UDC 519.688+519.683.7

APPLICATION OF PATTERN THEORY IN FORMALIZATION OF SYSTEMOLOGICAL UFO ANALYSIS

S. I. Matorin and D. B. El'chaninov

A systems analysis method is considered that agrees with the object-oriented paradigm for setting up information systems. Its procedures are formalized by pattern-theory methods, and also on an original approach to a system as a unit, function, and object.

INTRODUCTION

Object-oriented methodology is used for developing information systems as a basic way of setting up the software tools for the analytic interpretation of system activities. However, object-oriented analysis and design (OOAD) does not possess any of its own methods of analysis or simulation, and there is a lack even of methods for simply identifying the classes and objects needed for simulation. Using the widely advertised UML does not alter the situation because that is only a language, not a method.

An apparently natural use of systems analysis methods for OOAD is ineffective because methods in traditional system structure analysis are "completely orthogonal to the principles of object-oriented design" (p. 161 of [1]). Systems analysts have reached the unambiguous conclusion that one needs to diverge from traditional system methods in OOAD and instead use additional facilities, because existing system analysis methods "nevertheless do not allow one to identify an appropriate set of objects for the system: object observation is still controlled by views, intuition, and insight" (p. 26 of [2]).

This orthogonality occurs because traditional methods do not identify the hierarchy of classes in the subject area, i.e., do not handle conceptual classification simulation (CCS), and also do not support the object-oriented concept of encapsulation (separating the interface of an object from its realization).

The latter in turn indicates that the system in traditional analysis and in the systems approach is considered as a set of elements, properties, states, and so on. In the concept of a set, on the other hand, an element or part is primary in relation to the set (the whole). A set exists if and only if its elements are specified in some way. The set-theoretic concept of a system also does not provide for encapsulating the objects identified during the analysis (necessary for OOAD), since being a set is a phenomenon whose internal content cannot be identified. In the system concept, on the other hand, the concept of the system (the whole) is primary, which may or may not be represented as a set of interacting parts. Within the systems approach (systemology), emphasis is placed on the integrity and functioning of the system, which enables one to match up the systemological analysis with the OOAD requirements [3].

The lack of **OOAD** methods and the orthogonality mentioned above make it possible to set up a method of analysis and simulation that is simultaneously a system method and is in agreement with OOAD.

We consider the results obtained by systemology.

METHODOLOGY OF SYSTEMOLOGICAL UFO ANALYSIS

Developments in the systemological approach [4] to poorly formalized subject areas have led to a new systems theory that overcomes the set-theoretic approach to systems and instead describes specific system properties and relations [3]. The concepts from that theory give a formal semantic system with an adaptive alphabet, which serves as the basis of a new systems analysis method (UFO analysis), which agrees with the requirements and procedures of **OOAD** in information systems [5, 6].

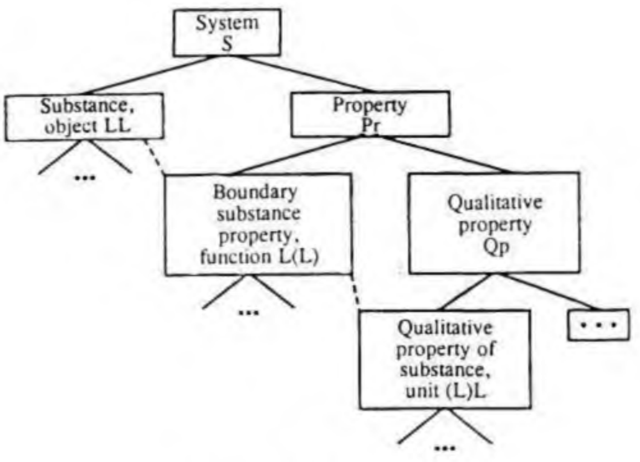

System S
Substance, object LL
Property Pr
Boundary substance property, function L(L)
Qualitative property Qp
Qualitative property of substance, unit (L)L
...

Fig. 1. Basic classification.

This agreement has been obtained by solving the following auxiliary problems:

1) setting up a method of identifying a class set as required and suitable for OOAD; and

2) introducing CCS procedures into systems technology.

Both of these amount to setting up a universal conceptual classification model for any subject area. That model constitutes a classification of all the components and properties of a system. The abstract level is considered as a basic one and specifies the category structure, which determines the method of selecting the classes needed for the simulation. The particular (sheet) classes in this classification scheme (and also examples of them) are used to simulate a particular subject area. They also provide the alphabet for the normative system in the new systemological analysis method.

The basic classification is a high-level conceptual classification model (taxonomic and parametric) for forms of system considered as flow objects and also for forms of their functions and structures (Fig. 1).

This classification is a modification of the [3] scheme, and the system concept is at a limiting level of abstraction, and thus as in (3) represents a unique category (root class).

As the principal forms of system we consider a substance (i.e., an object) as a carrier of the properties, while a property is considered as a characteristic of the carrier. This approach is justified by the following circumstances. First, the types of property that differ most extensively are boundary ones (which characterize the space and time restrictions on certain qualitative properties) and qualitative ones (which are identified apart from the parameters of those sense organs that allow one to observe the spatial and time characteristics of the outside world) [7, 8]. Secondly, the logic structure of a phenomenon with boundaries may be defined as objects or things, while the qualities are defined as properties. "Each thing has spatial characteristics and spatial boundaries, which cannot be said about an individually taken property, in which the expression 'having spatial boundaries' is inapplicable to features of any property of a thing, so one thing differs qualitatively from another" (p. 43 of [9]). Also, some philosophers consider that thinking originally gives rise to categories that reflect the properties and their carriers. We additionally note that the class hierarchy such as in the popular computing environment G2 for devising expert systems, has a unique root class 'item-or-value' with derivatives in the classes 'item' and 'value'.

It is thus best to distinguish primarily objects having boundary properties in the complete system set, of which the most important are the functional properties or functions. At the same time, it is best to distinguish properties characterizing qualities (qualitative properties).

