

## Formal bases of optimization procedures of system-object imitation models of processes and systems

Zhikharev A.G.<sup>1\*</sup>, Egorov I.A.<sup>2</sup>, Matorin S.I.<sup>2</sup>, Smirnova S.B.<sup>2</sup>, Udovenko I.V.<sup>2</sup>.,  
Shamraev A.A.<sup>1</sup>

<sup>1</sup> Belgorod State Technological University named after V.G.Shukhov

<sup>2</sup> Belgorod National Research University

zhikharev@bsu.edu.ru

### Abstract

*The paper discusses some optimization methods for system-object simulation models of processes and systems. The authors proposed some optimization principles in the article in order to increase the efficiency of the system-object analysis of organizational, business and industrial processes by improving the theoretical and instrumental means of optimizing the system-object simulation models. The paper shows that the optimization of the system-object model is firstly required in order to establish the conformity of the model with system-wide principles and patterns.*

*When constructing a simulation model, the authors are guided by the generalized characteristic of the system, that is a measure of systemicity. The article proposes the optimization of a system-object simulation model on the base of the management model of "supervisor-subordinates" in the environment of "UFOModeler". It is proved that the proposed optimization principles allow us to proceed to the development of optimization methods for structure, function and object parameters of similar simulated systems. Also, it should be noted that the principles of optimization of system-object simulation models considered in the work are discussional.*

**Keywords:** *system-object model, optimization of a model, system-wide principles and patterns, UFOModeler, monocentric idea.*

## 1. Introduction

In view of rapid development of science and technology in the modern world, developers, designers, engineers are increasingly resorting to the use of simulation as a method of studying objects and processes of the surrounding world. Moreover, it should be noted that the studied objects are becoming more complicated every year from the point of view of their structure, behavior and management of such objects, at the same time, appropriate tools and techniques are required to study and manage such complex systems. One of such techniques is simulation modeling, thanks to which there are possibilities to build simulators that are specialized software systems that replace an object or a real-world process with a sufficient degree of accuracy. System-object simulation is a modern method of constructing simulation models, based on the original graph-analytical approach "Unit-Function-Object". Moreover, the main feature of system-object simulation models is the ability to build a model that meets the system-wide principles and patterns [1-4]. The relevance of the considered problems is determined by the need to bring the simulated systems in line with the general system principles and laws, as shown in [5].

## 2. Methodology

For a formal statement of the problem, we use the provisions of calculus of systems as functional objects [6-9].

In terms of the calculus mentioned above, a system model is represented as:

$$M = (L, S), \tag{1}$$

where:

- M is a model of a system;
- L is a set of flow objects of a model, which elements are objects possessing only areas and no methods:

$$l=[r_1, r_2, \dots, r_k], \tag{2}$$

where:

- $l \in L$ ;
- k represents an amount of fields of the flow object l;
- $r_1, r_2, \dots, r_k$  are the areas of the flow object, that represent the “identifier – meaning” match;
- S is an array of the unit objects of a model.

Previously, the authors formulated a number of optimization methods, shown in detail in [10]. Let us consider some of the formal foundations of the optimization procedures for system-object models of systems, presented in terms of calculus of systems as functional objects. For that, it is proposed to use, firstly, some system-wide principles and patterns to serve as the optimality criteria [2]. Let us consider in more details the principle of communicativeness, according to which any system is to be connected to the environment (other systems) by a variety of communications, otherwise the existence of the system does not make any sense. If one considers a system-object model of the form (1), then in terms of calculus of systems, this principle can be formally represented in the following form:

$$\forall s \in S: \exists s. U = \emptyset \tag{3}$$

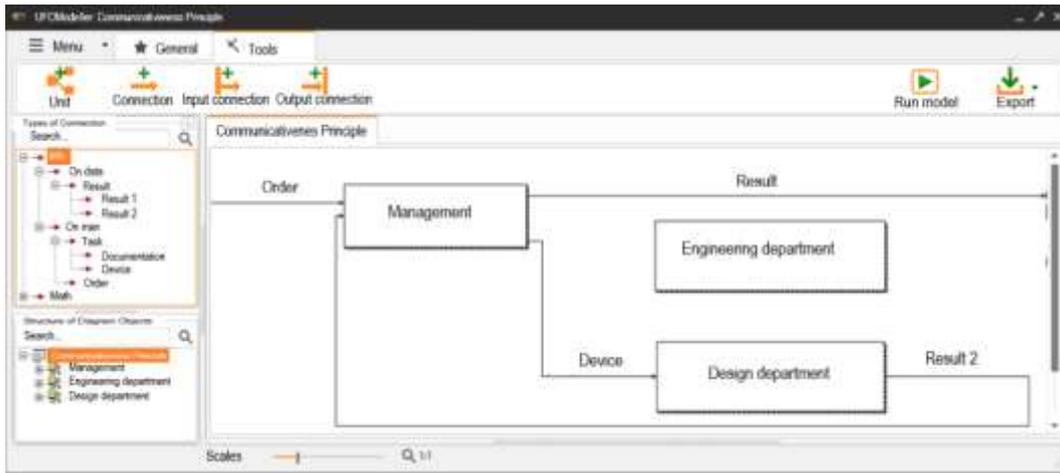
As in terms of system calculus, the connections of the unit object with the external environment are presented in the form of its interface stream objects  $U=(L?, L!)$ , then the correspondence of the modulated system to the principle of communicativeness is determined by the absence of the unit objects with an empty set of interface flow objects in the model. However, for the situation appearing when the unit object possesses the input and no output stream objects, or vice versa, it contradicts to the principle of communicativeness, therefore this principle can be clarified in the following form:

$$\forall s \in S: \exists s. L_? = \emptyset \ || \ s. L_! = \emptyset \tag{4}$$

Thus, the process of optimizing the model according to the principle of communicativeness consists in linking of the existing unit objects to others.

### 3. Results and Discussion

Let us consider the case of optimizing the principle of communicative model of the organizational and business process. The UFOModeler tool for modeling systems is used [11]. A model is presented in the following form (Fig. 1).



**Figure 1. Algorithm for implementing the unit object method**

Let us designate the flow objects of the model represented by the model:

- 11 for “documentation” (indicated in the “Link types” menu in Figure 1);
- 12 for “result 1” (indicated in the “Link types” menu in Figure 1);
- 13 for “device”;
- 14 for “result 2”;
- 15 for “order”;
- 16 for “result”.

At the same time, each of the stream objects contains the area of the stream object:  $r1$  for “quantity”.

Thus, a lot of stream objects takes the form:

$$L = \{ 11[r1], 12[r1], 13[r1], 14[r1], 15[r1], 16[r1] \} \quad (5)$$

Here are the formal names of the units of the model above:

- $s1$  for “management”;
- $s2$  for “engineering department”;
- $s3$  for “design department”.

Then, according to the formal description, the set of unit objects takes the following look:

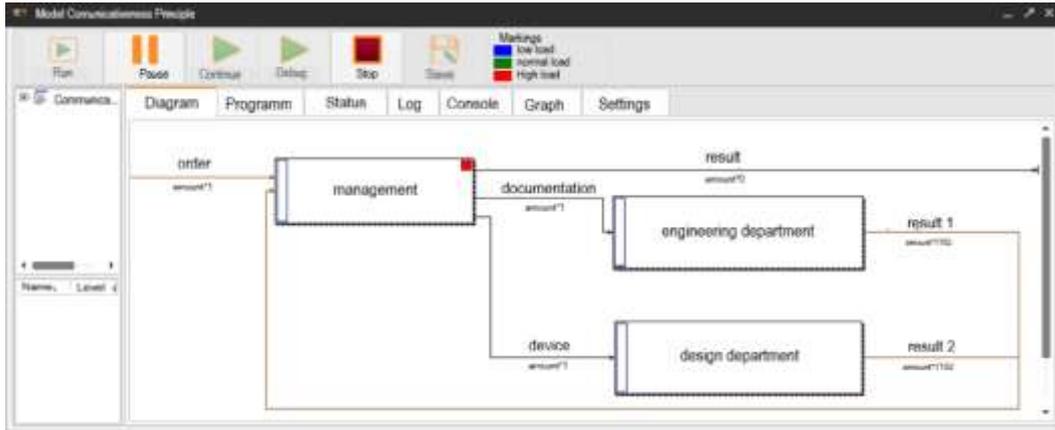
$$S = \{ s1[l? = \{14, 15\}, l! = \{13, 16\}; f(l?)l!; o], s2[l? = \emptyset, l! = \emptyset; f(l?)l!; o], s3[l? = \{13\}, l! = \{14\}; f(l?)l!; o] \} \quad (6)$$

According to the structure of the model set of unit objects (6), in the presented model of the organizational and business process, the “management” unit has an input interface  $l? = \{14, 15\}$ . This interface serves as a means to control the results of orders executions by the “design department” (like, device assembly), to receive tasks from the supersystem. Another interface of the model is an outcome interface  $l! = \{13, 16\}$ . The interface serves as a means to issue instructions to the subsystem units, to report on the results to the supersystem. The “design department” unit uses the input interface  $l? = \{13\}$  to accept instructions for processing. The output interface  $l! = \{14\}$  serves as a means of the “design department” to report on the execution of the tasks to the management.

