

Information Modeling of Business Processes Using Formal-Semantic Alphabet

S. I. Matorin^{a, *}, A. G. Zhikharev^{a, **}, and Yu. V. Dmitrieva^{a, ***}

^a *Belgorod State National Research University, Belgorod, Russia*

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

***e-mail: zhikharev@bsu.edu.ru*

****e-mail: uka506@mail.ru*

Received September 22, 2024

Abstract—Modern methods of information modeling of organizational-business and production-technological processes are considered. It is shown that the main problem is the lack of subject content in the symbols that are used in graphic notations. A method for constructing a formal-semantic alphabet of processes (functional units) is proposed, the elements of which, first, satisfy all of the requirements for formal systems and, second, have subject-oriented content as intersections of material, energy and information links. It is shown that the set of elements of the proposed alphabet is an algebraic group and also corresponds to the definition of a generator, as the main initial concept of pattern theory. At the same time, the example of a specific generator as a universal operator noted in the theory is clarified. The correspondence between some concepts and procedures of pattern theory and the concepts and procedures of the information modeling of processes is analyzed. This allowed to substantiate the possibility and expediency of using the formal apparatus of pattern theory (image algebra) for the formalization of information-modeling procedures.

Keywords: information graphoanalytical modeling of business processes, organizational and business and production and technological processes, system-object approach, pattern theory

DOI: 10.3103/S0005105524700444

INTRODUCTION

The information modeling of organizational-business and production-technological processes (business processes) is widely used to analyze various types of activities. When constructing information models of such processes, the graphical notations DFD, IDEF0, IDEF3, UML, and BPMN are used, in which the models provide an understanding of how the original resources are transformed into the final product.

The mentioned notations and the software tools that support them have proven their effectiveness in formulating requirements for software systems as well as in reengineering business processes [1–5]. However, their use is a complex, very labor-intensive, and time-consuming creative procedure. In addition, the information modeling of organizational-business and production-technological processes forms a subjective procedure carried out under the slogan “I am an artist, this is how I see it.” Its means do not correspond to the recommendations of developers and users of CASE tools, who note that the least harm to the organization is caused by modeling tools that deprive the developer of that part of the “creative” capabilities leading to the diversity of the presentation of organizational models [6].

This situation is due to the fact that the symbols in all graphic notations are abstract signs in no way related to the subject area (“functional block,” “flow,” etc.). The symbols acquire subject content in the course of constructing a model by an analyst using his knowledge of the subject area and understanding of the processes taking place. Thus, the mentioned notations are considered as formal normative systems that use signs/symbols completely formally without any meaningful interpretation that they may receive following their application to any subject area [7, 8].

Thus, it seems relevant to transform the currently formal means of modeling processes into more meaningful ones. This can be achieved through the assignment of subject content to the symbols of graphic notations, i.e., by creating a formal-semantic alphabet of processes, the elements of which, first, satisfy all of the requirements that are imposed on formal systems (explicit and strict description of all the properties and relations of all the symbols used by means of the formal system itself, as well as recognition of all symbols only by their form [8]) and, second, have subject-oriented content.

The present article is devoted to solving this problem.

Table 1. Classes of processes as functional units (alphabet)

		Outputs of functional units											
		V!	VE!	VI!	VEI!	E!	EV!	EI!	EVI!	I!	IV!	IE!	IVE!
Inputs of functional units	V?	+	+	+	+								
	VE?	+	+	+	+								
	VI?	+	+	+	+								
	VEI?	+	+	+	+								
	E?					+		+					
	EV?						+		+				
	EI?					+		+					
	EVI?						+		+				
	I?									+			
	IV?										+		+
	IE?											+	
	IVE?											+	+