In systemology, the properties of a system are understood as signs of its activity and are included in these connections and exchange flows with other systems in the supersystem structure. This means that one analyzes system properties by means of functional systemology in terms of an integral object "based primarily on the observation of those fluxes in which it is included as an element of a superobject, i.e., as a flow element in a network of closed exchange flows in the superobject. Naturally, these qualities will be observed simultaneously and a sufficiently complete characteristic of the function of this object is thereby provided, which gives an expression of the integrity, since in qualitative characteristics one cannot get a balance between the incoming and outgoing flows in that case" (p. 43 of [7]).

It is thus best to consider as qualitative properties of a substance or object its capacity to be a unit in a certain structure. The capacity to function at that unit supports the balance between influx and efflux on the incoming and outgoing links, i.e., the capacity to transform the elements of the incoming links into elements of the outgoing

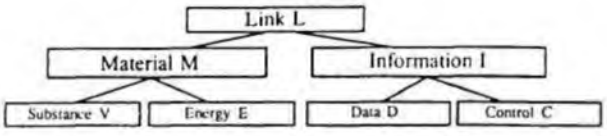


Fig. 2. Initial link classification.

ones, which will constitute boundary properties of the object or system (formal ones, in the present case functional ones).

The objects in this hierarchy are thus classified in accordance with the functions they perform. From the formal logic viewpoint and on the basis of the structural properties of that classification, one can write a genus-species definition: "An object is a system characterized by a function". This completely corresponds with the systemological concept of a system as an object with functions on the basis that a system is always also a flow object. Then also in accordance with the systemological approach, a function of an object is considered as one of its internal determinants, i.e., the cause of certain internal substance characteristics of it.

The functions in this hierarchy are classified in accordance with the flow units that they constitute in the system structure. Consequently, from the viewpoint of formal logic and on the structural properties of that classification, one can write a genus-species definition: "A function is a property or capacity to be a unit in a system structure". This means that a function is characterized by the elements in the incoming and outgoing links on which flows are exchanged with other objects. The region of definition for that function is the set of elements forming the input flows (links), and also the region of values, namely the set of elements forming the outgoing flows (links).

In accordance with the systemological approach, the flow (link) characteristics of a unit in the supersystem structure are considered as an external determinant for the system, i.e., the cause of a certain internal determinant or current function.

This scheme resembles the [3] one in constituting a taxonomic parametric classification, in which the objects are classified in accordance with their properties, forming part of the general class hierarchy. This means that in this hierarchy, as in [3], for each object (class or concept) there is not only a generic feature (higher-level class) but also a concept (class) having a species difference in content for the concept, i.e., the property of the object. The dashed lines in Fig. 1 show the relationship of a concept with its species difference (class and properties). As this classification is parametric, it enables one to classify system species in accordance with their property species, i.e., to incorporate the natural classification laws [10].

If one considers a system as a flow object whose function is due to a function in a superobject (supersystem), one can represent any particular system S* in terms of a class hierarchy, as in [3], with that system being an example of class S, and put formally as the triple $S^* = \langle LL, L(L), (L)L \rangle$, in which (L)L is a particular unit in the supersystem structure ((L) is an incoming link, L an outgoing link), with L(L) the class of functions that balance the given unit ((L) is the argument and L the function), and LL is the class of objects that realize the given functions (L object input, L object output). We call this triple a UFO element.

Here L denotes the type of elements with a material or information nature ($L = \{M, I\}$, with M a material element and I an information one; $M = \{V, E\}$, where V is a substance element and E an energy one; and $I = \{D, C\}$, where D is of data nature and C is of control nature, and so on), which applies at a certain level in the connected systems, and which these systems use to exchange data through their links, which are considered as flows (Fig. 2). This representation of a system as a phenomenon corresponding to an example of a UFO element agrees well with the concept of generator in pattern theory [6, 11]. By generator in that theory [12, 13] one understands an object having certain features α, as well as the incoming and outgoing links (in turn characterized by certain parameters β). In our case (Fig. 3), one can consider the examples of classes LL and L(L) as features of the generator, while (L) and L are links whose parameters are of type L (Fig. 2). Then the generator g_i as an example of UFO element takes the form $g_i = \langle L_2^i I_1^i, L_2^i(L_1^i), (L_1^i)L_2^i \rangle$.

Pattern theory involves the assumption that is a source that generates the set of generators $G = \{g_i\}$. In our case, that source generates UFO elements (generators) and is shown in Fig. 1 as part of the basic classification.

The hierarchy there constitutes the basis of a normative system in the new method of systemological analysis. The alphabetic symbols are UFO elements representing particular classes (sheets) in the classification scheme, and also the corresponding examples of those classes. The examples of UFO elements are used here to construct object models for systems and also to simulate the functioning of these models.

This alphabet consists of characters for the units, functions, and objects representing a system from the viewpoint of its structural, functional, and substance characteristics respectively. Isomorphism in classifying those

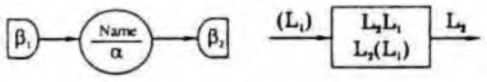


Fig. 3. Graphical formalisms: generator and UFO element.

three (in accordance with the laws of natural classification) enables one to consider a single combined hierarchy containing UFO elements. A UFO element is a system that corresponds to a certain unit (intersection of links or flows) in the supersystem structure, with a defined function (in general not unique) that balances the flows at that unit and a defined object (in general, not unique for each function) that realizes the given function.

Incorporating the natural classification laws into the basic classification gives an unambiguous formal-logic definition for each object in terms of the genus and the species difference. That classification scheme thus acts as an algorithm for the semantics of the normative-system signs, which transforms that system into an algorithmically constructed one. Consequently, this scheme provides an alphabet for that system having not only a completely abstract or strictly mathematical semantics but also an object-oriented form, so that alphabet can be considered as a formal-semantic one, while the normative system is also formally semantic. This implements the current suggestions of informatics experts, who consider that there has long been a practical need to transfer from formal mathematical analysis of information phenomena to content analysis and the driving force in self-organizing systems of social nature [14].