It is worth noting that the stream objects 11, 12 that unite the “management” and “engineering department” units are absent, and, in this case, the “engineering department” is to stand idle due to the lack of instructions (preparation of documentation) for execution. Thus, the model under analysis does not meet the principle of communicativeness. The optimization by the principle of communicativeness is to be carried out. We apply the join operators to the model under consideration:

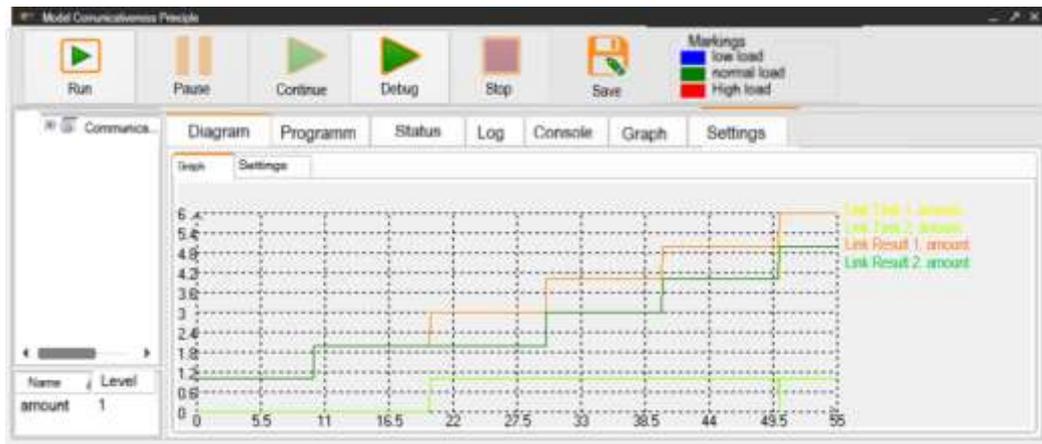
$$s_1 \xrightarrow{l_1} s_2; s_2 \xrightarrow{l_2} s_1 \quad (7)$$

Further, the action will be performed (7) on the model diagram and run the model in the UFOModeler environment on-line. The model optimization results are presented below (Fig. 2).



**Figure 2. The optimized model on function**

The receipt dynamics of new instructions and their implementation by departments over time is shown below (Fig. 3).



**Figure 3. The growth in the number of completed orders in an optimized model**

It can be seen in the table that over time, the number of completed orders by both departments is growing, while one new order is accepted at the same time. Thus, if any unit in the system does not communicate with the outside world through its own interfaces, then the system is not optimal from the point of view of the communication principle. It is worth noting that such an organization in which the units are not connected by streaming objects

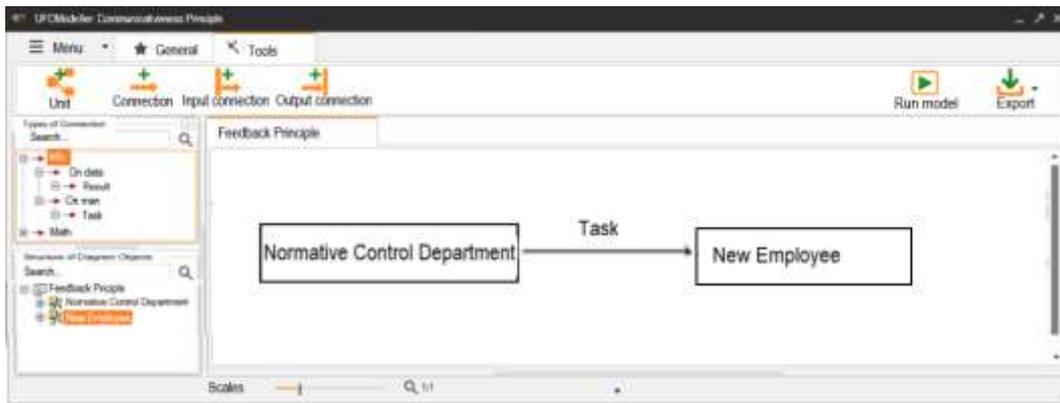
will not be able to fully function, which will ultimately lead to a decrease in interest in its services and a decrease in the number of orders.

