of classes specializations).

Clearly, the above rules describe the situation formally (ideally). This means that the first rule itself requires only maintaining by the function the flows defined by a node. The rule does not stipulate how (by what system-substance) this functional maintenance is provided. The rule of association requires only that the system has input and output links corresponding to the internal determinant (functional ability). In this case, it is not stipulated whether a medium or hypersystem meet the requirements. Thus, even if the system meets the requirements of one of the rules, the following situation may arise, for example: a system functioning at a node may have also other functional abilities, which are not specified by the node (i.e., by the hypersystem) in this case. Allowance for the first and the second rules simultaneously leads to the following obvious corollary.

Statement 1. $\forall (x_i)Y^j \exists Y^j x_i$, where $i = 1, \ldots, n$ and $j = 1, \ldots, m$, i.e., a functional unit must correspond to the external determinant D^{ex} represented by a flow-type node.

This statement requires already that not only the system has appropriate functional ability but also the operation of the system meets the requirements of the given node. This means that the system operation at the node must provide conversion of inputs submitted by the node directly to the outputs required by it.

We will call the system $Y^j x_i$ functionally corresponding to the node, i.e., satisfying Statement 1, or for which the expression $(y^j)X_i \to Y^j(x_i) \to Y^j x_i$ is true, the system adapted to the hypersystem inquiry. Such understanding of adaptation corresponds to the systemological approach to this concept as the relation of the domain of the required functional states to the domain of possible states [4] and to the concept of systematicity measure introduced in [6]. Therefore, in terms of the calculus of systems as functional flow-type objects, the degree of adaptation of a system can be estimated by the qualitative and quantitative matching of a function necessary for a specific node, for example $L^j(l_i)$, and a function, for example, $Y^j(x_i)$, which is fulfilled actually by the respective system. In this case, integral parameters such as input and output efficiency (for a functional element) and channel capacity (for a connecting element) should be used as quantitative characteristics.

The rule of balance means that any *i*th input (flow, link) flows directly or indirectly through intermediate links to some *j*th outputs (flows, links), and any *j*th output is fed definitely from some *i*th inputs. The rule of balance of inflows and outflows allows us to justify, in terms of the calculus of systems as functional flow-type objects, the validity of systemological presentation of closure of so-called internal flows (links). This justification can be considered as the following statement of the proposed calculus.

Statement 2. If there is a flow in a system, which is not related to continuous flows from the input to the output, then it is closed (forms a cycle), i.e., if there is an element Yx in the system with the function Y(x) forming a flow not associated with input-output flows, then an element Xy exists with the inverse function X(y) compensating (closing) it.

The proof of the statement by contradiction can be reduced to the assumption that the flow not associated with continuous flows from the input to the output does not form a closed cycle. Then this flow will manifest itself at some instant of time as external (functional) one that will violate the rule of balance.

Statement 2 is a formalization of the third rule of system decomposition considered in [2], namely, the rule of closedness.

To solve practical problems and create the method of analysis based on the proposed calculus of systems as functional flow-type objects, it is necessary to consider how objects (elements) of a real system may correspond to functional objects of a formally semantic alphabet. Clearly, far from all real objects are binary nodes with one input and one output. Various combinations of the situations described below are possible.

1. The object is a node with one input and one output, i.e., a function of conversion of one variable. We will call such an object an elementary or alphabetic object, since it will be a specific alphabetic class of the above hierarchy. The simplicity of an object does not mean that it cannot be further decomposed.

2. The object is a node with several inputs and one output, i.e., with a function of conversion of several variables. Such an object is a superposition of several alphabetic objects (functions) combined into one integral (emergent) substance because they ensure one common functionality. The object as a whole inherits from all respective alphabetic classes.

3. The object is a node with one input servicing all outputs and several outputs. Such an object is a composition of different alphabetic objects (functions) combined in one substance due to the uniformity of input flows. The object as a whole inherits from all respective alphabetic classes. Apparently, the disconnection of the functions is not possible or expedient.

4. The object is a node with several inputs and several outputs. Such an object is an aggregate consisting of several functionally independent alphabetic objects, each being an example of a definite alphabetic class of the above hierarchy. The object as a whole inherits from all respective alphabetic classes. These functions can be fulfilled by different objects.