Using a classification to generate alphabetic characters that constitute examples of the corresponding classes provides, as in [3] a distinctive feature for the alphabet. An ordinary formal system with a finite or infinite formal alphabet uses a finite and restricted number of initial concepts corresponding to the signs in the alphabet. This means that completely different subject areas are simulated by means of the same set of alphabetic characters in the normative system of some traditional system analysis method. On the other hand, specifying an alphabet by means of a classification can change the composition of the initial concepts and the corresponding alphabetic characters, i.e., allows one to adapt the alphabet and the normative system in accordance with the subject area.

This hierarchy (units, functions, and objects) enables one to use a detailed set of simulation facilities, i.e., alphabetic characters, to handle a particular task. For example, to simulate information business (such as organizations in the media), one can distinguish particular classes related to the data links and classes related to substances and energy, as well as ones in the form of abstract classes; to simulate electrical power businesses, it is necessary to specify the classes joined by energy links; and to simulate the transportation companies, one uses classes with substance links; and also to simulate production, one uses classes that simulate obtaining substances of appropriate form. The only thing overlooked here is a principle in accordance with which the properties of objects (functions and units) used to set up an object model are determined by their parametric taxonomic classification, i.e., by the basic class hierarchy incorporating those properties.

That hierarchy can be represented as the following pair of expressions in mathematical logic:

$$\exists! S = \{LL, Pr\} : Pr = \{L(L), Qp\} \rightarrow LL =$$
$$\langle S, L(L) \rangle \wedge Pr = \langle S, Qp \rangle;$$
$$(L)L \subset Qp \rightarrow L(L) = \langle Pr, (L)L \rangle,$$

in which S is a system, LL an object, Pr a property, L(L) a boundary property or function, Qp a qualitative property, and (L)L a flow unit.

To produce generators in a particular case, one must be able to form alphabetic characters (UFO elements, i.e., generators) having a particular semantic structure instead of an abstract one and corresponding to the subject area. Naturally, there must be scope for specializing the abstract conceptual model and constructing a model for the particular subject area. The algorithm for such construction of an analytical model is based on incorporating the forms of link (subclasses of class L) that provide interaction for the subsystems.

Such specialization requires one to use the following rule for class specialization:

$$YX \subset LL,\ YX = \langle LL, Y(X) \rangle \leftrightarrow Y(X) \subset L(L),$$
$$Y(X) = \langle L(L), (X)Y \rangle \leftrightarrow (X)Y \subset (L)L.$$

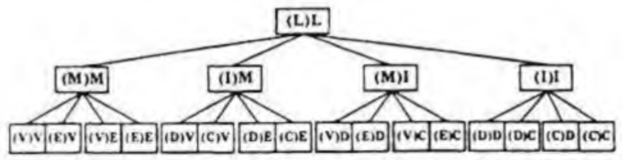


Fig. 4. Classification of alphabetic UFO elements from their units.

The following models then apply in accordance with this rule for any system that incorporates two forms of link: material M and information I: for Y = M and X = M for systems with material links:

$$Mm \subset LL,\ Mm = \langle LL,\ M(M) \rangle \leftrightarrow M(M) \subset L(L),$$

$$M(M) = \langle L(L),\ (M)M \rangle \leftrightarrow (M)M \subset (L)L;$$

and with Y = M and X = I, for systems with outgoing material links and incoming information ones:

$$Mi \subset LL,\ Mi = \langle LL,\ M(I) \rangle \leftrightarrow M(I) \subset L(L),$$

$$M(I) = \langle L(L),\ (I)M \rangle \leftrightarrow (I)M \subset (L)L;$$

while with Y = I and X = M, for systems with outgoing information links and incoming material ones:

$$Im \subset LL,\ Im = \langle LL,\ I(M) \rangle \leftrightarrow I(M) \subset L(L),$$

$$I(M) = \langle L(L),\ (M)I \rangle \leftrightarrow (M)I \subset (L)L;$$

and finally with Y = I and X = I, for systems with information links:

$$Ii \subset LL,\ Ii = \langle LL,\ I(I) \rangle \leftrightarrow I(I) \subset L(L),$$

$$I(I) = \langle L(L),\ (I)I \rangle \leftrightarrow (I)I \subset (L)L.$$

Further specialization (in the UFO element class hierarchy) gives models defining the alphabetic elements for systems that incorporate more detailed forms of link on account of the consideration of the forms of material link: substance V and energy E, and also forms of information link: data D and control C. As a result of such specialization at this level, one can obtain for example the following classes of UFO elements (Fig. 4), which are defined by the corresponding units and are considered as nonintersecting classes of generators constituting set G.

Assigning the normative-system alphabet some subject-oriented semantics and adaptivity, as in [3], improves the interpretation characteristics and simplifies its use in analysis and simulation in various detailed situations. However, as in [3], this does not reduce the formality of that normative system, since algorithmic semantic and syntactic specification means that it will satisfy all the requirements imposed on a formal system from the viewpoint of explicit and rigorous description of the facilities used in such a formal system for describing the properties and relationships between all the characters, and also from the viewpoint of recognizing all the characters only from their form.

To identify the object structure and to construct an object model, it is necessary to specify certain rules for manipulating the alphabetic characters, which provides for transforming the alphabet obtained in the normative system by means of the proposed class hierarchy.

As the characters in the proposed alphabet by definition constitute various system components (units, functions, objects), the rules for operating with these characters should be based on system relationships considered in the relevant systems approach. Functional systemology considers the following as the basic system relationship: a relationship of supporting the functional capacity of the whole [7]. This enables one to formulate system components and a law of system decomposition as a basic rule in operating with the alphabetic characters: "Elements at level i in the system should be in a relationship of support to the functional capacity of level i + 1 (the systems should support a supersystem, while the subsystems should support a system, and so on)".

That law is obeyed by obeying the following rules for system decomposition implied naturally from the concepts of functional systemology:

1) the correct connection between objects (systems) in accordance with the qualitative and quantitative characteristics of their links (attachment rule);

2) the provision of qualitative and quantitative balance between influx and efflux on the incoming and outgoing functional links, i.e., the correct choice of the internal determinant (function) in accordance with the external one: a unit or balance rule;

3) provision of match in the characteristics of the system substance (object) with the internal determinant: a function or realization rule; and

4) closure of the internal links: closure rule.