Next, we consider the process of optimizing the model according to the feedback principle, according to which stability in complex dynamic systems is achieved by closing feedback loops. Formally, this principle can be represented as the following expression:

$$\exists s_i \in S, s_j \in S: l_m \in s_i \cdot L_2, l_m \in s_j \cdot L_1, l_n \in s_j \cdot L_2, l_n \in s_i \cdot L_1 \quad (8)$$

Accordingly, the optimization of the model according to the feedback principle consists in joining the corresponding types of links of nodal objects using the join operation.

Here is an example of optimizing the organizational and legal model of the system on the basis of feedback. We compose the model in the UFOModeler environment, as presented below (Fig. 4).



**Figure 4. An example of a closed system connection**

Let us designate the flow objects of the model represented by the model:

- 11 for “commission”;
- 12 for “result”, the relationship between the new employee and the department is absent in the diagram (in Figure 4, the presence of the relationship is reflected in the “Types of Relations” window).

Similar to the previous example, the presented stream objects have the field of the stream object: r1 for "quantity". Many stream objects will be represented as follows:

$$L = \{ 11[r1], 12[r1] \} \quad (9)$$

The model represents the following units:

- s1 for «normative control department»;
- s2 for «new employee».

Formal description of the unit objects set is represented as follows:

$$S = \{ s1[l? = \emptyset, l! = \{11\}; f(l?)!; O], s2[l? = \{11\}, l! = \emptyset; f(l?)!; O] \} \quad (10)$$

The model considers the organization of the probationary period of a new employee in the department of normative control.

The “normative control department” unit issues an order for execution via the  $l! = \{11\}$  output interface. The “new employee” unit accepts the order through the  $l? = \{11\}$  input interface and proceeds with the execution of the order.

However, due to the lack of communication (I2) inside the normative control department, represented by the head of the department, it cannot control the process of fulfilling the instructions by a new employee. The head of the department cannot evaluate the average speed of execution of instructions by an employee:

- the employee completed all the tasks, but does not inform the management when ready to execute the following task;
- the employee does not cope with the tasks; the department submits too many orders for execution.

Thus, labor productivity drops sharply. The model under consideration does not meet the feedback principle.

The feedback optimization is to be carried out for this model. We carry out the connection of I2 and write the action in the form of an expression:

$$s_2 \xrightarrow{l_2} s_1 \quad (11)$$

The flow object I2 is to be added to the model diagram. Then, the model is run in real time (Fig. 5).



**Figure 5. Execution of the optimized model in real time**

The growth dynamics of the number of executed orders and the number of orders accepted for execution correlates with the previous example. The dynamics graph is similar to Figure 3.

It is worth noting the negative aspect in the functioning of structural units of enterprises with an untimely report of employees on the execution of orders.

Thus, in order to increase labor productivity, if there is a “head-subordinate” relationship, then to increase production efficiency, optimization based on the feedback principle is necessary.

Next, we consider the principle of monocentrism, which follows from the principle of communicativeness, according to which a stable system has one center, and polycentricity leads to disruption of coordination processes, which in the long run leads to loss of integrity [12].

Formally, the principle of monocentrism can be represented as follows:

$$\forall s \in S: \exists s. L_2 = \{l_1\} || s. L_1 = \{l_1, l_2 \dots l_m\} \quad (12)$$

Thus, the optimization of model (1) according to the principle of monocentrism consists in eliminating streaming connections that allow disrupting the coordination process of subordinate objects. The elimination of relationships is carried out by the following methods:

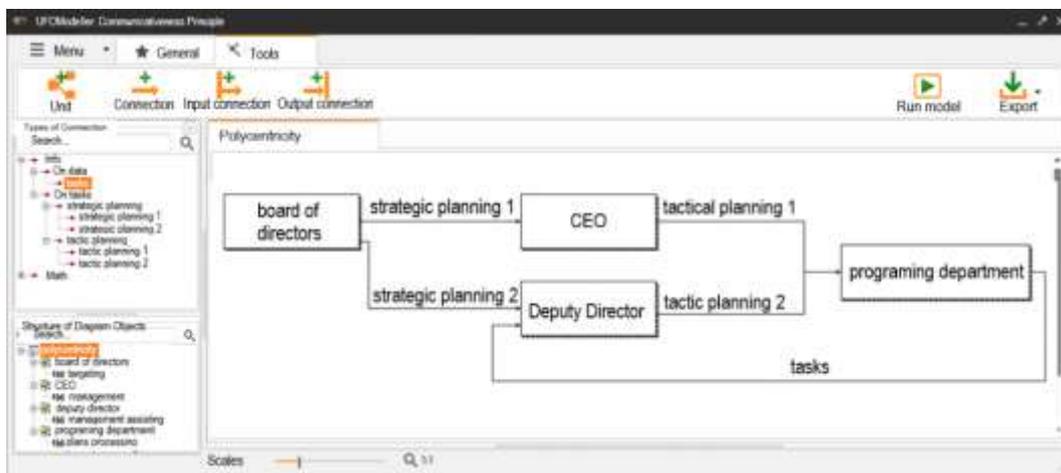
- combining control objects through output stream objects;

