

# Integration of Formalization Tools for Graphical–Analytical “Unit–Function–Object” Models

O. A. Zimovets and S. I. Matorin

E-mail: ozimovets@bsu.edu.ru; matorin@bsu.edu.ru.

**Abstract**—A new method for the formal description of systems in terms of “Unit”, “Function”, and “Object” based on the comparative study and integration of tools from the Grenander algebraic theory of patterns and Milner’s process calculus is considered. Decomposition and aggregation procedures are formalized for the graphic–analytical models “Unit–Function–Object” using the proposed approach.

**Keywords:** “Unit–Function–Object” approach, theory of patterns, process calculus, formalization of graphic elements, interface decomposition, operations on functions

**DOI:** 10.3103/S0147688213060099

## INTRODUCTION

The “Unit–Function–Object” (UFO) system–object graphical–analytical approach is one of many modeling and analytical tools for business systems and processes [1]. The idea of the UFO-approach is to consider any given system (including business systems and processes) from three perspectives. On one hand, the system is an intersection of input and output links/flows, i.e., a *Unit* (knot). On the other hand, it is the process of converting the elements that are received via input flows into elements released via output flows, i.e., a *Function*. Finally, it is a material phenomenon that implements the function of converting inputs into outputs, i.e., an *Object*. Integration of these three aspects makes it possible to represent any business system as a “Unit-Function-Object” element or UFO-element that formalizes three obvious facts. First, any given system is a part of the structure (a knot) of a higher-level system (supersystem). Second, functioning (conversion of inputs into outputs) is a necessary aspect of the system. Third, any system (that functions and is a part of structure) does exist as a material phenomenon (personnel, premises, equipment, documents, etc.).

The UFO-technology for visual graphical–analytical modeling and analysis of complex (first of all, organizational) systems is developed based on the UFO-approach. The system is implemented as a dedicated CASE-based UFOtoolkit [2]. Systems are analyzed using UFO tools on the basis of computer graphic UFO-models, which represent systems in terms of “Units”, “Functions”, and “Objects”. The UFO-technology has already been successfully applied, for example, for the design of the maintenance system for TV and radio broadcasting networks [3, 4] and the design of an outdoor lighting control system [5].

Field experience from applying the UFO-approach and technology was a convincing argument for the necessity and possibility of formalization of their key propositions aimed at higher efficiency and effectiveness. Grenander’s theory of patterns [6] as well as pi-calculus and Milner’s process calculus [7, 8] were used in recent attempts at such formalization. The analysis of the formalization results showed that it is pertinent to integrate Grenander’s and Milner’s algebraic tools for higher formalization of the UFO-approach and UFO-models.

## 1. A COMPARATIVE STUDY OF THE THEORY OF PATTERNS AND PROCESS CALCULUS

To perform integration of the theory of patterns (PT) and the calculus of communication systems (CCS), several terms from these approaches should be compared. The similarity and close relationship between the algebraic tools of the PT and CCS were noted and documented in [8]. This fact is considered below in more detail based on [9, 10]. The main connection between the PT and CCS may be seen in description of the term “configuration” [9] on one hand and the term “process” [10] on the other. Both these terms are defined by a certain graph characterized by its composition and structure (Table 1).

The analysis of papers [9, 10] and the presented table allow us to make conclusions that will also be presented in table form (Tables 2 and 3).

It may be stated on the basis of Table 1 that the CCS possesses a more abstract algebraic apparatus in comparison with the PT, and the set of configurations may be considered as a subset of the process graphs:  $(\mathbf{G}, \sigma) \subset (\mathbf{S}, \mathbf{R})$ .

In addition, the CCS, when taken separately (without using PT tools), is not able to provide correct for-

**Table 1.** Comparison of the pattern-theory approach and process calculus

Pattern theory (PT)	Calculus of communication systems (CCS)
<p><i>Configuration</i> <math>\mathbf{c} = (\mathbf{G}, \sigma)</math>,                      where <math>\mathbf{G}</math> is the set of generators (vertices of the configuration graph);  <math>\sigma</math> is the set of <i>connections</i> of generator links (that determine the structure of the configuration graph)</p>	<p><i>Process</i> <math>\mathbf{P} = (\mathbf{S}, \mathbf{R})</math>,                      where <math>\mathbf{S}</math> is the set of states of the process (vertices of the process graph);  <math>\mathbf{R}</math> is the set of <i>transitions</i> between the states of the process (which determine the structure of the process graph);</p>
<p><i>The generator</i> <math>\mathbf{g}</math> (the set of which forms the set <math>\mathbf{G}</math>, which consists of nonoverlapping classes) is a labeled object with links, which is characterized by attribute <math>\alpha</math> and parameters of input and output links <math>\beta</math>. It is considered as a graphic formalism.                      The <i>similarity transformation</i> <math>\mathbf{S}</math> is the mapping of the set <math>\mathbf{G}</math> onto itself, which does not extend the generator beyond its class;</p>	<p><i>The initial state</i> <math>\mathbf{s}^0 \in \mathbf{S}</math> is distinguished from the set of <i>states</i> <math>\mathbf{S}</math>;</p>
<p>The <i>type of connection</i> <math>\Sigma</math> is the set of all allowed sets of links <math>\sigma</math>.                      The <i>matching (or connection) relationship</i> <math>\rho</math> is the parameter of the relative correspondence of links (<math>\beta\rho\beta^*</math>).                      The <i>regular</i> or <i>allowed</i> configuration is the configuration, for which (<math>\beta\rho\beta^*</math>) is fulfilled for any connection (<math>\beta, \beta^* \in \sigma \in \Sigma</math>).                      The <i>internal relationships</i> of the configuration are the links involved in connections implied by the structure <math>\sigma</math>.                      The <i>external relationships</i> of the configuration <math>\mathbf{ext}(\mathbf{c})</math> are the links not involved in connections implied by the structure <math>\sigma</math>;</p>	<p><i>The labeled transition system</i> <math>(\mathbf{S}, \mathbf{R})</math> over the <i>set of actions</i> <math>\mathbf{Act}(\mathbf{P})</math>, which is divided into classes called <i>input actions</i> (<math>\alpha?</math>), <i>output actions</i> (<math>\alpha!</math>), and <i>internal actions</i> (<math>\alpha\tau</math>), is specified</p>