Pattern theory deals with similarity transformations, i.e., mappings of G into itself that do not produce a generator from the class. In our case, a similarity transform takes the form: $f: G \to G$; $f(g_i) = g_j$, in which g_i and g_j are such that $(L^i)L^i = (L^j)L^j$. The detailed form of f is determined by the particular subject area.

Similarity transformation makes it possible to formalize the construction and adaptation of object models. That formalization is based on representing object models as linked alphabetic generators (UFO elements), which in pattern theory corresponds to compiling configurations from generators. The set R of regular configurations is distinguished on the basis of the rules and constraints on permissible combinations of the generators.

The above rules for system decomposition in our case constitute the main constraint on the permissible generator combinations, i.e., examples of UFO elements.

Here the attachment rule can be given a formal description as follows: two generators g_i and g_i may be attached one to other if at least one of the following is obeyed: $L_2^j = (L_1^i)$; $L_2^i = (L_1^j)$. This rule sets constraints on the regular configuration structure.

This rule concerns the classes of alphabetic generators (UFO elements), but there is practical value for simulation not only in such generators but also in ones that are combinations of alphabetic ones (Fig. 4). For example, from the classes (V)V and (E)D one can obtain the combined class $(V, E)E, D \subset (V)V \cup (E)D$.

In order to introduce these combinations of classes of alphabetic generators $(X_1Y_1, (X_2)Y_2, \ldots, (X_i)Y_i, \ldots, (X_n)Y_n$, we use the following rules for alphabet combination:

1. $(X_1, X_2, \ldots, X_i, \ldots, X_n)Y_1, Y_2, \ldots, Y_i, \ldots, Y_n \subset \bigcup_{i=1}^{n} (X_i)Y_i$

2. If $X_i = X_j$, then $(X_1, \ldots. X_i, \ldots, X_{j-1}, X_j, X_{j+1}, \ldots, X_n) = (X_1, \ldots, X_i, \ldots, X_{j-1}, X_{j+1}, \ldots, X_n)$.
3. If $Y_i = Y_j$, then $Y_1, \ldots, Y_i, \ldots, Y_{j-1}, Y_j, Y_{j+1}, \ldots, Y_n = Y_1, \ldots, Y_i, \ldots, Y_{j-1}, Y_{j+1}, \ldots, Y_n$.

Regular configurations composed of combinations of alphabetic generators are constituted in accordance with this rule, which extends the attachment rule described above for the case of several types of inputs and outputs.

The alphabet combination rule enables one to consider various combinations of the situations described below.

1. An object (system) is a unit with one input and one output, i.e., is a transformation function for one variable. Such an object will be called an elementary one or an alphabetic object, since it will be an example of a particular alphabetic class of the above hierarchy. The fact that the object is elementary does not mean that it cannot be decomposed further.

2. The object (system) is a unit with several inputs and one output, i.e., it is a transformation function for several variables. That object is a composition of several alphabetic objects (functions) combined into one integral (emergent) substance in connection with the fact that they support a single general functionality. The object as a whole will inherit from all the corresponding alphabetic classes.

3. The object (system) is a unit with one input and several outputs, which are served by the single input. That object is a superposition of differing alphabetic objects (functions) combined into a single substance because of identity in the input fluxes. The object as a whole will inherit from all the corresponding alphabetic classes. Mostly speaking, dissecting these functions is not possible or desirable.

4. The object (system) is a unit with several inputs and several outputs. That object clearly constitutes an aggregation of several functionally independent objects, each of which is an example of a certain class of the first, second, or third type as described above. The object as a whole will inherit from all the corresponding alphabetic classes. In principle, these functions may be performed by different objects.

In pattern theory, one distinguishes more regular or perfect configurations in regular configuration space by introducing a measure. Systemology in turn uses a measure on the system structure to evaluate it, with the structure defined as the relation between the regions of required functional states and the region of possible ones. The first region constitutes an external query to the system from the supersystem, and in systemology it is called the external determinant, while the region of possible states representing the actual system functioning is called the internal determinant, which determines the internal (supporting) system characteristics (i.e., the subsystems).

The above attachment rule does not give the characteristics of a particular configuration and instead only defines the class of configurations that are structurally similar. To introduce a measure and to distinguish these configurations one from another, one must apply constraints that will enable one to determine the detailed characteristics of a given structure with regular configuration. For that purpose we describe formally the balance and realization rules.

Balance rule. At a unit in a regular configuration, there should be a balance between the influx and efflux on the entering and leaving links. This means that a unit $(L_1)L_2$ in a regular configuration may be put into correspondence with a generator g_i (example of UFO element), while the function $L_2^i(L_1^i)$ that belongs to the class of functions $L_2(L_1)$ fits the expression $L_2^i(L_1^i) \in L_2(L_1)$.

Realization rule. At a unit $(L_1)L_2$ in a regular configuration, there can be only a generator g_i (example of a UFO element) for which the object $L_2^iL_1^i$ belongs to the class of objects L_2L_1, i.e., $L_2^iL_1^i \in L_2L_1$.

Then in terms of UFO elements considered as generator classes one can say that $\bigcup_{i=1}^{n} (X_i)Y_i$ is the class of required functional states of the generator; $\bigcup_{i=1}^{m} Y,(X_t-)$ is the class of possible functional states for the generator; and $\bigcup_{i=1}^{k} Y_iX_i$ is the class of internal supporting (object) characteristics of the generator.

This allows one to introduce a functionally adapted generator (UFO element), which is adapted to the external determinant $\bigcup_{i=1}^{n} (X_i)Y_i$, with the generator having the internal determinant of the form $\bigcup_{i=1}^{\kappa} Y_i(X_i)$, i.e., this generator is one for which the balance rule applies.

Also, one has a substantially adapted generator (UFO element), which in application to an internal determinant $\bigcup_{i=1}^{n} Y_i(X_i)$ gives a generator whose object characteristic is of the form $\bigcup_{i=1}^{n} Y_iX_i$, i.e., this is a generator for which the realization rule applies.