FORMAL-SEMANTIC ALPHABET OF PROCESSES

The science of sign systems uncompromisingly asserts that any sign represents a unity of the signifier and the signified and of form and content [9]. Consequently, the signs of the alphabet of a formal system, in fact, cannot but have meaning in themselves, i.e., a meaningful interpretation before they are included in any expression and its interpretation. Otherwise, these signs simply would not be signs at all! For greater clarity, it is necessary to recognize the fact that in any formal system there are signs either related to abstract concepts of a very large volume (variables, quantifiers, functional blocks, etc.), or denoting specific mathematical or logical operations (differentiation, conjunction, unification, etc.). At the same time, the use of signs in any formal system always presupposes the knowledge and use of their meaning. For example, it is quite obvious that the expressions $A \wedge B$ and $A \vee B$ differ from each other not only in the form of the sign standing between A and B, but also, first of all, in the meaning behind this form. Thus, despite the denial of this fact in some works [10], the signs of a formal system still have a certain content.

The problem that complicates the description of specific system properties and relations by formal means due to their deep and diverse substantive nature is the absence of subject content in the signs of a formal system (including the symbols of graphic notations as formal normative systems).

However, this problem can be overcome by using an alphabet, the signs of which are deliberately assigned not only an abstract, but also a certain subject-conceptual meaning according to some pre-agreed rule.

The symbols of graphic notation (functional sides or processes and flows/connections) can be given a subject meaning before building any model because all

connections (flows) between processes belong to one of three non-intersecting classes: substance (V), energy (E), and information (I), i.e., the connections between organizational-business and production-technological processes (business processes) are always substantive rather than abstract. The division of connections into material and informational ones and material connections into real and energy ones was first proposed for modeling processes within the framework of the system-object approach [11, 12].

There are the following limitations on the transformation of input flow processes into output ones:

- (1) the substance flow V cannot be transformed without a remainder either into the energy flow E or into the information flow I, i.e., if the substance flow is at the input (V?), then the substance flow will also be at the output (V!);
- (2) the energy flow E cannot be transformed either into the substance flow V or into the information flow I;
- (3) the information flow I cannot be transformed either into the substance flow V or into the energy flow E.

These circumstances determine the existence of a very limited set of process classes as the alphabetical functional units presented in Table 1, where the functional units that provide the processes of transformation of substance, energy, and information are presented by their input (?) and output (!) connections. Of all of the connections, one (the first) denotes the flow transformed in the node for the transformation of which this node is intended. The remaining connections are in support of this.

The functional units in Table 1 can be meaningfully interpreted through the corresponding real production processes, for example:

V?/V!E!: transformation of substance into substance with the production of energy (exothermic pro-

Table 2. Rules for joining functional units

		Inputs of functional units											
		V?	E?	I?	VE?	EV?	VI?	IV?	EI?	IE?	VEI?	EVI?	IVE?
Outputs of functional units	V!	+			+	+	+	+			+	+	+
	E!		+		+	+			+	+	+	+	+
	I!			+			+	+	+	+	+	+	+
	VE!				+	o					+	o	+
	EV!				o	+					o	+	o
	VI!						+	o			+	+	o
	IV!						o	+			o	o	+
	EI!								+	o	+	+	o
	IE!								o	+	o	o	+
	VEI!										+	o	o
	EVI!										o	+	o
	IVE!										o	o	+

cess of product release that is accompanied by the release of energy: ammonia production);

V?E?I?/V!I!: transformation of substance into substance using energy and information and the production of information (endothermic process of product release with the control of input and output parameters: smelting of a certain type of metal);

E?/E!I!: energy conversion into energy with the receipt of information (energy production from other types of energy with control of output parameters: the operation of a solar or tidal power plant with the control of electric power parameters);

V?E?I?/V!: substance conversion into substance using energy and information (endothermic process of product release with control of input parameters: the manufacture of metal products by casting into molds);

I?E?/I!E!: information conversion into information using energy and obtaining energy (obtaining information by replacing one electronic storage medium with another: downloading files from the internet to a computer);

I?V?E?/I!V!: information conversion into information using substance and energy and obtaining substance (obtaining information by replacing an electronic storage medium with a material one: printing documents).

The contents of all alphabetical functional units are presented in work [13].