**Table 2.** Conclusions on the compositions of the corresponding graphs

Pattern theory (PT)	Calculus of communication systems (CCS)
The generator $\mathbf{g}_i$ is introduced as an independent entity with formally defined properties	The state $\mathbf{s}_i$ is not considered an independent entity and its properties are not defined
The hierarchy of configurations may be created via representation of each generator within the configuration as a lower-level configuration	The hierarchy of processes cannot be created, because the state cannot be represented as a process (and in any other way)

malization of analysis/decomposition (Table 2) and synthesis/aggregation (Table 3) of UFO-elements, and, therefore, UFO-models. It should be noted that graphical–analytical system UFO-models obtained using UFO-approach have a hierarchical nature and thus cannot be constructed and analyzed without using aggregation and decomposition procedures.

On the other hand, restriction to PT tools (rejection of CCS), which provide adequate overall representation of knot and object characteristics of the UFO-element, do not allow users to obtain complete representation of functional (process) characteristics,

which essentially are the internal characteristics of the element (system).

The performed analysis is used as a base for possible solutions of the problem of the integration of the algebraic tools of the PT and CCS for the formalization of the UFO-approach and UFO-models.

## 2. A NEW FORMALIZATION TECHNIQUE FOR THE UFO APPROACH

Consider the formalization of the UFO-element as a graphic representation of a system that borrows

**Table 3.** Conclusions on the structure of the corresponding graphs

Pattern theory (PT)	Calculus of communication systems (CCS)
The configuration graph has both closed internal links and unclosed external ones	The process graph only has closed internal links/transitions and has no unclosed external links
Two or more configurations may be naturally connected via unclosed links, thereby forming more complex configurations	Processes cannot be connected in the same sense, since they only may be united using a specific (artificial) technique via their (initial) states, rather than links/transitions

terms, notations, and symbols from both algebraic approaches. First of all, the UFO model  $M_{UFO}$  should be formally defined. The UFO model is a graph that is characterized by its composition and structure (similarly to a configuration, or a process graph). UFO elements act as the vertices of the graph, and flows (links), which connect the vertices are the edges:  $M_{UFO} = (E, L)$ , where  $E$  is the set of UFO elements, and  $L$  is the set of names of links between UFO elements.

Sets of all UFO-elements  $E$  and links  $L$  consist of non-overlapping classes. This division implies that a class of elements (or class of links) only includes the elements (and links) that belong to the same level (layer) of the hierarchy of the subject area. The lower hierarchy level will be designated as  $E^{-1}$  for UFO elements and  $L^{-1}$  for links. In addition (similarly to [2]), the set  $L$  will be considered at each hierarchy layer, which in turn consists of four nonoverlapping classes  $S, E, C,$  and  $D$  in a way that  $L = S \cup E \cup C \cup D$ . These classes are interpreted as substance links, energy links, control links, and data links, respectively. Moreover, the term “similarity transformation” will be considered for the set  $E$ , and the terms “type of connection”  $\Sigma$  and “matching (or connection) relationship”  $\rho$  will be considered for the set  $L$ , similarly to PT. In the case of business-system or business-process modeling, it is reasonable to consider the term “matching relationship”  $\rho$  as an equality with accuracy up to the opposite sign, because only the connections of equal links, namely one input and one output, are considered in applied business modeling.

The initial point of the formal identification of the system as a UFO-element  $e \in E$  is its representation as a tuple [1, 2]:  $e = \langle U, F, O \rangle$ . Here,  $U$  is a “Unit”, i.e., the set of output and input links that characterize the knot occupied by the identified system;  $F$  is a “Function”, i.e., the class of functions that characterize techniques and processes (procedures) of conversion of the knot’s input links into the input ones;  $O$  is the “Object”, i.e., the set of properties (attributes) that characterize the class of objects that implement the given class of functions.

The “Unit” of a UFO-element is defined using the notation accepted for CCS as follows:  $U = (L^?, L!)$ , where  $L^? \subset L$  is the set of input links, and  $L! \subset L$  is the set of output links of the given unit.