One uses the term simply adapted generator for one whose object characteristic takes the form $\bigcup_{i=1}^{n} Y_iX_i$ in the presence of an internal determinant $\bigcup_{i=1}^{n} Y_i(X_i)$ and external determinant $\bigcup_{i=1}^{n} (X_i)Y_i$.

When one speaks of adaptation, one means that the region of possible states for a generator at a certain unit is, in accordance with common sense, wider than the region of required states determined for that unit. Any other choice of initial material, e.g., to construct or simulate the business system, is meaningless. However, we understand that in fact there are other cases, which imposes the obligation to provide means of describing them.

This adapted generator and the explanation given for it enable one to formalize the system measure μ_s, which is used subsequently to define the more regular configurations:

$$\mu_S = |\bigcup_{i=1}^{n} Y_i(X_i)| * |\bigcup_{i=1}^{n} Y_iX_i| / |(\bigcup_{i=1}^{n} Y_i(X_i)| * |\bigcup_{i=1}^{n} Y_iX_i|)$$

in which $m \geq n$, $\kappa \geq n$; (X_i) or X_i are the input links of system S, while Y_i are the exit links of system S from the hierarchy of link classes (Fig. 2).

We extend the concept of similarity transformation f introduced for the generator set G to the set R of regular configurations, where one must bear in mind that in pattern theory any configuration set is determined by a structure •that in our case (in terms of UFO elements) is characterized by the connection of units and also by a composition, which in our case is characterized by functional objects (functions and objects) of the generators. Then the similarity transformation on set R of regular configurations can be defined as follows:

$$\text{structure}(fz) = \text{structure})z,$$

$$\text{composition}(fz) = \{fg_1, fg_2, ... fg_j, ... fg_h\}.$$

Such transformation is one of the means of formal representation for the process and result in system adaptation (as an UFO element, i.e., a generator) on the basis of the structure, function, and substance. This is provided by extending the concepts of functional and substantial adaptation to any configuration in accordance with the similarity transformation concept, where one considers an adapted configuration also, since any configuration in essence is a higher-level generator.

To analyze and synthesize systems as UFO elements, it is useful to employ a binary operator from pattern theory on the space of regular configurations. In our case, that operator can be introduced as follows.

For two configurations z_1 and z_2 there exist sets $B(z_1)$ and $B(z_2)$ whose elements are external links for the corresponding configurations. Out of the links constituting these sets one can form a list σ_{12} of pair connections between those links. In our case, this can be done only from links of the same type in accordance with the attachment rule. A combined configuration can be denoted by $z_1\sigma_{12}z_2$, where

$$\text{composition}(z_1\sigma_1Z_2) = \text{composition}(z_1) \cup \text{composition}(z_2),$$

$$\text{structure}(z_1\sigma_{12}z_2) = \text{structure}(z_1) \cup$$

$$\text{structure}(z_2) \cup \sigma_{12}.$$

Then the binary operator here is the above attachment rule, whose application gives a list σ_{12} and a combined configuration $z_1\sigma_{12}z_2$. We call this the attachment operator.

Pattern theory also deals with the annihilation operator, which when applied to a certain configuration annihilates all the generators of a given class in it. In our case, we introduce it as follows.

We consider a configuration $z_1\sigma_{12}z_2$ for which the inclusion $B(z_1\sigma_{12}z_2) \subset B(z_1)$ applies, which means that the external links of the configuration z_1 and $z_1\sigma_{12}z_2$ coincide, and also that the external links of z_2 are internal and closed ones for the configuration $z_1\sigma_{12}z_2$. The annihilation operator, which annihilates all generators of the same class as that to which configuration z_2 belongs and leads to our obtaining a configuration z_1 with the set of external links $B(z_1)$.

Annihilation simulates for example the elimination of internal processes that do not increase the usefulness of the product during reengineering of any business.

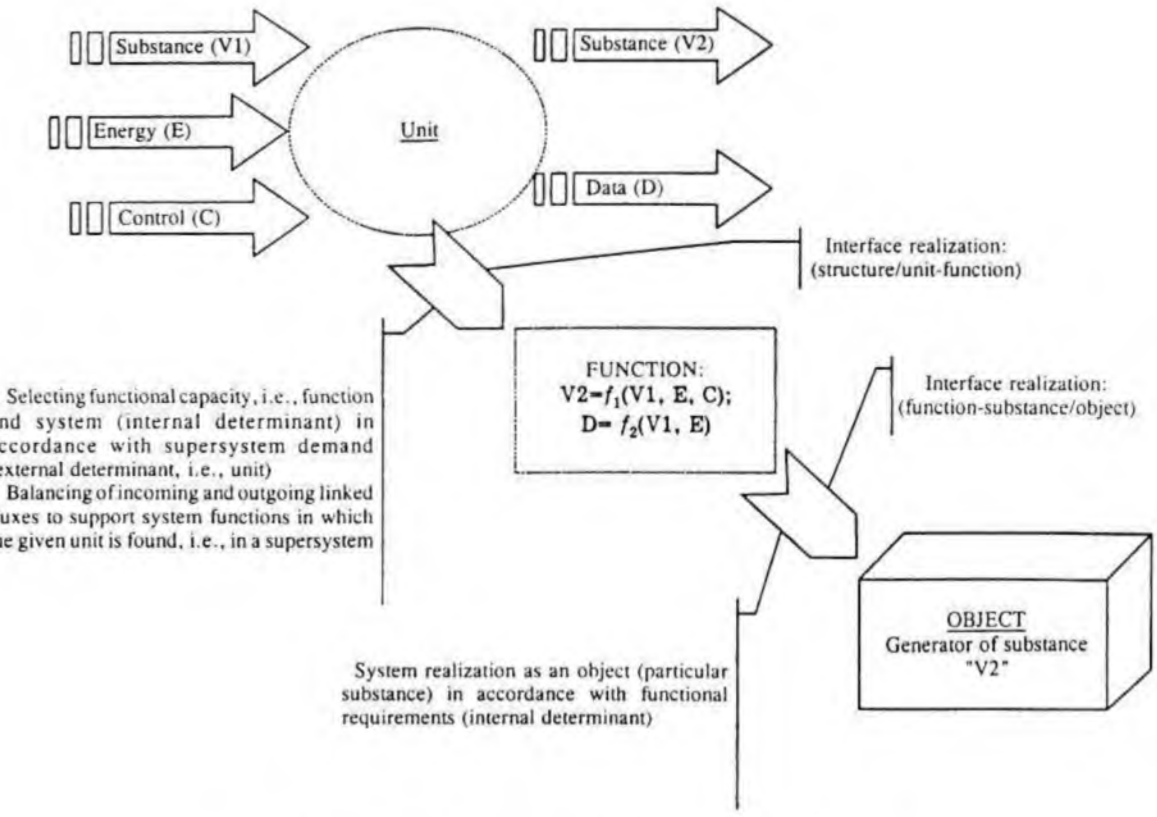
In fact, in most cases one observes distorted forms of regular configurations (for example, it is difficult to identify an organization that works in accordance with all the regulatory documents), so these are called deformed configurations. The deformation mechanism is governed by the deformations of the configurations and the generators. Configuration deformation is its transformation involving violation of the similarity because the attachment rule is violated. Generator deformation is a transformation of it with the violation of similarity because the balance and realization rules are violated.