USE OF FORMAL-SEMANTIC ALPHABET OF PROCESSES

Taking into account the information presented in Table 1, we can formulate the rules for joining functional units (elements of the alphabet) to each other (Table 2).

The connection options for nodes shown in Table 2 marked with “+” as well as the processes denoted by these nodes can be meaningfully interpreted, for example:

V?/VE! * VEI?/VI!: conversion of atmospheric nitrogen into ammonia with the release of energy and the subsequent use of ammonia and the energy obtained during its production for the production of hydrogen with control of the energy consumed and the amount of hydrogen obtained;

E?/EI! * VEI?/V!: production of electricity with control of its parameters and its subsequent use for the production of aluminum from alumina;

IE?/IE! * IVE?/IV!: downloading files from the Internet to a computer and then printing these files on a printer.

The interpretation of the connection options for nodes marked in Table 2 as “o” is difficult, but within the framework of the formal regulatory system, such connections have a right to exist. Perhaps their meaningful interpretation will be obtained in the course of further research.

The set of functional units (FUs), i.e., the alphabet and the rules of its use can be described in terms of U. Grenander’s pattern theory [14]. The set of FUs consists of disjoint classes of functional units (processes). Each functional unit is an object with connections (potentially possible). The connections are characterized by the indices β , which, in our case, correspond to three types of flows (substance V, energy E, and information I). In this case, the connections/flows can be input (?) and output (!), i.e. $\beta_g = \beta_g? \cup \beta_g!$. Thus, each functional unit or process can be considered as a *generator* g (in accordance with the original concept of U. Grenander’s pattern theory), and the entire set of FUs can be considered as a set of *specific generators* $g \in FU$ defined on a certain *support space* (a set of specific

intersecting connections/flows: V, E, and I) [14]. Generators can be connected to each other in accordance with the indicators of connections. In this case, the same-name flows/connections are connected.

The proposed understanding is a development of the approach presented in [15, 16] and consists in the fact that nodes, as abstract intersections of connections, were considered in these publications as generators, while in the present work generators are defined on the support space as nodes that transform specific material, energy, and information flows, i.e., as specific processes or operators. The expediency of considering this type of generators is mentioned in [14]: “Case 1.3.18 (universal operators). Every generator is an operator with inputs x_1, x_2, \dots and outputs y_1, y_2, \dots ” [14, p. 22]. However, by contrast with our approach, Grenander does not specify the subject area of the operators’ functioning.

In addition, the set of FUs (specific generators) presented in Table 1 can be considered as an *algebraic group* (in contrast to the abstract generators in [15, 16]), as the binary operation $*$: $FU * FU \rightarrow FU$ is defined on it and the following axioms are satisfied:

(1) associativity,

for example: $(VE^?/VI! * VI^?/V!) * V^?/VEI! = VE^?/VI! * (VI^?/V! * V^?/VEI!)$;

(2) the possibility of using a neutral element (0/0) in the form of a node that does not transform any flows (i.e., a node without input and output),

for example: $0/0 * VE^?/VI! = VE^?/VI!$ or $VE^?/VI! * 0/0 = VE^?/VI!$, i.e., the neutral element does not change the transformation;

(3) the presence of an inverse element,

for example: $V^?/VI! * VI^?/V! = VI^?/V! * V^?/VI! = 0/0$, i.e., there was no transformation.

Pattern theory provides for the unification of generators into *configurations* in accordance with a certain system of rules or restrictions that determine which configurations must be considered regular [14]. In our case, the rules for connecting the links of functional units from Table 1 for constructing regular configurations (rk) are defined in Table 2, which shows how the output links of units (generators of $g \in FU$) can be connected to the input links. According to the theory [14], a configuration is characterized by the composition of generators ($rk = g_1, g_2, \dots, g_i, \dots, g_n$) and structure, i.e. the type of connection of generators. In this case, it is a linear order that forms concatenation chains having the property of associativity, but not commutativity.