- building a hierarchy among control objects.

In the first case, the objects of the super-system form one common control stream object. This method is applicable under additional conditions for  $s_i$  and  $s_j$  units with  $L_i \equiv L_j$  and  $O_i = O_j$ . That is, the control objects are to have the same output interfaces and similar stream objects.

In the second case, optimization according to the principle of monocentrism can be achieved by building a chain of control objects in a hierarchy, so that each subsequent object becomes subordinate to a superior one up to the target object. Moreover, each object in the built-in hierarchy, when forming its own control signal, takes into account everything transferred to it through the input interface from the parent unit. As a result, each object in the chain exerts its own indirect controlling effect on the final object.

Consider, as an example, a model of the organizational and business process, in which the mechanism of interaction between the structural units of a research and production enterprise when planning future tasks is presented. The model is compiled in the UFOModeler environment (Fig. 6).



**Figure 6. Model of a research and production enterprise**

This model does not meet the requirements of the principle of monocentrism, since the diagram contains the connections “tactical planning 1” and “tactical planning 2”, which can have a mutually antagonistic control effect on the object of the “programming department” node, which ultimately violates the integrity of the system. The result of real-time simulation of such a model in relation to the performance of the “programming department” node is reduced to the function graph, indicated in green by light green in Figure 3 (lower efficiency).

Here is a list of objects involved in the preparation of a formal description of optimization actions on a model. Stream objects in the presented model are denoted as follows:

- 11 for “strategic planning 1”;
- 12 for “strategic planning 2”;
- 13 for “tactical planning 1”;
- 14 for “tactical planning 2”;
- 15 for “tasks”.

In this case, the stream objects 11-14 possesses the field of the r1 stream object that is the "plan" of the string type, and the 15 – r0 stream object possesses the "completed" mark of the integer type. Accordingly, a set of stream objects takes the form:

$$L = \{ 11[r1], 12[r1], 13[r1], 14[r1], 15[r0] \} \quad (13)$$

Here are the formal names of the units of the above model:

- s1 for “board of directors”;
- s2 for “CEO”;
- s3 for “deputy director”;
- s4 for “programming department”.

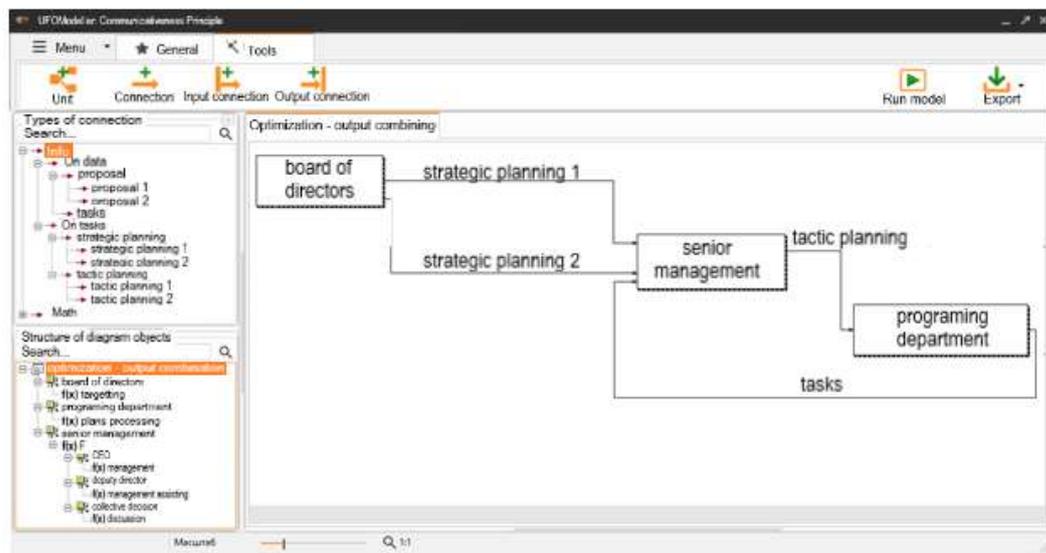
Then, according to the formal description, the set of unit objects looks like this:

$$S = \{ s1[l? = \emptyset, l! = \{11, 12\}; f(l?)!; O], s2[l? = \{11\}, l! = \{13\}; f(l?)!; o], s3[l? = \{12, 15\}, l! = \{14\}; f(l?)!; O], s4[l? = \{13, 14\}, l! = \{15\}; f(l?)!; O] \} \quad (14)$$

