

УДК 004.94; 658.5

DOI: 10.18413/2518-1092-2018-3-4-0-5

Бузов А.А.<sup>1</sup>  
Жихарев А.Г.<sup>2</sup>  
Маторин С.И.<sup>2</sup>

**О ПОСТРОЕНИИ БИБЛИОТЕК В СИСТЕМНО-ОБЪЕКТНЫХ  
ИМИТАЦИОННЫХ МОДЕЛЯХ**

<sup>1)</sup> ЗАО «СофтКоннект», ул. Студенческая д. 19, корпус 1, г. Белгород, 308023, Россия

<sup>2)</sup> Белгородский государственный национальный исследовательский университет, ул. Победы д. 85,  
г. Белгород, 308015, Россия

*e-mail: buzov@mail.ru, zhikharev@bsu.edu.ru, matorin@bsu.edu.ru*

**Аннотация**

Статья посвящена вопросам разработки методики создания библиотек узловых объектов для реализации компонентного подхода в системно-объектном имитационном моделировании. С применением исчисления функциональных объектов – аппарата, формализующего процедуры системно-объектного моделирования процессов и систем, авторами сформулированы теоретические положения процедур построения библиотек узловых объектов, а также дальнейшего их использования в системно-объектных моделях. Рассматриваются формальные описания процесса создания библиотеки, операции импорта и экспорта узловых объектов. Сформулировано определение интерфейса узлового объекта, что, в свою очередь, позволяет автоматически подбирать узловые объекты, находящиеся в библиотеке для использования в системно-объектной модели. Кроме того, авторами рассматриваются формальные основы процедур анализа элементов библиотеки на соответствие конкретному месту в системно-объектной модели. Это достигается за счет использования алгоритма расчета коэффициента системности, показывающего степень соответствия системы запросам надсистемы. Таким образом, сформулированные теоретические положения механизмов формирования библиотек узловых объектов позволяют реализовать программные механизмы хранения отдельных частей системно-объектной модели в специальной библиотеке-репозитории для дальнейшего их использования.

**Ключевые слова:** системно-объектный подход «Узел-Функция-Объект»; имитационное моделирование; библиотека элементов; внутренние параметры системы; УФО-элемент; узловой объект; потоковый объект.

UDC 004.94; 658.5

Buzov A.A.<sup>1</sup>  
Zhikharev A.G.<sup>2</sup>  
Matorin S.I.<sup>2</sup>

**ABOUT THE CONSTRUCTION OF LIBRARIES IN SYSTEM-OBJECT  
SIMULATION MODELS**

<sup>1)</sup> «SoftConnect» Ltd., 19 Studencheskaya St., building 1, Belgorod, 308023, Russia

<sup>2)</sup> Belgorod State National Research University, 85 Pobedy St., Belgorod, 308015, Russia

*e-mail: buzov@mail.ru, zhikharev@bsu.edu.ru, matorin@bsu.edu.ru*

**Abstract**

The article is devoted to the development of methods for creating libraries of nodal objects for implementing the component approach in system-object simulation modeling. With the use of calculus of functional objects - the apparatus, which formalizes the procedures of the system-object modeling of processes and systems, the authors formulated the theoretical provisions of the procedures for constructing libraries of node objects, as well as their further use in system-object models. We consider the formal description of the process of creating a library, the operation of

import and export of node objects. The definition of the node object interface is formulated, which, in turn, allows you to automatically select the node objects that are in the library for use in the system-object model. In addition, the authors consider the formal foundations of the procedures for analyzing library elements for compliance with a specific place in the system-object model. This is achieved through the use of an algorithm for calculating the coefficient of systemicity, which shows the degree of compliance of the system with the requirements of the super-system. Thus, the formulated theoretical positions of the mechanisms for the formation of libraries of nodal objects make it possible to implement software mechanisms for storing individual parts of the system-object model in a special library-repository for further use.

**Keywords:** system-object approach "Node-Function-Object"; simulation modeling; element library; internal system parameters; UFO-element; node object; streaming object.