Equivalence classes that are defined on a set of configurations containing information about unconnected (external: $ext\ rk = ext^? rk \cup ext! rk$) links of a configuration are considered images in pattern theory (iz) [14], i.e., unconnected links of a certain configuration correspond to input and output connections of a certain image: $ext^? rk = \beta_{iz}^?$ and $ext! rk = \beta_{iz}!$.

In this case, the following aspect of Grenander’s theory which we will use further is important for us: “it will sometimes be quite natural to consider objects that are images at a certain level of formal description as generators in a higher-level formalism” [14, p. 15].

In terms of pattern theory, we can formulate a certain statement that ensures the use of the proposed formal-semantic alphabet for modeling processes.

Statement. $\forall iz \in \Phi Y \exists rk(g_1, g_2, \dots, g_i, \dots, g_n): g_i \in \Phi Y$, i.e., for any image belonging to a set of functional units, there exists a configuration of generators that belong to the same set.

First, any $iz \in FU$ can always be associated with one generator $g_i \in FU$ (see Table 1), for which $\beta_{g_i}^? = \beta_{iz}^?$ and $\beta_{g_i}! = \beta_{iz}!$, i.e., they have the same input and output connections. One isolated generator is considered in pattern theory [14] as a configuration, the type of connection, of which is called *monoatomic*. Thus, we always have a configuration corresponding to the image ($ext^? rk(g_i) = \beta_{iz}^?$ and $ext! rk(g_i) = \beta_{iz}!$), and the above statement is formally true in this case, but it is completely meaningless in terms of content.

Second, any $iz \in FU$ can always be associated with the two generators $g_i \in FU$ and $g_j \in FU$ (see Table 1), for which $\beta_{g_i}^? = \beta_{iz}^?$, $\beta_{g_i}! = \beta_{iz}!$ and $\beta_{g_j}! = \beta_{g_i}^?$. This is ensured by the symmetry characteristic of an algebraic group. Thus, we will always obtain a configuration corresponding to the image ($ext^? rk(g_i * g_j) = \beta_{iz}^?$ and $ext! rk(g_i * g_j) = \beta_{iz}!$), and the above statement is, in this case, also formally correct and has an informal meaning.

Third, if a configuration of two generators does not address the problem, then arbitrary $g_i \in FU$ or their chains can be connected to the inputs and outputs of the image iz (or only to the inputs or only to the outputs). As a result, we will obtain (as if “internal” with respect to the original image iz) the image $iz' \in FU$. This always allows us to find either one or two generators that close the chain and form a configuration corresponding to the original image with a given number of generators.

The presence of a larger number of variants for connecting generators (functional units) in Table 2 increases the possibility of creating a configuration corresponding to the image.

CONCLUSIONS

Information graph-analytical modeling begins with the construction of a context model of the process, which only presents its external interactions (functional connections). Then the process is decomposed into subprocesses, the interaction between which provides external functional connections of the context process.

A context model of a process can be considered in terms of pattern theory as an image, and decomposition of the process in the same terms can be considered

an analysis of the image [14] by finding a configuration corresponding to the image. The configuration consists of generators, which in our case are functional units (processes) of the formal semantic alphabet.

Using the methods of pattern theory, this article shows that use of the proposed formal semantic alphabet for any image makes it possible to construct a corresponding configuration. This means that, for any context model of a process, it is possible to construct a decomposition diagram of this process, if we use the proposed formal semantic alphabet.

The above reasoning confirms the possibility and expediency of using the formal apparatus of pattern theory (image algebra) for formalizing procedures for information modeling of business processes, at least by means of the system-object approach, as it is within the framework of this approach that the classification of connections/flows is used and the idea of creating a formal-semantic alphabet was formed [11, 12].

The results of this study allow for the use of the proposed formal-semantic alphabet as a constructor when building models of organizational-business and production-technological processes. At the same time, the process of construction itself is a formalized procedure, which allows for the possibility of developing a clear algorithm for its implementation and, consequently, creating software tools that allow building process models in an automated (and therefore less subjective) way compared to existing ones.

FUNDING

This work was supported by ongoing institutional funding. No additional grants to carry out or direct this particular research were obtained.