In the model, s1, s2, s3 units carry out a control action through communications according to the hierarchical structure of the enterprise. The control effect is reflected, firstly, for the goals issued by the s1 unit, in the form of a strategic plan for senior management of the enterprise (the s2, s3 units), and secondly, in the formulated tactical tasks that the senior management of the enterprise puts before the departments (the s4 unit of programming department is taken). The s4 unit reports on tasks to the s3 unit by transmitting data on the number of completed tasks.

We also believe that the input interfaces of the s2, s3 units received equal values of the fields of stream objects, which is an important criterion when considering an optimization example by the method of “building a hierarchy among control objects”.

The optimization is divided into two steps according to the method of “combining control objects through output stream objects”. At the first step, it is necessary to combine the s2, s3 units into a common subsystem of “senior management” (Fig. 7).



**Figure 7. Integration of enterprise management into a general subsystem**

We denote the subsystem of "senior management" by the set  $S^*$ . The  $l_1$  and  $l_2$  input stream objects, as well as the  $l_8$  output stream object "tactical planning". Thus, the  $l_3$  and  $l_4$  stream objects no longer make sense and are to be removed from the  $L$  set. Formally, the above action can be written as follows:

$$L = L \setminus \{l_3, l_4\}, s_1 \xrightarrow{l_1} \rightarrow s^*, s_1 \xrightarrow{l_2} \rightarrow s^*, s^* \xrightarrow{l_8} \rightarrow s_4 \quad (15)$$

Then the  $S^*$  set is to contain the following elements:

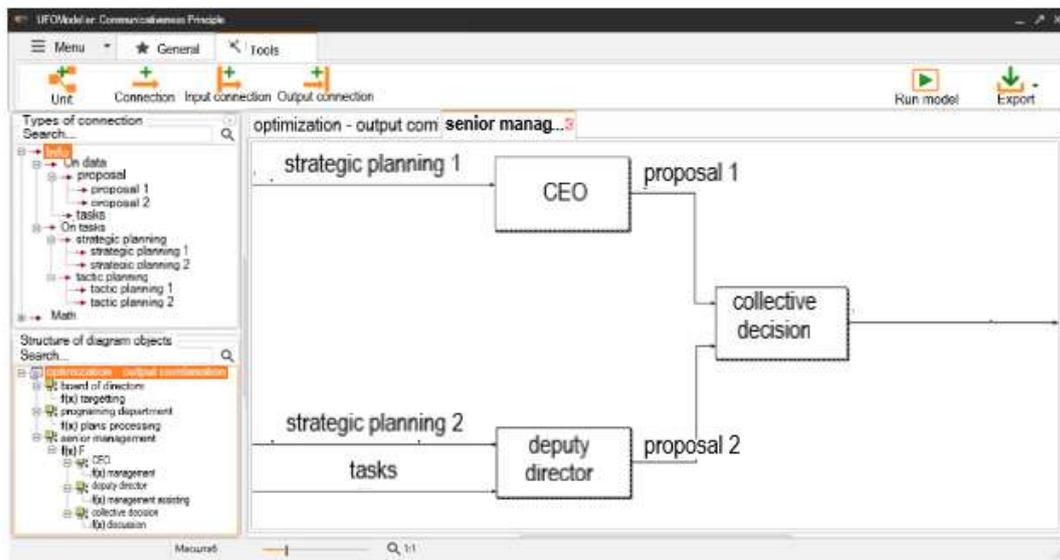
$$S^* = \{ s_2[l_1=\{11\}, l_2=\{13\};f(l_1)l_2;o], s_3[l_1=\{12, 15\}, l_2=\{14\};f(l_1)l_2;O] \} \quad (16)$$

The units forming the  $S^*$  subsystem are deleted from the  $S$  set, thus a new  $S'$  units set of the optimized model:

$$S = S \setminus S^*, S' = S \cup S^* \quad (17)$$

Next, the generated  $S^*$  subsystem is described. It contains the "CEO"  $s_2$  unit and the "Deputy Director"  $s_3$  unit. We supplement the subsystem with the  $s_{23}$  collective decision unit, showing the joint processing of the data of the two units.