The previously suggested definition, which is similar to that of Process in CCS ([8, 11]), is used for the “Function” of the UFO-element. According to this definition, the “Function” of the UFO-element may be represented as follows:

$$F = (S, S^0, L\tau),$$

where  $S$  is the set of subprocesses of the process, which corresponds to the “Function”, implemented by UFO-elements that belong to the class  $E^{-1}$ ;  $S^0 \subset S$  is the set of input ( $S^?$ ) and output ( $S!$ ) interface subpro-

cesses ( $S^0 = S^? \cup S!$ ; the set of input links of the set  $S^?$  includes the set of links  $L^?$ , and the set of output links of the set  $S!$  includes the set of links  $L!$ ); and  $L\tau \subset L^{-1}$  is the set of links in  $S$  that perform the transition of the elements from the underlying layer of related subprocesses:  $s_i \xrightarrow{L\tau_{ij}} s_j$ . In other words, although it is similar to the CCS, the labeled transition system  $(S, L\tau)$  is considered over the set of flows (links), rather than actions, as in the CCS. The elements of the set of flows  $Act(F)$ , which corresponds to the set of actions in the CCS, are also interpreted as the input, output, or transition of the element (with the name of the flow). Only descriptions of internal flows (transitions of elements) are of interest at the level of “Functions”, because input and output flows are described at the level of “Units”.

The characteristics of the generator as an object with links in PT [9] are used for the definition of the “Object”. Thus, it is possible to define the “Object” of a UFO-element as follows:

$$O = (n, \alpha, \beta?, \beta!), \tag{1}$$

where  $n \in N$  is the name of the “Object” from the set  $N$  of object names;  $\alpha$  is the set of attributes of the “Object”  $n$ ;  $\beta?$  is the set of parameters of the set of input links  $L^?$ ; and  $\beta!$  is the set of parameters of the set of output links  $L!$ .

Thus, the following expression may be composed as a formal definition of a particular system  $e_i$  as a UFO-element:

$$e_i = \langle (L_i^?, L_i!), (S_i, S_i^0, L_i\tau), (n_i, \alpha_i, \beta_i?, \beta_i!) \rangle.$$

To solve applied problems, this definition should be supplemented with matrices, which would specify the structure  $L_i^? \times S_i^?$  of the input links of the UFO-element  $e_i$ , the structure  $L_i! \times S_i!$  of the output links, and the structure  $S_i \times S_i$  of the internal flows  $(S_i, L_i)$  of the Function of the UFO-element  $e_i$ .

### 3. FORMALIZATION OF THE DECOMPOSITION OF A UFO ELEMENT

If a UFO-element (system) is considered as a whole at a certain step of system analysis and its internal functional structure is not taken into account (at the level of context), then the equation in parentheses for the “Function” of the UFO-element will be as follows:  $(\{s_i^0 \in S_i\}, \{s_i^0 \in S_i^0\}, L_i\tau = \emptyset)$ . In other words, the UFO-element with a zero “Function” defined in [8, 11] by analogy with a zero (empty) process in the CCS is considered in this case. Then, the expression for the system as a UFO-element at the given level of consideration (context model) will be as follows:

$$e_i = \langle (L_i^?, L_i!), (\{S_i^0\}), (n_i, \alpha_i, \beta_i?, \beta_i!) \rangle.$$

When applied problems are solved using visual graphical–analytical modeling tools, the internal functional structure of the UFO-element has to be taken into account, i.e., it is necessary to consider UFO-elements not only at the context level, but in terms of decomposition as well. Decomposition of the system (a UFO-element) strictly into interface subsystems (subprocesses) is of special interest. A special role of this decomposition (the term *interface* decomposition is suggested) may be discovered in numerous examples of functional modeling [12]. The formal definition for this decomposition is introduced below.

**Definition.** Decomposition of the system is called interface decomposition, when  $S = S^0$ .

According to this definition, the expression in parentheses for the “Function” of the UFO-element with the internal structure taken into account will be as follows in the case of interface decomposition:  $(S_i^0, S_i^0 L_i \tau_{?i})$ . Then, since  $S_i^0 = S_i^? \cup S_i^!$ , the expression for the system as a UFO-element at the first step of the decomposition will be as follows:

$$e_i = \langle (L_i^?, L_i^!), ((S_i^? \cup S_i^!), (S_i^? \cup S_i^!), L_i \tau_{?i}), (n_i, \alpha_i, \beta_i^?, \beta_i^!) \rangle.$$

The formula is simplified by the removal of the repetitive set of signs and identification of the actual position of the internal flow. As a result, the following expression for the UFO-element is derived at the first step of the interface decomposition of the system:

$$e_i = \langle (L_i^?, L_i^!), (S_i^?, L_i \tau_{?i}, S_i^!), (n_i, \alpha_i, \beta_i^?, \beta_i^!) \rangle.$$

The following proposition is valid for interface decomposition.

**Proposition.** If the internal functional structure of the UFO element at the level of decomposition is characterized by condition  $L_i \tau_{?i} = \{I_i \tau_{?i}\}$  (i.e., is a single-element set), then linear order is its type of connection  $\Sigma$  and interface decomposition takes place.

**Proof.** It follows directly from the validity of the condition  $L_i \tau_{?i} = \{I_i \tau_{?i}\}$  that  $S/S^0 = \emptyset$ . In turn, the latter implies  $S = S^0$ , i.e., interface decomposition takes place according to the definition. If  $L_i \tau_{?i} = \{I_i \tau_{?i}\}$ , then the output link of the first subprocess is connected to the input link of the last process, which corresponds to the type of connection  $\Sigma$  defined in the PT as linear order [9].