Ввиду бурного развития науки и техники, в современном мире, разработчики, проектировщики, инженеры все чаще прибегают к применению имитационного моделирования как метода исследования объектов и процессов окружающего мира. Причем следует отметить, что исследуемые объекты с каждым годом усложняются с точки зрения их структуры, поведения и управления такими объектами. И для исследования и управления такими сложными системами требуются соответствующие средства и методики. Одной из таких методик является – имитационное моделирование, благодаря которому имеются возможности построения специальных симуляторов – программно-аппаратных комплексов замещающих объект или процесс реального мира с достаточной степенью точности. Большинство современных программных средств, позволяющих строить имитационные модели, содержат встроенные библиотеки готовых компонентов модели, что позволяет существенно сократить и упростить процедуру разработки имитационной модели. Наличие библиотек готовых элементов модели позволяет строить имитационную модель – как конструктор - из готовых частей, что естественно проще для разработчика, чем программировать модель «с нуля».

Системно-объектный метод имитационного моделирования представляет собой современную технологию описания функционирующих систем, в основу которой положен системный подход «Узел-Функция-Объект». С целью формализации процедур имитационного моделирования процессов и систем авторами разработаны положения исчисления функциональных объектов [1,2], в рамках которого системно-объектная модель представляется как:

$$M=(L,S), \quad (1)$$

где:  $M$  – модель системы;

$L$  – множество потоковых объектов модели, элементы которого представляют собою объект, которые не имеет методов и имеет лишь поля (2):

$$l=[r_1, r_2, \dots, r_k], \quad (2)$$

где:

$l \in L$ ;

$k$  – количество полей потокового объекта  $l$ ;

$r_1, r_2, \dots, r_k$  – поля потокового объекта, представляющие собой пару «идентификатор-значение».

$S$  – множество узловых объектов модели, элементы которого описываются следующей формой (3):

$$s=[U, f, O], \quad (3)$$

где:

$U$  – представляет собою множество полей для описания интерфейсных потоковых объектов узлового объекта  $s$ .

$f$  – представляет собою метод узлового объекта  $s$ , описывающий функцию преобразования входящих интерфейсных потоковых объектов (входящих связей системы)  $L?$  в выходящие -  $L!$ .

$O$  - представляет собою множество полей для описания объектных характеристик узлового объекта (системы)  $s$ .

Причем, узловые объекты модели  $M$  представляют собой ключевые элементы модели, а множество потоковых объектов – определяет отношения между узловыми объектами модели [3, 4].

Библиотека готовых элементов системно-объектной модели в таком случае будет иметь вид (1), причем  $|L|=0$ . То есть, модель-библиотека будет иметь лишь узловые объекты, и не будет иметь в своем составе потоковые объекты [5]. Тогда библиотеку системно-объектной модели можно рассматривать как множество узловых объектов следующего вида:

$$S^+ = [s_1, s_2, \dots, s_n], \quad (4)$$

где  $n$  – количество узловых объектов, хранящихся в библиотеке.

Рассмотрим подробнее элемент библиотеки, а точнее его формальную сторону [6, 7]. Как было отмечены выше, элемент библиотеки будет представлять собой отдельную смоделированную систему. В рамках исчисления функциональных объектов систему, описанную выражением (3), будем представлять в виде следующего выражения:

$$s_n = [L?, L!; f(L?)L!; O?, O!, Of] \quad (5)$$

Графический формализм, представляющий собою элемент библиотеки, представим в следующем виде:

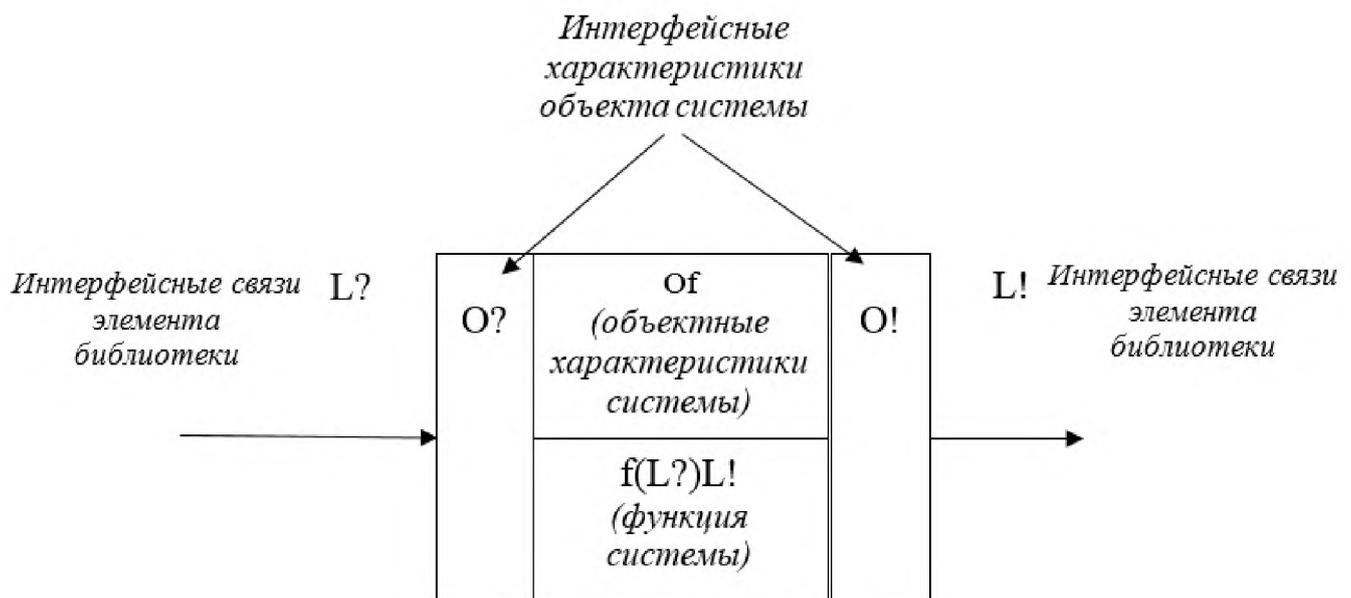


Рис. 1. Графический формализм элемента библиотеки  
Fig. 1. Graphical formalism of the library element

Как видно из рисунка 1, каждый элемент библиотеки представляет собой УФО-элемент с соответствующими интерфейсными связями, по которым и проводится анализ на соответствие текущего элемента заданным характеристикам. Соответственно, библиотека элементов будет представлять собою набор узловых объектов, не связанных между собой [8].

Рассмотрим абстрактную библиотеку  $S_M$  состоящую из следующих элементов, как показано на рисунке 2.