Set R and similarity transformation f together with the attachment and annihilation operators define the algebra on the configuration space, which we propose to call the UFO algebra, which can be used to formalize the construction of object models for systems as well as for optimizing or adapting them.

To construct a context model for the system that describes the specifications for it, one uses a representation of that system as a high-level generator (compound UFO element as a combination of alphabetic ones). The model decomposition used in examining the system or in designing a new one is described by means of the attachment operator as the construction of a configuration with given exit links. The resulting model for the existing or proposed system can be upgraded by using a similarity transformation (generator adaptation). The model for an existing system can also be upgraded by using the annihilation operator, e.g., by annihilating internal business processes that do not increase the user value of the product.

The basic hierarchy thus allows one to consider any system or subject area as a set of interacting UFO elements, since any phenomenon occurring in reality is a structured part of some larger whole (interacts with other phenomena), and which functions in a definite fashion and at the same time is some material formation. This representation can be realized in the unit-function-object approach (Fig. 5), which provides for structural, functional, and object simulation simultaneously [6].

The resulting normative system is described by an adaptive versatile and dynamic alphabet having content semantics, which enables one to implement the new analysis and simulation method in accordance with the



Substance (V1)
Energy (E)
Control (C)
Unit
Substance (V2)
Data (D)
Interface realization: (structure/unit-function)
FUNCTION: $V2=f_1(V1, E, C)$; $D=f_2(V1, E)$
Interface realization: (function-substance/object)
Selecting functional capacity, i.e., function and system (internal determinant) in accordance with supersystem demand (external determinant, i.e., unit)
Balancing of incoming and outgoing linked fluxes to support system functions in which the given unit is found, i.e., in a supersystem
OBJECT
Generator of substance "V2"
System realization as an object (particular substance) in accordance with functional requirements (internal determinant)

Fig. 5. Principle of the unit-function-object approach (UFO),

requirements of OOAD. **UFO ANALYSIS ALGORITHM**

We consider the UFO analysis algorithm of [6].

The algorithm has the following basic steps:

1) identifying units and links from the functional links of the system as a whole;

2) identifying functions performed by the observed units; and

3) identifying the objects corresponding to this functionality.

A specific feature is that the steps can be automated by means of a formal semantic alphabet if suitable ready-made detailed classes are present in the previously prepared unit-function-object hierarchy (UFO hierarchy). The first step may be identified with the analysis stage, the second with the design one, and the third with the implementation.

Figure 6 shows the general form of the UFO analysis algorithm, which indicates that this analysis begins with examining the customer's requirements through a prism based on the initial link classification (Fig. 2), which along with the UFO hierarchy (Fig. 1) should first be adapted for the analysis and simulation, which means that they should include detailed links, units, functions, and objects as used in that subject area. The larger the species of detailed classes previously prepared in these specifications, the more effectively and simply the analysis and design can be performed.

The analysis begins with constructing a context model that shows the links and as largely as possible is denoted by characters from the formal semantic alphabet, i.e., characters for the ready-made detailed link classes. That model corresponds to the system precedent diagram in the UML object simulation language, and also to the SADT or DFD context diagram.

Then one should fill the system units table with the identified links. The rows in that table correspond to input links in the system and the columns to output ones. The table is analyzed iteratively to identify the units in the system structure, which is necessary in order to observe units for which one can identify particular functions that support the balance of incoming and outgoing fluxes at a given unit, as found from among the elements known for