When interface decomposition with linear order occurs, the latter expression for the UFO-element at the first step of decomposition of the system will be as follows:

$$e_i = \langle (L_i^?, L_i^!), (\{s_i^?\}, \{I_i \tau_{?i}\}, \{s_i^!\}), (n_i, \alpha_i, \beta_i^?, \beta_i^!) \rangle.$$

Interface decomposition with linear order is used to solve a number of applied problems of functional business modeling. In particular, it may be used for modeling of administrative procedures that correspond to government and municipal services provided

to people in electronic form within the federal E-Russia program. Most graphical–analytical models of administrative procedures considered as UFO-elements are formalized at the context level using the expression

$$e_i = \langle (\{I_i^?\}, \{I_i^!\}), (\{s_i^0\}), (n_i, \alpha_i, \beta_i^?, \beta_i^!) \rangle.$$

and the following expression is used for the first step of decomposition:

$$e_i = \langle (\{I_i^?\}, \{I_i^!\}), (\{s_i^?\}, \{I_i \tau_{?i}\}, \{s_i^!\}), (n_i, \alpha_i, \beta_i^?, \beta_i^!) \rangle.$$

where  $I_i^? \in L_i^?$ ,  $I_i^! \in L_i^!$ ,  $s_i^0 \in S_i^0$ ,  $s_i^? \in S_i^?$ ,  $s_i^! \in S_i^!$ ,  $I_i \tau_{?i} \in L_i \tau_{?i}$

#### 4. FORMALIZATION OF AN AGGREGATION OF UFO ELEMENTS

System analysis using graphical–analytical UFO-models implies not only decomposition, but aggregation of UFO-elements of the model as well. To provide complete formalization of such models, it should be possible to define aggregation on the basis of the proposed algebraic apparatus, similarly to decomposition, as discussed above.

First of all, we consider the approach to formalization of aggregation of systems as UFO-elements using two binary UFO-elements  $e_i$  and  $e_j$  as an example. These elements are represented at the context level using the following expressions:

$$e_i = \langle (\{I_i^?\}, \{I_i^!\}), (\{s_i^0\}), (\beta_i^?, \beta_i^!) \rangle,$$

$$e_j = \langle (\{I_j^?\}, \{I_j^!\}), (\{s_j^0\}), (\beta_j^?, \beta_j^!) \rangle,$$

The parameters  $n$  and  $\alpha$  are insignificant for the solution of the problem and therefore are neglected below for brevity. Aggregation conditions for UFO-elements defined in [2] as *the rules of system decomposition* are specified for the selected variant of formalization of graphical–analytical UFO-models as follows. Two systems  $e_i$  and  $e_j$  represented as UFO-elements may be aggregated into one system (one UFO-element  $e_{ij}$ ), if at least one pair of conditions is fulfilled: first,  $I_i^! = I_j^?$  and  $\beta_i^! \subseteq \beta_j^?$ ; second,  $I_i^? = I_j^!$  and  $\beta_i^? \supseteq \beta_j^!$ . In other words, UFO-elements are aggregated according to the rules of adjunction of image algebra from PT [9].

This level of formalization shows that the correspondence of unit and object characteristics (those of structure and substance, respectively) is a necessary and sufficient condition for the aggregation of systems into a higher-level system (supersystem). In fact, the construction of a system from certain parts does not require any knowledge on their functioning algorithms. The possibility for the connection of these parts at the interface level, i.e., the level of names and characteristics of links, may be sufficient.

However, consistent understanding and analysis of the system that emerges as a result of assembly are

**Table 4.** Operations on processes and functions

Calculus of communication systems (CCS)	UFO-approach
Process: $\mathbf{P} = (\mathbf{S}, \mathbf{s}^0, \mathbf{R})$	Function: $\mathbf{F} = (\mathbf{S}, \mathbf{S}^0, \mathbf{L}\tau)$
Prefix operation: $\alpha.\mathbf{P} = (\mathbf{S} \cup \{\mathbf{s}^0, \notin \mathbf{S}\}, \mathbf{s}^0, \mathbf{R} \cup \{\mathbf{s}^0, \alpha, \mathbf{s}^0\})$ , where $\mathbf{s}^0, \alpha, \mathbf{s}^0$ indicates the link/transition $\alpha$ between states $\mathbf{s}^0$ and $\mathbf{s}^0$	Prefix operation: $\mathbf{s}?.\mathbf{F} = (\mathbf{S} \cup \{\mathbf{s}^? \notin \mathbf{S}\}, \{\mathbf{s}^? \in \mathbf{S}^?\}, \mathbf{L}\tau \cup \{\mathbf{s}^?, \mathbf{l}\tau_{\eta_i}, \{\mathbf{s}_i \in \mathbf{S}\}\})$ Prefix operation: $\mathbf{s}!\mathbf{.F} = (\mathbf{S} \cup \{\mathbf{s}! \notin \mathbf{S}\}, \{\mathbf{s}! \in \mathbf{S}!\}, \mathbf{L}\tau \cup \{\{\mathbf{s}_i \in \mathbf{S}\}, \mathbf{l}\tau_i, \mathbf{s}!\})$
Alternative composition: $\mathbf{P}_1 + \mathbf{P}_2 = (\mathbf{S}_1 \cup \mathbf{S}_2 \cup \{\mathbf{s}^0, \notin \mathbf{S}_1 \cup \mathbf{S}_2\}, \mathbf{s}^0, \mathbf{R}_1 \cup \mathbf{R}_2 \cup \{(\mathbf{s}^0, \alpha, \mathbf{s}_1)   (\mathbf{s}_1, \alpha, \mathbf{s}_1) \in \mathbf{R}_1\} \cup \{(\mathbf{s}^0, \alpha, \mathbf{s}_2)   (\mathbf{s}_2, \alpha, \mathbf{s}_2) \in \mathbf{R}_2\})$	Alternative composition by input: $\mathbf{s}?.(\mathbf{F}_1 + \mathbf{F}_2) = (\mathbf{S}_1 \cup \mathbf{S}_2 \cup \{\mathbf{s}^? \notin \mathbf{S}_1 \cup \mathbf{S}_2\}, \{\mathbf{s}^? \in \mathbf{S}_1^? \cup \mathbf{S}_2^?\}, \mathbf{L}\tau_1 \cup \mathbf{L}\tau_2 \cup \{\mathbf{s}^?, \mathbf{l}\tau_{\eta_1}, \mathbf{s}_1\} \cup \{\mathbf{s}^?, \mathbf{l}\tau_{\eta_2}, \mathbf{s}_2\})$ Alternative composition by output: $\mathbf{s}!\mathbf{.}(\mathbf{F}_1 + \mathbf{F}_2) = (\mathbf{S}_1 \cup \mathbf{S}_2 \cup \{\mathbf{s}! \notin \mathbf{S}_1 \cup \mathbf{S}_2\}, \{\mathbf{s}! \in \mathbf{S}_1! \cup \mathbf{S}_2!\}, \mathbf{L}\tau_1 \cup \mathbf{L}\tau_2 \cup \{\mathbf{s}_1, \mathbf{l}\tau_{\eta_1}, \mathbf{s}!\} \cup \{\mathbf{s}_2, \mathbf{l}\tau_{\eta_2}, \mathbf{s}!\})$