It is worth mentioning that the union of units at the exit in the UFOModeler environment is achievable through the creation of a third unit and the addition of two new stream objects, which are  $l_6$  for "sentence 1" and  $l_7$  for "sentence 2", which are responsible for transferring the solution put forward for discussion to  $s_{23}$  unit (an analogue of the stream objects union), and then the subsystem forms the  $l_8$  output stream object (Fig. 8).



**Figure 8. The construction of unit objects of the subsystem "senior management"**

Let us present a formal description of the second step of the method under consideration:

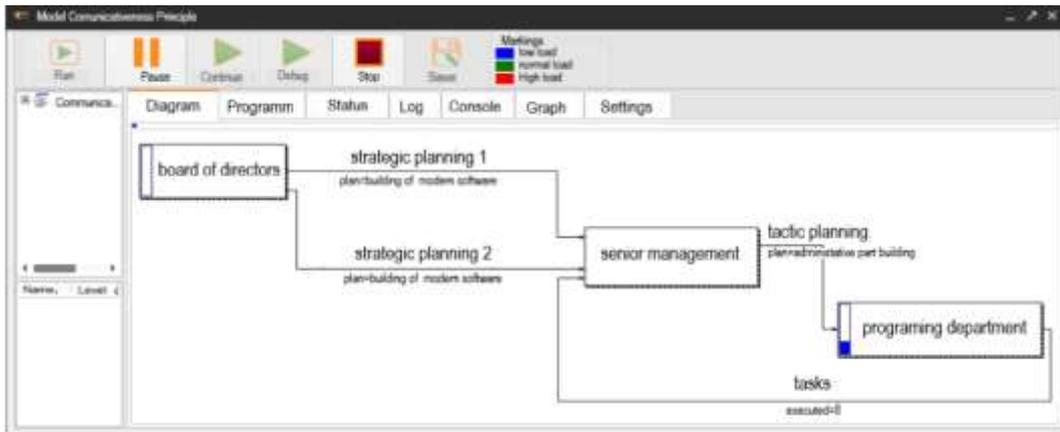
$$s_2 \xrightarrow{l_6} \rightarrow s_{23}, s_3 \xrightarrow{l_7} \rightarrow s_{23}, s_{23} \xrightarrow{l_8} \rightarrow s_4 \Rightarrow s_{23}[l_1=\{16, 17\}, l_2=\{18\};f(l_1)l_2;o], \quad (18)$$

$$S^* = S' \cup \{s_{23}\}$$

Since in a formal representation, the  $s_{23}$  unit is an imitation of the union of stream objects with a concomitant change in the function of the object for  $s_3$  unit.

$$s_3.f \leftarrow [f(l_1)l_2 \rightarrow f(l_1, l_2)l_2; l_1 \rightarrow l_2; o_1 \rightarrow o_2, of_{23}] \quad (19)$$

The simulation is run in real time (Fig. 9).



**Figure 9. Optimization by building a hierarchy among control objects**

It can be seen from the results of the model that through the input interfaces the S\* subsystem received a control action from the s1 unit (the fields r1 of the I1 and I2 stream objects contain the meaning “creation of modern software”). Next, let us imagine the work of the S\* subsystem. Objects of the s2 and s3 units made suggestions for implementing the strategic plan through the I6 and I7 stream objects for discussion (field I6 . r1 of the stream object contains the meaning "create the administrator part"; field I7 . r1 of the stream object contains the meaning "create the user part"), came to a common opinion (the r1 field of the I8 stream object contains the meaning “creation of the administrator part”), passed the general control action through the I7 stream object. Thus, the input interface of the s4 “programming department” unit got unanimous control action. In accordance with the tactical plan, the programming department built an operational plan, within which it is necessary to implement 30 tasks. The blue color in the progress column on the left side of the “programming department” unit indicates the number of completed tasks. The graph of the task execution function over time coincides with the orange function in Figure 3.

The optimization is carried out by the method of “building a hierarchy among control objects”. The direction of the I3 stream object is changed in the original model. The connection between the s2 and s4 units is deleted and the I3 stream object is entered between the s2 and s4 units (Fig. 10).