ALGORITHM OF NFO/UFO-ANALYSIS

Representing the systemological theory as a formal semantic calculus of systems as functional flow-type objects based on the NFO-hierarchy and appropriate formally semantic alphabet allows us to formalize the method of object-oriented systemological analysis and simulation of organizational and information systems.

The method allows us to use formalized rules for revealing classes and objects of data domain during an object-oriented analysis and design based on the use of tables [10]. The algorithm of the method consists in coordinated revealing and simulating structural, functional, and substantial characteristics of the system and can be presented by the following main steps:

1. Revealing nodes (intersections) of interval/supporting flows of links in the structure of the system being developed based on functional links of the system as a whole given by requirements for simulation or development.

2. Revealing functionality supporting (providing/balancing) the revealed nodes (in the general case, there can be several functions for one node).

3. Determining objects corresponding to the revealed functionality, i.e., those realizing it.

The method referred to provides automation of the second and third steps through the use of formally semantic alphabet when a template of NFO-hierarchy is available, which contains ready nodes, functions, and objects appropriate for the given problem. In this case, the first step may be identified with the system analysis, the second with its design, and the third with its implementation.

The method of systemological analysis is being used now in developing CASE-toolkit of new generation: the UFO-toolkit (User Functional Object) supporting the NFO-analysis procedures and facilitating the analysis and design of complex systems for both professional analysts and experts in various data domains.

TOOLKIT OF UFO-ANALYSIS

The software (UFO-toolkit) for supporting UFO-technology of systemological object-oriented analysis and simulation represents the CASE-toolkit using a knowledge base of special configuration. The tool is intended for construction of object and simulation models of complex dynamic (organizational) systems and is characterized by a series of basic features.

Its use:

— reduces significantly the labor-consumption of design owing to intensified automation of analytical activity,

— increases the objectivity of the analysis and the adequacy of simulation,

— makes it possible to apply in the analysis and simulation the component technology automating the creation of models through the use of ready (alphabetic) functional objects presented in the knowledge base of the Tool in the form of NFO-elements, and

— provides intelligent interaction with the user, in particular, making familiar the ready components (NFO-elements).

The toolkit can be applied:

— in constructing models of the existing and planned business in conducting the re-engineering of business processes,

— in carrying out consulting projects,

— in developing distributed information systems with application of CORBA tools of business objects (BOF),

— in developing technical systems with application of the CALS-technology and the STEP system of standards.

Application of the UFO-toolkit allows us to perform an interconnected analysis, simulation, and design of structure, composition of elements and functional characteristics of systems and processes, including those not having a mathematical interpretation. In this case, there is a real possibility to combine various aspects of the consideration of a system in one object model (diagram of objects interaction) and, thus, the possibility to uniformly construct external and internal models of a business system described by the same simulation language. Moreover, the operation of the designed system is simulated in various variants to determine jams and idle times and to calculate various integral indices.

Tools being created should provide representation of any system (a subsystem, etc.) as a three-element Node–Function–Object structure. Here, a node is a cross point of input and output flows of linking elements in the structure of a system of a layer higher than the given one. The function is the translation of the input into the output, i.e., a process ensuring the balance of inflows and outflows (links) of the given node. The object is a substance realizing the function.

The structure considered (the NFO-element) is a basis both for the decomposition of a complex system and for synthesis of a complex system from simpler parts. The tool provides and supports the classification of NFO-elements based on the classification of flows of linking elements whose intersection form the nodes. Formally, an NFO-element completely corresponds to the above formalism and may also be presented as a class of the UML object-simulation language.

Attributes of such a class have the following signature:

Object1Name_Attribute1Name [multiplicity] [: type] [= initial value]
    [{line of features}];
    ...
Object1Name_AttributeNName [multiplicity] [: type] [= initial value]
    [{line of features}];
    ...
ObjectKName_AttributeNName [multiplicity] [: type] [= initial value]
    [{line of features}];

Functions of a such class have the following signature:

Function1Name [(in_direction] Link1Name: type [initial value]), ...
    ([in_direction] LinkNName: type [initial value]);
([out_direction] Link1Name: type [initial value]), ...
    ([out_direction] LinkMName: type [initial value])]
    [type of the returned value] [line of features];
    ...
FunctionKName [(in_direction] Link1Name: type [initial value]), ...
    ([in_direction] LinkRName: type [initial value]);
([out_direction] Link1Name: type [initial value]),...
    ([out_direction] LinkTName: type [initial value])]
    [type of the returned value] [line of features].