impossible if the functioning of the system as a whole is not taken into account. Thus, a new functionality arises after the aggregation of UFO-elements, whose formal description will be carried out using definitions of operations on functions that were proposed for the first time in [11] by analogy with operations on processes in the CCS. However, the definitions will be specified and extended in the present paper (Table 4).

Due to the introduction of interface decomposition, definitions of operations on functions considered at the context level, i.e., in the case of  $\mathbf{F}_i = (\{\mathbf{s}_i^0 \in \mathbf{S}_i\}, \{\mathbf{s}_i^0 \in \mathbf{S}_i\}, \mathbf{L}\tau = \emptyset) = \mathbf{s}_i^0$  are of special interest. In this case, the definitions presented in Table 4 will be as follows:

Prefix operation:

$$\begin{aligned} \mathbf{s}?.\mathbf{s}_i^0 &= (\{\mathbf{s}_i^0\} \cup \{\mathbf{s}^?\}, \{\mathbf{s}^?\}, \{\mathbf{s}^?, \mathbf{l}\tau_{\eta_i}, \{\mathbf{s}_i^0\}\}) \\ &= (\{\mathbf{s}_i, \mathbf{s}^?\}, \{\mathbf{s}^?, \mathbf{s}_i!\}, \{\mathbf{s}^?, \mathbf{l}\tau_{\eta_i}, \{\mathbf{s}_i!\}\}) \\ &= (\{\mathbf{s}^?\}, \{\mathbf{l}\tau_{\eta_i}\}, \{\mathbf{s}_i^?\}); \end{aligned}$$

Postfix operation:

$$\begin{aligned} \mathbf{s}?.\mathbf{s}_i^0 &= (\{\mathbf{s}_i^0\} \cup \{\mathbf{s}!\}, \{\mathbf{s}!\}, \{\{\mathbf{s}_i^0\}, \mathbf{l}\tau_{\eta_i}, \mathbf{s}!\}) \\ &= (\{\mathbf{s}_i, \mathbf{s}!\}, \{\mathbf{s}_i^?, \mathbf{s}!\}, \{\{\mathbf{s}_i^?\}, \mathbf{l}\tau_{\eta_i}, \mathbf{s}!\}) \\ &= (\{\mathbf{s}_i^?\}, \{\mathbf{l}\tau_{\eta_i}\}, \{\mathbf{s}!\}); \end{aligned}$$

Alternative composition by input:

$$\begin{aligned} \mathbf{s}?.(\mathbf{s}_1^0 + \mathbf{s}_2^0) &= (\{\mathbf{s}_1^0\} \cup \{\mathbf{s}_2^0\} \cup \{\mathbf{s}^?\}, \{\mathbf{s}^?\}, \\ &\{\mathbf{s}^?, \mathbf{l}\tau_{\eta_1}, \mathbf{s}_1^0\} \cup \{\mathbf{s}^?, \mathbf{l}\tau_{\eta_2}, \mathbf{s}_2^0\}) = (\{\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}^?\}, \\ &\{\mathbf{s}^?, \mathbf{s}_1!, \mathbf{s}_2!\}, \{\mathbf{l}\tau_{\eta_1}, \mathbf{l}\tau_{\eta_2}\}); \end{aligned}$$

Alternative composition by output:

$$\begin{aligned} \mathbf{s}!\mathbf{.}(\mathbf{s}_1^0 + \mathbf{s}_2^0) &= (\{\mathbf{s}_1^0\} \cup \{\mathbf{s}_2^0\} \cup \{\mathbf{s}!\}, \{\mathbf{s}!\}, \\ &\{\mathbf{s}_1^0, \mathbf{l}\tau_{\eta_1}, \mathbf{s}!\} \cup \{\mathbf{s}_2^0, \mathbf{l}\tau_{\eta_2}, \mathbf{s}!\}) = (\{\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}!\}, \\ &\{\mathbf{s}_1^?, \mathbf{s}_2^?, \mathbf{s}_2!\}, \{\mathbf{l}\tau_{\eta_1}, \mathbf{l}\tau_{\eta_2}\}). \end{aligned}$$