CONFLICT OF INTEREST

The authors of this work declare that they have no conflicts of interest.

REFERENCES

1. Elizarova, N.N. and Arkhangel'skaya, E.L., Application of graph-analytical method of analyzing the subject area in the design of information systems, *Vestnik Ivanovskogo Gosudarstvennogo Energeticheskogo Universiteta*, 2010, no. 4, pp. 1–5.
2. Kaziev, V.M., *Vvedenie v sintez i modelirovanie sistem* (Introduction to Analysis, Synthesis and Modeling of Systems), Moscow: Prosveshchenie/Binom, 2016.
3. Kachala, V.V., *Obshchaya teoriya sistem i sistemnyi analiz* (General Systems Theory and Systems Analysis), Moscow: Goryachaya Liniya-Telekom, 2017.
4. Maklakov, S.V., *Modelirovanie biznes-protsessov s ALL-Fusion Process Modeler (BPwin 4.1)* (Business Process Modeling with ALLFusion Process Modeler (BPwin 4.1)), Moscow: Dialog-MIFI, 2004.
5. Dubeikovskii, V.I., *Effektivnoe modelirovanie s AllFusion Process Modeler 4.1.4 i AllFusion PM* (Efficient Modeling with AllFusion Process Modeler 4.1.4 and AllFusion PM), Moscow: Dialog-MIFI, 2004.
6. Rubtsov, S., Which CASE tool will do the least harm to the organization?. <https://www.lobanov-logist.ru/library/362/54334/>. Cited September 19, 2024.
7. Maslov, S.Yu., *Teoriya deduktivnykh sistem i ee primeneniye* (Deductive Systems Theory and Its Applications), Moscow: Radio i Svyaz', 1986.
8. Petrov, Yu.A., *Metodologicheskie voprosy analiza nauchnogo znaniya* (Methodological Issues in Analyzing Scientific Knowledge), Moscow: Vysshaya Shkola, 1977.
9. Stepanov, Yu.S., *Semiotika* (Semiotics), Stepanov, Yu.S., Ed., Moscow: Raduga, 1983.
10. Hilbert, D. and Bernays, P., *Grundlagen der Mathematik*, Berlin: Springer, 1968.
11. Matorin, S.I., Popov, A.S., and Matorin, V.S., Organization simulation technology in the light of a new unit-function-object approach, *Autom. Doc. Math. Linguist.*, 2005, vol. 39, no. 1, p. 1.
12. Matorin, S.I., Zhikharev, A.G., Zimovets, O.A., Tubol'tsev, M.F., and Kondratenko, A.A., *Teoriya sistem i sistemnyi analiz. Uchebnik* (Systems Theory and Systems Analysis: A Textbook), Matorin, S.I., Ed., Moscow: Knorus, 2021.
13. Matorin, S.I., Pesotsky, S.A., Zhikharev, A.G., and Dmitrieva, Yu.V., An improved alphabet for graphic-analytical modeling of processes using system-object analysis, *Iskusstvennyi Intellekt i Prinyatie Reshenii*, 2024, no. 2, pp. 62–75. <https://doi.org/10.14357/20718594240205>
14. Grenander, U., *Pattern Synthesis: Lectures in Pattern Theory*, Applied Mathematical Sciences, vol. 18, New York: Springer, 1976. <https://doi.org/10.1007/978-1-4612-6369-2>
15. Matorin, S.I. and El'chaninov, D.B., Application of pattern theory in formalization of systemological UFO analysis, *Nauchn.-Tekhn. Inform., Ser. 2. Protsessy Sist.*, 2002, no. 11, pp. 1–11.
16. Matorin, S., El'chaninov, D., Zin'kov, S., and Matorin, V., Systems synthesis and analysis in the light of the node-function-object approach, *Nauchn.-Tekhn. Inform., Ser. 2. Protsessy Sist.*, 2006, no. 8, pp. 10–16.

Translated by L. Solovyova

Publisher's Note. Allerton Press remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

AI tools may have been used in the translation or editing of this article.