An NFO-element is intended for simulation of any system component, i.e., of both proper functional elements and components usually called the links. This results in the uniformity of the conceptual and formal presentation of all types of systems in the NFO-technology. Thus, the formal systems presentation of all phenomena in the data domain is provided in a strict correspondence with the conceptual provisions of the systemology [4].

The UFO-toolkit has the following options of application:

1. Conceptual simulation 1: modification of the base hierarchy of the types of linking elements and construction of classification of the external functional and the internal supporting links of the system being developed (data domain).

2. Conceptual simulation 2: modification of the Nodes–Functions–Objects hierarchy of classes (NFO-hierarchy) for a specific data domain (construction of a template) or for the system being simulated (project development).

3. Decomposition of the system being developed: filling the table below (System Nodes) with the input and output links and identification of the flow-type nodes of links inside a system, connecting system inputs with its outputs.

4. Object simulation of the system: construction of the model of interaction of the objects of the developed system, beginning with its context model.

5. Simulation of system operation: visualization of time changes in system state (degrees of activity of links and objects) and appearance of flows at object outputs according to their functions.

6. Compiling, visualization, and printing reports: printing diagrams, compiling and printing lists or tables of links, nodes, functions (with reference to the nodes), and objects (with reference to the functions), and compiling, visualization, and printing of reports in the results of simulation.

The use of the tools assumes a preliminary specialization of the classification of flows of connecting elements and NFO-hierarchy with regard for a specific data domain. Such a use should allow us to create templates of classification for various data domains providing the simulation by sets of alphabetic NFO-elements presenting both functional and linking objects of the system, i.e., subsystems. Simulation (in order to analyze and design complex dynamic objects) is carried out by the following steps (see Fig. 3).

1. Construction of a context model (the top level of the hierarchy) of the system being analyzed/designed as a black box with the indicated input and output links (functional), which should be presented in the classification (in the diagram of classes) of links. The process may begin either with the classification of links or with the construction of the context model of the system.

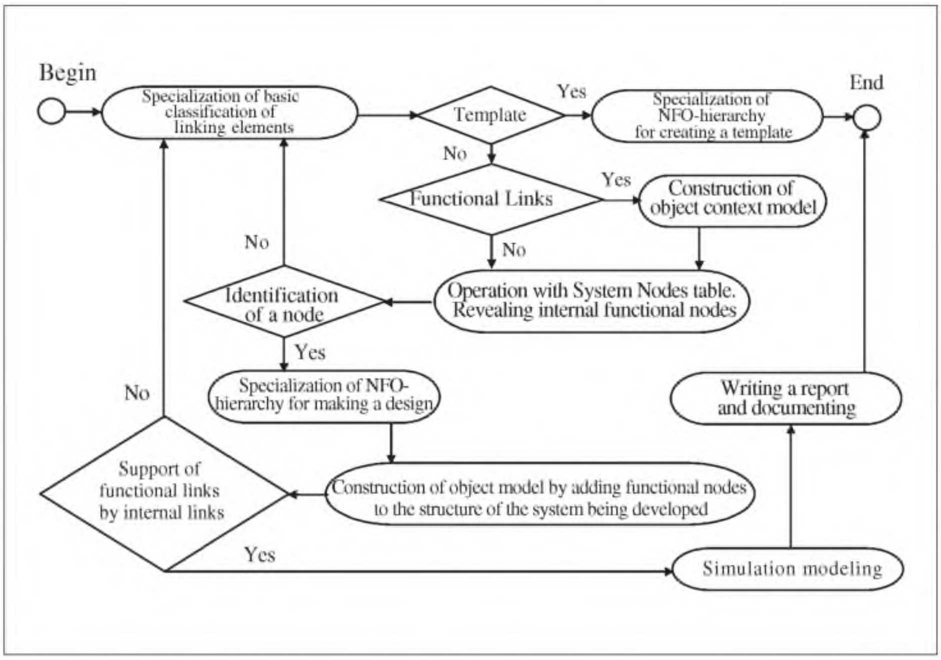

Fig. 3. Block diagram of NFO-analysis as UML activity diagram.