It can be seen from the presented definitions that the following obvious equalities are valid for Prefix operation and Postfix operation on functions  $\mathbf{s}_i^0$  and  $\mathbf{s}_j^0$ :  $\mathbf{s}_i^0?.\mathbf{s}_j^0 = \mathbf{s}_j^0!\mathbf{.s}_i^0$  and  $\mathbf{s}_i^0!\mathbf{.s}_j^0 = \mathbf{s}_j^0?.\mathbf{s}_i^0$ . In addition, Alternative composition by input and Alternative composition by output may be united into one operation as follows:

$$\begin{aligned} \mathbf{s}?.\mathbf{s}!\mathbf{.}(\mathbf{s}_1^0 + \mathbf{s}_2^0) &= (\{\mathbf{s}_1^0\} \cup \{\mathbf{s}_2^0\} \cup \{\mathbf{s}^?\} \cup \{\mathbf{s}!\}, \\ &\{\mathbf{s}^?\} \cup \{\mathbf{s}!\}, \{\mathbf{s}^?, \mathbf{l}\tau_{\eta_1}, \mathbf{s}_1^0\} \cup \{\mathbf{s}^?, \mathbf{l}\tau_{\eta_2}, \mathbf{s}_2^0\} \\ &\cup \{\mathbf{s}_1^0, \mathbf{l}\tau_{\eta_1}, \mathbf{s}!\} \cup \{\mathbf{s}_2^0, \mathbf{l}\tau_{\eta_2}, \mathbf{s}!\}) \\ &= (\{\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}^?, \mathbf{s}!\}, \{\mathbf{s}^?, \mathbf{s}!\}, \{\mathbf{l}\tau_{\eta_1}, \mathbf{l}\tau_{\eta_2}, \mathbf{l}\tau_{\eta_1}, \mathbf{l}\tau_{\eta_2}\}). \end{aligned}$$

We assume that the first pair of aggregation conditions for the aforementioned elements  $\mathbf{e}_i$  and  $\mathbf{e}_j$  is valid. Then, the system  $\mathbf{e}_{ij}$  is obtained as a union of elements  $\mathbf{e}_i$  and  $\mathbf{e}_j$ , which is represented by the expression

$$\mathbf{e}_{ij} = \langle (\{\mathbf{l}_i^?\}, \{\mathbf{l}_i!\}), (\{\mathbf{s}_{ij}^0\}), (\beta_i^?, \beta_j!) \rangle.$$

where the functionality of the element  $\mathbf{e}_{ij}$  may be defined as

$$\mathbf{s}_{ij}^0 = \mathbf{s}_i^0?.\mathbf{s}_j^0 = (\{\mathbf{s}_i^?\}, \{\mathbf{l}\tau_{ij}\}, \{\mathbf{s}_j!\}).$$

in accordance with Prefix operation (see Table 4 and its specifications).

If the second pair of aggregation conditions is fulfilled, then the system  $\mathbf{e}_{ij}$  obtained as a union of elements  $\mathbf{e}_j$  and  $\mathbf{e}_i$  will be represented by the expression

$$\mathbf{e}_{ij} = \langle (\{\mathbf{l}_j^?\}, \{\mathbf{l}_j!\}), (\{\mathbf{s}_{ij}^0\}), (\beta_j^?, \beta_i!) \rangle,$$

where the functionality of the element  $\mathbf{e}_{ij}$  may be defined as

$$\mathbf{s}_{ji}^0 = \mathbf{s}_i^0 \cdot \mathbf{s}_j^0 = (\{\mathbf{s}_j^0\}, \{\mathbf{I}\tau_{ji}\} \{\mathbf{s}_i^0\})$$

in accordance with Postfix operation (see Table 4 and its specifications).

In addition to graphical–analytical UFO-models that are formalized based on configurations with linear order connection  $\Sigma$ , the models with tree connection  $\Sigma$  (in terms of PT) [9] are of applied interest as well. This type of connection arises when, for example, a decision-making element with possible alternatives, which corresponds to the if...then...else... conditional operator in programming, has to be represented in the model.

To develop an algebraic description for aggregation of UFO-elements into a configuration with a tree connection, Alternative composition by input and Alternative composition by output (Table 4) will be used. The total of three elements will be considered. First of all,  $\mathbf{e}_i$  and  $\mathbf{e}_j$  are considered in the same sense as earlier, i.e., elements that correspond to two alternative work flows. In addition, the following element is introduced:

$$\mathbf{e}_k = \langle (\{\mathbf{I}_k^0\}, \{\mathbf{I}_{k1}^0, \mathbf{I}_{k2}^0\}), (\{\mathbf{s}_k^0\}), (\beta_k^0, \beta_{k1}^0 \cup \beta_{k2}^0) \rangle,$$

which is an element that represents testing of a certain condition. Assume that the aforementioned aggregation condition is fulfilled in such a way that  $\mathbf{I}_{k1}^0 = \mathbf{I}_i^0$ ,  $\beta_{k1}^0 \subseteq \beta_i^0$ ;  $\mathbf{I}_{k2}^0 = \mathbf{I}_j^0$ ,  $\beta_{k2}^0 \subseteq \beta_j^0$ . Then, adjunction of the element  $\mathbf{e}_k$  to  $\mathbf{e}_i$  and  $\mathbf{e}_j$  will result in the formation of the system  $\mathbf{e}_{ijk}^P$ , which provides the branching of work flows represented by the expression