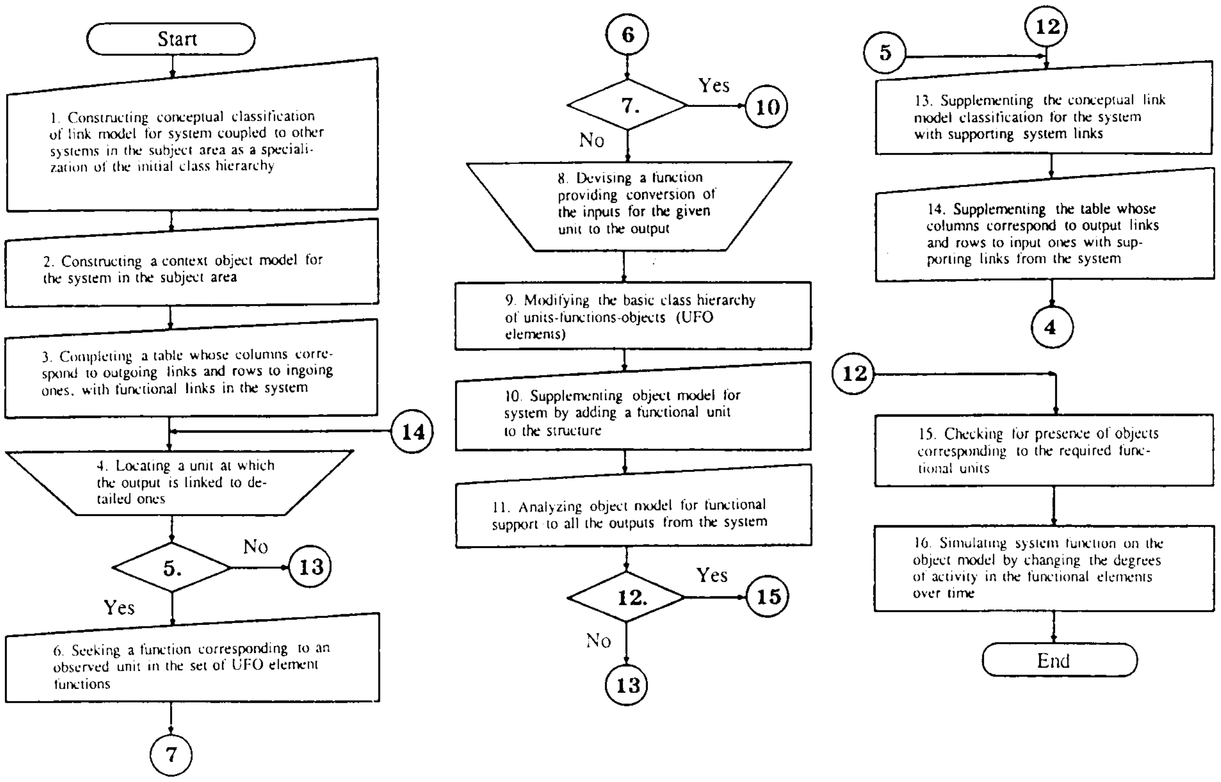

Start
1. Constructing conceptual classification of link model for system coupled to other systems in the subject area as a specialization of the initial class hierarchy
2. Constructing a context object model for the system in the subject area
3. Completing a table whose columns correspond to outgoing links and rows to ingoing ones, with functional links in the system
14
4. Locating a unit at which the output is linked to detailed ones
5.
No
13
Yes
6. Seeking a function corresponding to an observed unit in the set of UFO element functions
7
6
7.
Yes
10
No
8. Devising a function providing conversion of the inputs for the given unit to the output
9. Modifying the basic class hierarchy of units-functions-objects (UFO elements)
10. Supplementing object model for system by adding a functional unit to the structure
11. Analyzing object model for functional support to all the outputs from the system
12.
Yes
15
No
13
5
12
13. Supplementing the conceptual link model classification for the system with supporting system links
14. Supplementing the table whose columns correspond to output links and rows to input ones with supporting links from the system
4
12
15. Checking for presence of objects corresponding to the required functional units
16. Simulating system function on the object model by changing the degrees of activity in the functional elements over time
End

Fig. 6. UFO analysis algorithm.

the given UFO or devised for it. If one cannot observe units with a certain functionality, it is necessary to add new forms of link supporting the functions at increasingly deep levels in the hierarchy. The added links are also written into the table, and the analysis is repeated until the system is represented as a network of flow units for which one can assign known functions or devise new ones.

The UFO analysis algorithm is completed by searching for suitable objects to realize the functions among those prepared as UFO elements or acquired from outside. There is a test for adequacy in the determination of functional objects for identified units in that one has to satisfy the customer on the simulation of the system functioning.

The simulation model based on the object model is set up as a result of the completely obvious and suitable description thus obtained, e.g., about market relationships in terms of systemology concepts.

Recently, there has been a fundamental reconsideration of the purposes and targets in the theory and practice of business activity.

The main purpose of a business now is considered to be not to obtain a profit but instead to satisfy the clients of that business [15]. That concept is really implemented for example by making changes in planning productive activity. At present, in addition to material resource planning MRP and enterprise resource planning, the basis has become customer synchronized resource planning CSRP.

This view on system functions implies that a system should satisfy not its own needs but instead the demand from a higher-level system (supersystem), which is a task for functional systemology, in which one can simulate business processes in the making of products (goods and services), meeting market demands (supersystem), and meeting consumer demands (clients). Suitable representation and analysis of market dynamics can be based on the functional relationship between systems, which occurs in the exchange of elements at deep levels in these systems, which corresponds to the actual processes in which businesses operate on the market.

The clients constitute a system forming the market environment conditions for the business. The clients' needs for the products are needs for a functional link to the business. To satisfy a client of the business system, the latter provides a link between the business processes and the client in the form of an exchange flow. It provides the client with his product in exchange for information (his needs plus money signs). To provide a client with something (support the connection), the business system must contain what is needed to satisfy the demand. This creates a situation in which the business system has an element at a certain depth in the hierarchy for which it is necessary to create conditions necessary for such transfer. Consequently, the business operates to satisfy client demand by adaptation of it to the demand from the supersystem (obtained through the client).

In systemology terms, this market adaptation process for a business system can be described as follows:

1) in accordance with client demand (external determinant), the business system provides appropriate functions (internal determinant);

2) in accordance with the internal determination, the business system transmits the request from the supersystem (client) to one of its subsystems, and so on; and

3) from some potentially suitable initial material, the subsystem generates (obtains, collects, produces, creates, and so on) an elementary system, i.e., a substance for that business system suitable for exchange on the functional link to the client.

The essence of business processes from the systemology viewpoint is that there may be some disruption of the support to functional equilibrium between the business system and the market (as a supersystem) in conjunction with the potential capacity of that system to restore the equilibrium and actually to do so.

Business processes can be simulated, i.e., the operations of the business system to satisfy client demand, by visualizing deviations from the rules of system decomposition in the model, which must be eliminated over a certain period. These deviations should be tracked down to a particular depth in the system at which an elementary system is established or arises to provide the substance participating in the exchange.

The object model is transformed to the simulation one as follows.

First, one constructs a model adequately reflecting the internal structure of the system and the logical structure of the phenomena in it. In the present case, this is done by means of an adaptive UFO element alphabet.