### Figure 10. Relationship diagram of the optimized model

We write a formal representation of this method:

$$s_2 \cdot L_1 = L_1 \setminus \{l_3\}, s_4 \cdot L_7 = L_7 \setminus \{l_3\}, s_2 \xrightarrow{l_3} s_3 \quad (20)$$

The model for execution is run in real time. The result of the model's function as a whole is comparable with the result shown in Figure 9. The difference is that the control action taken by the s2 unit with respect to the s3 unit is advisory and may contradict the control action of the s3 unit with respect to the s4 unit. This aspect distinguishes the two described optimization methods. At the same time, the l2 and l3 flow objects entering the s3 unit cannot be considered polycentric, since the control action provided by the links does not contradict each other (the control effect of the l3 flow objects is based on the control action of the s1 unit) and does not violate the coordination of the controlled object, therefore, in this case, the principle of monocentrism is not violated.

## 4. Conclusion

Thus, the object of s4 "programming department" unit, in a model optimized by the principle of monocentrism, is engaged in solving problems that are consistent with all control links. In this case, the s4 unit does not experience additional control action, which will certainly affect the speed of tasks in a positive way.

The considered optimization principles according to the relevant system-wide principles and laws show that the presence of correctly built connections is necessary to create an effective, communicating system. At the same time, the life experience of any organizational system always proves that compliance with the principles proposed by the principles makes the organizational system resistant to manifestations of the external environment.

In the future, on the basis of system-object models of processes and systems formulated by the principle of optimization, the authors will develop methods and algorithms for model optimization that will automate these procedures.

## Acknowledgments

The research is financially supported by the projects of the Russian Foundation for Fundamental Research No. 18-07-00355, No. 19-07-00290 and No. 19-07-00111.

## References

- [1] Matorin S.I., Zhikharev A.G. "System-wide regularities as the substantive elements of a system theory based on a system-object approach", Scientific Bulletin of Belgorod State University. Series: Economics, IT., vol.45, no 2, (2018), pp. 372-385.
- [2] Matorin S.I., Zhikharev A.G. "Accounting for system-wide regularities in the system-object modeling of organizational knowledge", Scientific and Technical Information Processing, vol. 46, no 6, (2019), pp. 1-9.
- [3] Zhikharev A.G., Matorin S.I., Kuprieva I.A., Kulumbaev E.B., Smirnova S.B. "The language of the description of the functional objects", Journal of Engineering and Applied Sciences, vol. 12, special is. 6, (2017), pp. 7859-7864.
- [4] Zhikharev A.G., Matorin S.I., Kuprieva I.A., Smirnova S.B., Boiko A.I. "On the language handling technology and its peculiarities", Journal of Engineering and Applied Sciences, vol. 12, special is. 4, (2017), pp. 6864-6869.

- [5] Matorin S.I., Zhikharev A.G., Belov S.P. “The view of system-wide laws during the modelling complex systems unis the system-object approach”, *Science. Innovations. Technologies*, no 3, (2018), pp. 7-21.
- [6] Matorin S.I., Zhikharev A.G. “Calculation of the function objects as the systems formal theory basis”, *Advances in Intelligent Systems and Computing*, 679, (2018), pp. 182-191.
- [7] Matorin S.I., Zhikharev A.G., Zimovets O.A. “Object Calculus in the Sys-tem–Object Method of Knowledge Representation”, *Scientific and Technical Information Processing*, vol. 45, no. 5, (2018), pp. 307-316.
- [8] Zhikharev A.G., Matorin S.I., Zaitseva N.O. “About perspectives of simulation technological processes functioning with using system-object approach node-function-object”, *International Journal of Applied Engineering Research*, vol. 10, no. 12, (2015), pp. 31363-31370.
- [9] Matorin S.I., Zhikharev A.G., Zimovets O.A. “The elements of general theory of the systems in terms of system-object approach of «Unit-Function-Object»”, *International Journal of Applied Engineering Research*, vol. 10, no. 24, (2015), pp. 44831-44837.
- [10] Zhikharev A.G., Buzov A.A., Egorov I.A., Kuznetsov A.V., Zhinkina Yu.V. “Optimization of system-object simulation models. Part 1”, *Scientific Bulletin of Belgorod State University. Series: Economics, IT.*, vol. 46, no 2, (2019), pp. 311-325.
- [11] Matorin S.I., Zhikharev A.G., Zaitseva N.O. “Simulation using a system-object approach”, *Applied Informatics*, vol. 10, no. 6 (60), (2015), pp. 91-104.
- [12] Bogdanov A. A. “Tectology: General organizational science. 2 parts”, Moscow, Economics, (1989).