$$\mathbf{e}_{ijk}^P = \langle (\{\mathbf{I}_k^0\}, \{\mathbf{I}_i^0, \mathbf{I}_j^0\}), (\{\mathbf{s}_{ijk}^P\}), (\beta_k^0, \beta_i^0 \cup \beta_j^0) \rangle,$$

where the functionality of the element  $\mathbf{e}_{ijk}^P$  may be defined as

$$\begin{aligned} \mathbf{s}_{ijk}^P &= \mathbf{s}_k^0 \cdot (\mathbf{s}_i^0 + \mathbf{s}_j^0) \\ &= (\{\mathbf{s}_i, \mathbf{s}_j, \mathbf{s}_k\}, \{\mathbf{s}_k^0, \mathbf{s}_i^0, \mathbf{s}_j^0\}, \{\mathbf{I}\tau_{ki}, \mathbf{I}\tau_{kj}\}). \end{aligned}$$

in accordance with Alternative composition by input.

If work flow partitioning takes place, then these flows will eventually merge. Consider the variant of algebraic description of aggregation of UFO-elements, where work flow merging occurs. Element  $\mathbf{e}_i$  will be considered in the same sense as earlier, i.e., as an element of one of the alternative work flows, testing element  $\mathbf{e}_k$  will be simplified as follows:

$$\mathbf{e}_k = \langle (\{\mathbf{I}_k^0\}, \{\mathbf{I}_k^0\}), (\{\mathbf{s}_k^0\}), (\beta_k^0, \beta_{k1}^0) \rangle,$$

and the element that represents merging of flows into one of the alternatives,  $\mathbf{e}_j$ , will, on the contrary, be complicated:

$$\mathbf{e}_j = \langle (\{\mathbf{I}_j^0, \mathbf{I}_2^0\}, \{\mathbf{I}_j^0\}), (\{\mathbf{s}_j^0\}), (\beta_{j1}^0 \cup \beta_{j2}^0, \beta_{j1}^0) \rangle,$$

We assume that the aforementioned aggregation condition is fulfilled in a way such that  $\mathbf{I}_i^0 = \mathbf{I}_1^0$ ,  $\beta_i^0 \subseteq \beta_{j1}^0$ ;  $\mathbf{I}_k^0 = \mathbf{I}_2^0$ ,  $\beta_k^0 \subseteq \beta_{j2}^0$ . Then, adjunction of  $\mathbf{e}_i$  and  $\mathbf{e}_k$  to the element  $\mathbf{e}_j$  will result in the formation of the system

$\mathbf{e}_{ijk}^C$  which provides merging of the work flows represented by the expression

$$\mathbf{e}_{ijk}^C = \langle (\{\mathbf{I}_i^0, \mathbf{I}_k^0\}, \{\mathbf{I}_j^0\}), (\{\mathbf{s}_{ijk}^C\}), (\beta_i^0 \cup \beta_k^0, \beta_j^0) \rangle,$$

where the functionality of the element  $\mathbf{e}_{ijk}^C$  may be defined as

$$\begin{aligned} \mathbf{s}_{ijk}^C &= \mathbf{s}_j^0 \cdot (\mathbf{s}_i^0 + \mathbf{s}_k^0) \\ &= (\{\mathbf{s}_i, \mathbf{s}_j, \mathbf{s}_k\}, \{\mathbf{s}_i^0, \mathbf{s}_k^0, \mathbf{s}_j^0\}, \{\mathbf{I}\tau_{ij}, \mathbf{I}\tau_{kj}\}). \end{aligned}$$

in accordance with Alternative composition by output.

Finally, we consider the often encountered situation where branching of the flows, as well as their merging, occur on the same three elements. We consider the element of one of alternative flows  $\mathbf{e}_i$  in the same sense as in all previous cases, the testing element  $\mathbf{e}_k$  as in case of branching flows, and the element that represents merging of flows as in the previous case. We assume that the aforementioned aggregation condition is fulfilled in such a way that  $\mathbf{I}_{k1}^0 = \mathbf{I}_i^0$ ,  $\beta_{k1}^0 \subseteq \beta_i^0$ ;  $\mathbf{I}_{k2}^0 = \mathbf{I}_j^0$ ,  $\beta_{k2}^0 \subseteq \beta_j^0$ ;  $\mathbf{I}_i^0 = \mathbf{I}_2^0$ ,  $\beta_i^0 \subseteq \beta_{j2}^0$ . Then, adjunction of elements  $\mathbf{e}_i$ ,  $\mathbf{e}_k$ , and  $\mathbf{e}_j$  to each other will result in formation of the system  $\mathbf{e}_{ijk}^{PC}$ , which provides both branching and merging of work flows at the same time, represented by the expression

$$\mathbf{e}_{ijk}^{PC} = \langle (\{\mathbf{I}_k^0\}, \{\mathbf{I}_i^0\}), (\{\mathbf{s}_{ijk}^{PC}\}), (\beta_k^0, \beta_j^0) \rangle,$$

where the functionality of the element  $\mathbf{e}_{ijk}^{PC}$  may be defined as