Second, each UFO element from the alphabet is put into correspondence with a certain time parameter, which allows one to divide all the UFO elements in the object model into three categories:

1) passive ones, which do not operate at the given instant;

2) active ones, which operate at that moment; and

3) functional ones, which are in a state of elevated preparedness, i.e., compound objects in which some of the ingoing links have already shown activity and some have not as yet.

Third, the initial conditions for the simulation are defined, i.e., one specifies the links active at the initial instant.

Fourth, the time is reckoned on some scale and observations are made on the state changes (activity, passivity, readiness) for objects in the model up to some preset instant or up to some state arising.

One can track the activity of the objects and links and determine for example the effort involved in a given business process. If in addition to the time parameter, the alphabetic UFO elements are put into correspondence with a cost parameter, one can determine the cost of the business process, i.e., in essence perform a function-cost analysis.

Also, any parameters can be assigned to UFO elements, including formal functional relationships between the input and output parameters of the links, for example, by means of a script mechanism. Such a model allows one to simulate not only the consumption of time and money but also the obtaining of an appropriate quantitative result from the business process.

The UFO hierarchy adaptation noted above for the initial steps in the algorithm consists in specializing it as regards particular units, functions, and objects in the subject area, i.e., it produces an ontology model. The UFO element specifications in that case will contain the following information:

1. For the object (substance parameters): engineering and working characteristics (design, climatic and mechanical working conditions, reliability, necessary and available stocks of energy, materials, and information, as well as productivity and so on); and also the cost and working time. Here it may be useful to rank objects on various features.

2. For a function (process parameters): description of the input to output conversion, i.e., the functional protocol; formal description of the functional dependence if it exists and is necessary in the form of a script or macro. In essence, this represents data on the internal determinant of the corresponding system.

3. For a unit (structure parameters): qualitative (quantitative) characteristics of the flows (including carrying capacity); and cost and time characteristics. In essence, this represents data about different cases in the use of the object, i.e., on the external determinant of the corresponding system.

The analysis and design involve detailed units, functions, and objects for which that information is available, and if so, one can recognize the units and automatically determine the functions and objects for them if one uses the characters in the formal semantic alphabet. If the alphabetic elements are software objects realized in the form of ready-made classes, one can say that UFO analysis is a component of the appropriate technologies under CORBA (Business Object Facility BOF). In the latter case, the CASE software facility that automates the UFO analysis may function within the framework of the business object component architecture BOCA as a framework, which works as a tool for linking business objects into a system and provides a form of convenient working points for carrying out the tasks imposed on it [16]. If on the other hand technical objects are considered as alphabetic elements, then the UFO analysis will be matched to the CALS technology.

CONCLUSIONS

The basic class hierarchy has been used in proposing an alphabet that can be used to construct object models for example for organizations. That alphabet together with the system decomposition rules can constitute a formal semantic normative system, whose features are as follows:

1) it objectivizes object and class decomposition for the system;
2) it provides a range of functional objects (alphabetic symbols) for each form of system in accordance with a unified principle; and
3) it provides for simulating by computer the properties of the classes and the examples of object models.

That normative system allows one to formulate the UFO analysis algorithm for a system analysis method in which the procedures and results for the first time agree with the OOAD requirements. The algorithm in turn can produce a new generation of CASE tools because it provides for transforming such tools into software systems based on knowledge.

REFERENCES

1. G. Buch, Object-Oriented Analysis and Design With Examples of Applications in C++ [Russian translation], 2nd edition, Izd. Binom, Moscow; Nevskii Dialekt, St. Petersburg, 1998.
2. E. Jordan and K. Argila, Structural Models in Object-Oriented Analysis and Design [Russian translation], LORI, Moscow, 1999.
3. S. I. Matorin, "Systemology and the object-oriented approach: Formalization problems and interfacing prospects", Nauch. Tekh. Inform., Ser. 2, no. 8, pp. 1-8, 2001.
4. M. F. Bondarenko, S. I. Matorin, and E. A. Solovyova, "Analysis of systemological tools for conceptual modeling of application fields", Automatic Document and Mathematical Linguistics, Allerton Press Inc., New York, vol. 30, no. 2, pp. 33-45, 1997.
5. S. I. Matorin, "A new method of systemological analysis matched to a procedure for object-oriented design. Part 1", in: Cybernetics and Systems Analysis [in Russian], no. 4, pp. 119-132, 2001.
6. S. I. Matorin, "A new method of systemological analysis matched to a procedure for object-oriented design, Part 2", in: Cybernetics and Systems Analysis [in Russian], no. 1, pp. 118-130, 2002.
7. G. P. Mel'nikov, Systemology and Linguistic Aspects of Cybernetics [in Russian], Sov. Radio, Moscow, 1978.
8. S. I. Matorin, "Systemological research or the structure of system categories", Automatic Document and Mathematical Linguistics, vol. 31, no. 2, Allerton Press Inc., New York, pp. 4-9, 1998.
9.1. Ya. Chupakhin, Methodological Problems in Concept Theory [inRussian], Izd. LGU, Leningrad, 1973.
10. E. A. Solov'eva, Natural Classification: Systemological Principles [in Russian], KhTURE, Kharkov, 1999.
11. D. B. El'chaninov and S. I. Matorin, "Formalizing systemological concepts by means of pattern theory techniques", Iskusstvennyi Intellekt, no. 2, 2002.
12. W. Grenander, Lectures on Pattern Theory. Part 1. Pattern Synthesis [Russian translation], Mir, Moscow, 1979.
13. W. Grenander, Lectures on Pattern Theory. Part 2. Pattern Analysis [Russian translation], Mir, Moscow, 1981.
14. Yu. M. Kanygin and G. I. Kalitich, Principles of Theoretical Informatics [in Russian], Nauk. Dumka, Kiev, 1990.
15. M. Hammer and J. Champy, Corporation Reengineering: A Manifest for the Revolution in Business [Russian translation], Finansy i Statistika, Moscow, 1997.
16. M. Anshina, Business Object Biography, http://www.tops.ru\publishing\pub_021.html