2. Revealing functional nodes in the structure of the simulated system, i.e., the nodes whose function either is already known or can be formulated as a result of design. The above-mentioned table System Nodes is used for this purpose, where input links designate the rows and output links the columns. These links should also be presented in classification of links, with the process beginning with the functional links shown on the context model. If templates (alphabet) of links and NFO-elements are used for simulation, a possibility should be provided for automatic identification of the nodes known to the tool. If templates are not used or there is no necessity to use them, internal links, which support functional links shown on the context model, should be added to the table. The process goes on until the balance of inflows and outflows (links) of the context model is established. Here, the manual identification of nodes should be accompanied by the modification of the classification of links and NFO-hierarchy.

3. Construction of the hierarchical object model of the system being analyzed/designed. During an analysis or design, this model represents (at each level of hierarchy) a population of interconnected functional nodes identified with the help of the table. Here, a function should be selected (from all known types stored in NFO-hierarchy) for each node, which maximally precisely balances the given node. For each function, a respective object (from all known types stored in the NFO-hierarchy) optimally realizing it from the point of view of the given project (customer) should be specified. Therefore, the object model of a system should represent (at each level of hierarchy) a population of interconnected functional objects.

4. Simulation of system operation. This process should be provided through animation of the object model, which is reduced to visualization of time changes in the activities of links and functions of objects (with its possible scaling). It is necessary to account for the characteristics of the objects (for example, reliability) and the possibilities to calculate the indices allowing us to compare various types of the object model with respect to the functions selected for given nodes and objects for given functions (for example, time or cost indices). Moreover, for the functions, which can be described mathematically, a possibility should be provided to calculate their values during simulation, for example, with the use of the mechanism of scripts. The purpose of the simulation is to reveal braking or idle links, functions or objects and to determine the best structure and composition of the system being designed.

5. Providing a possibility of documenting all diagrams of classes, object model (at any level of hierarchy), tables, and simulation results.

Thus, the availability of conceptual and formal tools of the systemological theory of functional flow-type objects, the method of analysis and investigation of objects of this theory (the NFO-analysis), the toolkit (the CASE-toolkit supporting the NFO-analysis) and, therefore, the technique of such a study allows us to speak about the origination and establishment of a systemological technology of the analysis, simulation, and design of complex dynamic systems, which we propose to call the NFO/UFO/JAO-technology. This result allows us to create CASE-toolkits of a new generation representing knowledge-based systems.

REFERENCES

1. S. I. Matorin, "A new scientific trend in systemological analysis of complex dynamic objects," Vestn. KhGPU. New Solutions in Modern Technologies, No. 81, 15-18 (2000).
2. S. I. Matorin, "A new method of systemological analysis coordinated with the object-oriented design procedure. Part 1," Kibern. Sist. Analiz, No. 4, 119-132 (2001).
3. E. A. Solovyeva, The Natural Classification: Systemological Fundamentals [in Russian], KhTURE, Kharkov (1999).
4. G. P. Mel'nikov, The Systemology and Language Aspects of Cybernetics [in Russian], Sov. Radio, Moscow (1978).
5. M. F. Bondarenko, E. A. Solovyeva, and S. I. Matorin, Fundamentals of Systemology [in Russian], KhTURE, Kharkov (1998).
6. M. F. Bondarenko, S. I. Matorin, and E. A. Solovyova, "Analysis of systemological tools for conceptual modeling of application fields," Automatic Document and Math. Linguistics, Allerton Press Inc., New York, **30**, No. 2, 33-45 (1997).
7. W. Grenander, Lectures on the Pattern Theory. 1. Synthesis of Patterns [Russian translation], Mir, Moscow (1979).
8. S. I. Matorin, "Systemology and object-oriented approach (problems of formalizing and prospects of combining)," NTI, Ser. 2, No. 8, 1-8 (2001).
9. J. van Gig, Applied General Systems Theory [Russian translation], Mir, Moscow (1981).
10. S. I. Matorin, "The NFO analysis," ASU i Pribory Avtomatiki, No. 4, 3-12 (2001).