$$\begin{aligned} \mathbf{s}_{ijk}^{PC} &= \mathbf{s}_k^0 \cdot \mathbf{s}_j^0 \cdot (\mathbf{s}_i^0 + \mathbf{s}_j^0) \\ &= (\{\mathbf{s}_i^0\} \cup \{\mathbf{s}_j^0\} \cup \{\mathbf{s}_k^0\}, \{\mathbf{s}_k^0\} \cup \{\mathbf{s}_j^0\}, \\ &\quad \{\mathbf{s}_k^0, \mathbf{I}\tau_{ji}, \mathbf{s}_i^0\} \cup \{\mathbf{s}_k^0, \mathbf{I}\tau_{ji}, \mathbf{s}_j^0\} \cup \{\mathbf{s}_i^0, \mathbf{I}\tau_{i1}, \mathbf{s}_j^0\}) \\ &= (\{\mathbf{s}_i, \mathbf{s}_j, \mathbf{s}_k\}, \{\mathbf{s}_k^0, \mathbf{s}_j^0\}, \{\mathbf{I}\tau_{ki}, \mathbf{I}\tau_{kj}, \mathbf{I}\tau_{ij}\}). \end{aligned}$$

in accordance with the union of the operations of Alternative composition by input and Alternative composition by output under the condition of  $\mathbf{s}_2^0 = \mathbf{s}_j^0$  (Table 4).

## CONCLUSIONS

A new method for formal description of systems in terms of “Units”, “Functions”, and “Objects” has been developed on the basis of a comparative study and integration of algebraic tools from Grenander’s theory of patterns and Milner’s process calculus. Thus, it has become possible to formulate the basic

terms of the system calculus as three-element “Unit-Function-Object” constructions .

A formalization technique for visual graphical–analytical models of administrative procedures has been developed based on the analysis of the execution of administrative procedures and the application of basic terms of the system calculus as UFO-elements. Thus, it has become possible to propose and formally describe a dedicated interface method for the decomposition of administrative procedures; to formalize aggregation of elements of graphical–analytical models of administrative procedures into configurations with linear order and tree connections; to formalize nonlinear elements of graphical–analytical models of administrative procedures (which represent branching and merging of work flows); to improve the definitions of aggregation conditions (rules of system decomposition) for UFO-elements; to propose new operation on functions and specify the definitions of operations that were proposed earlier.

The proposed formal tools are successfully applied for the modeling and analysis of administrative procedures, which corresponds to government and municipal services provided to people in electronic form within the federal E-Russia program.

#### ACKNOWLEDGMENTS

The authors thank A.B. Petrovskii for useful notes. The work was supported by the Russian Foundation for Basic Research, project no. 10-07-00266.

#### REFERENCES

1. Matorin, S.I., On new method of systemologic analysis matched with procedure object-oriented designing. Part 2, *Kibern. Sist. Anal.*, 2002, no. 1, pp. 118–130.
2. Matorin, S.I., Popov, A.S., and Matorin, V.S., Simulation of organization systems according to new “node–function–object” approach, *Nauchn.-Tekhn. Inform.*, Ser. 2., 2005, no. 1, pp. 1–8.
3. Matorin, S.I., Zimovets, O.A., and Trubitsin, S.N., Visual graphoanalytical models for presentation about service of teleradionet, *Iskusst. Intel. Prin. Resh.*, 2008, no. 3, pp. 52–63.
4. Zimovets, O.A., Zhikharev, A.G., Matorin, S.I., and Trubitsin, S.N., System-object simulation of television and broadcasting net, *Inform. Tekhnol. Vychisl. Sist.*, 2009, no. 3, pp. 75–87.
5. Mikhelev, M.V. and Matorin, S.I., Simulation of business-processes in management of external illumination, *Zh. Nauchn. Publ. Aspir. Doktor.*, 2009, no. 3, pp. 136–139.
6. Matorin, S.I. and El’chaninov, D.B., Application of pattern theory for formalization of systemological UFO-analysis, *Nauchn.-Tekhn. Inform.*, Ser. 2., 2002, no. 11, pp. 1–11.
7. Mikhelev, M.V. and Matorin, S.I., Formalization UFO-elements with help of algebraic device of PI-calculation, *Nauchn. Vedom. Bel. Gos. Univ.*, Ser. “Informatika”, 2010, no. 19, pp. 145–150.
8. Zhikharev, A.G. and Matorin, S.I., Method for formalization of organizational knowledge, *Iskusstv. Intel. Prin. Resh.*, 2011, no. 2, pp. 52–63.
9. Grenander, U., *Lectures in Pattern Theory. Vol. 1. Pattern Synthesis*, New York: Springer-Verlag, 1976.
10. Milner, R., Parrow, J., and Walker, D.A., *Calculus of Mobile Processes, Part I*, LFCS Report 89–85, University of Edinburgh, 1989.
11. Zhikharev, A.G. and Matorin, S.I., On new formalized method of organizational knowledge presentation, *Nauchn. Vedom. Bel. Gos. Univ.*, Ser. “Informatika”, 2010, no. 19, pp. 133–140.
12. Dubeikovskii, V.I., *Praktika funktsional’nogo modelirovaniya s all fusion process modeler 4.1. Gde? Zachem? Kak?* (Practice of Functional Simulation with All Fusion Process Modeler 4.1. Where? For What? How?), Moscow: DIALOG – Mos. Inzh. Fiz. Inst., 2004.

*Translated by A. Amitin*