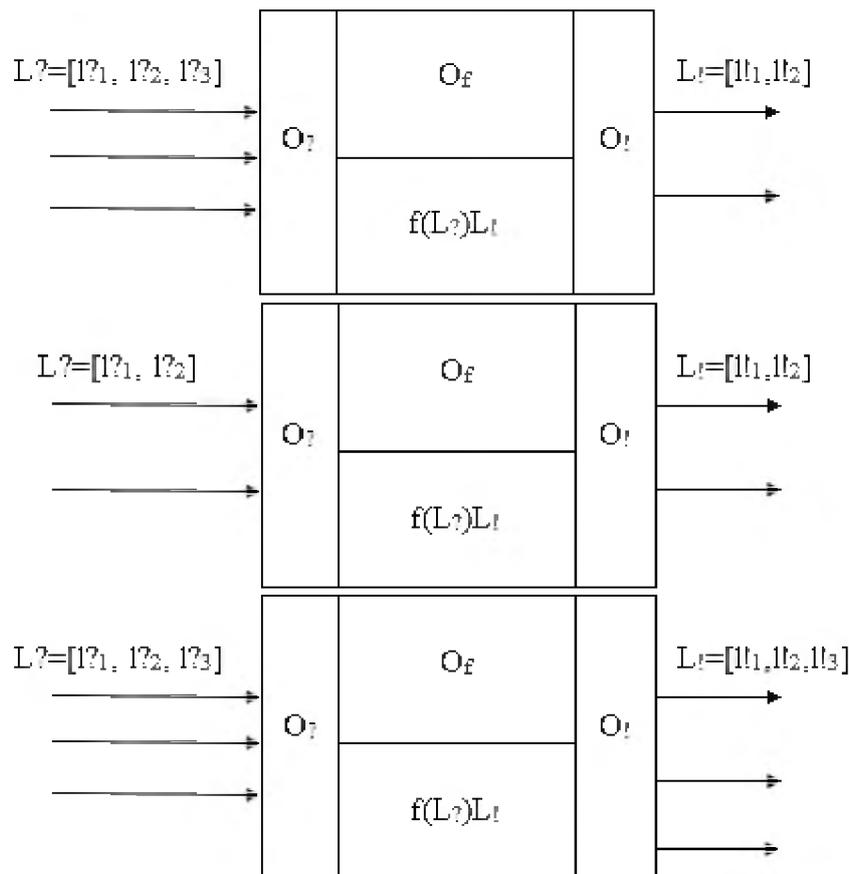


Рис. 2. Графический формализм абстрактной библиотеки узловых объектов  
Fig. 2. Graphical formalism of the abstract library of nodal objects

Очевидно, что для пополнения библиотеки и ее дальнейшего использования, необходимо рассмотреть как минимум две операции на системно-объектных моделях: добавление узлового объекта в библиотеку – экспорт элемента и импорт узлового объекта из библиотеки [9]. Для описания данных операций, исчерпывающим набором параметров являются интерфейсы импортируемых и экспортируемых узловых объектов. Внутренняя организация таких объектов (функция и объект системы) для рассматриваемых операций не имеет значения. Из рисунка 2 видно, что для каждого отдельного узлового объекта интерфейсом является его идентификатор, то есть имя и наборы входных и выходных потоковых объектов с учетом их структур и типов полей. Рассмотрим в качестве примера узловой объект со структурой интерфейсных связей, как показано на рисунке 2 – первый элемент. Формальный вид данного узлового объекта представлен ниже:

$$s_n = [ L? = \{ l?_1, l?_2, l?_3 \}, L! = \{ l!_1, l!_2 \}; f(L?)L!; O?, O!, O_f ] \quad (6)$$

Тогда, интерфейс узлового объекта соответствует структурной характеристике системы  $U$  из выражения 3, что соответствует основным положениям методологии «Узел-Функция-Объект». Однако, помимо структурной составляющей интерфейса узлового объекта, в случае импорта и экспорта элементов, важную роль имеет типовая структура интерфейсных потоковых объектов, т.е. необходимо учитывать типы данных полей потоковых объектов, составляющих интерфейсную часть узлового объекта [10]. Таким образом, если в узловом объекте, описанном в выражении 5, интерфейсные потоковые объекты имеют следующую структуру:

- $l?_1=(r_1, r_2)$
- $l?_2=(r_1)$
- $l?_3=(r_1)$
- $l!_1=(r_1, r_2, r_3)$
- $l!_2=(r_1, r_2)$ ,

тогда интерфейс узлового объекта запишем в следующем виде:

$$U_s = \begin{cases} L? = \begin{cases} l?_1 = (r_1, r_2) \\ l?_2 = (r_1) \\ l?_3 = (r_1) \end{cases} \\ L! = \begin{cases} l!_1 = (r_1, r_2, r_3) \\ l!_2 = (r_1, r_2) \end{cases} \end{cases} \quad (7)$$

В общем виде выражение 7 можно записать следующим образом:

$$U_s = \begin{cases} L? = \begin{cases} l?_1 = (r_1, \dots, r_{i_1}) \\ \dots \\ l?_n = (r_1, \dots, r_{i_n}) \end{cases} \\ L! = \begin{cases} l!_1 = (r_1, \dots, r_{j_1}) \\ \dots \\ l!_m = (r_1, \dots, r_{j_m}) \end{cases} \end{cases} \quad (8)$$

Рассмотрим подробнее операцию импорта узлового объекта  $s$  из системно-объектной модели  $M$  в библиотеку  $S$ . Пусть дана иерархия потоковых объектов модели, содержащая три вещественных потоковых объекта со своими полями следующего вида:

$$L_M = [l_1^v = \{r_1^1, r_2^1\}, l_2^v = \{r_1^2, r_2^2, r_3^2\}, l_3^v = \{r_1^3\}] \quad (9)$$

Множество потоковых объектов модели, описанное выражением 9, имеет следующий вид:

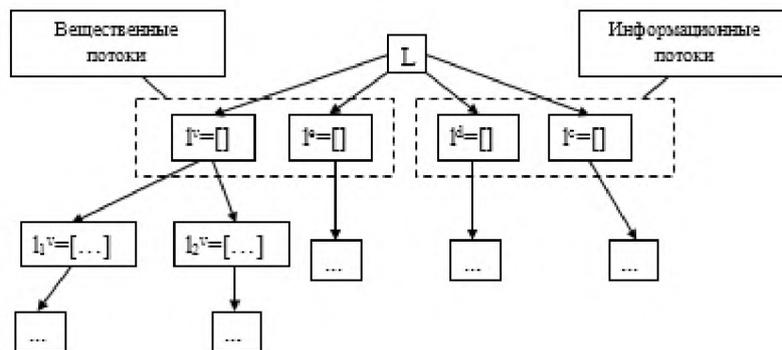


Рис. 3. Множество потоковых объектов системы в виде иерархии  
Fig. 3. Set of stream objects of the system in the form of hierarchy

Также пусть дана соответствующая системно-объектная модель  $M=(L,S)$ , причем множество узловых объектов имеет следующий вид:

$$\begin{aligned} S &= [s_1=(L?=\emptyset, L!={l!_1}); f(L?)L!; O?, O!, Of), \\ s_2 &= (L?=\{l?_1, l?_3\}, L!={l!_2}); f(L?)L!; O?, O!, Of), \\ s_3 &= (L?=\{l?_2\}, L!={l!_3}); f(L?)L!; O?, O!, Of) \end{aligned} \quad (10)$$

Множество потоковых объектов имеет следующий вид:

$$L=[l_1=\{s_1,s_2\}, l_2=\{s_2,s_3\}, l_3=\{s_3,s_2\}] \quad (11)$$

Графически пример описанный выражениями 9, 10 и 11 имеет следующий вид:

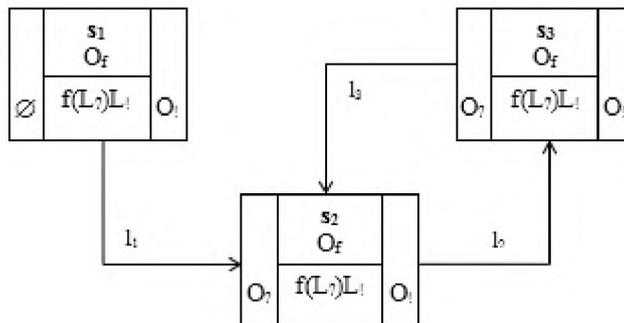


Рис. 4. Пример системно-объектной модели  
Fig. 4. An example of a system-object model

Тогда, для импорта узлового объекта  $s_2$  системно-объектной модели  $M$  в библиотеку  $L_M$  опишем оператор в следующем формате:

$$L_M^* = \text{import}(M, s_2, L_M) \rightarrow (L^* = \emptyset; S^* = [s_1 = (L^? = \emptyset, L! = \{l!_1\}; f(L^?)L!; O^?, O!, Of), s_3 = (L^? = \{l^?_2\}, L! = \{l!_3\}; f(L^?)L!; O^?, O!, Of)]; L_M^* = [s_2 = (L^? = \{l^?_1, l^?_3\}, L! = \{l!_2\}; f(L^?)L!; O^?, O!, Of)]$$

Результатом данной операции является библиотека  $L_M^*$ , пополненная узловым объектом  $s_2$  и системно-объектная модель  $M^*(\emptyset, S^*)$  следующего вида:



Рис. 5. Результат импорта узлового объекта в библиотеку  
Fig. 5. Result of importing a node object to the library

Как было отмечено выше, для работы с узловым объектом, помещаемым в библиотеку, необходимо также учитывать его интерфейс. Для рассматриваемого примера, интерфейс объекта  $s_2$  примет следующий вид:

$$U_{s_2} = \begin{cases} L^? = \begin{cases} l^?_1 = (r_1^1, r_2^1) \\ l^?_3 = (r_1^3) \end{cases} \\ L! = \{l!_2 = (r_1^2, r_2^2, r_3^2)\} \end{cases} \quad (12)$$

Следует отметить, что описанная операция импорта узлового объекта в примере предполагает извлечение вышеупомянутого элемента из модели в библиотеку, однако же, в процессе моделирования пользователь может сохранить узловой объект в библиотеку путем копирования, то есть без удаления первого из исходной модели и освобождения соответствующих потоковых объектов. В таком случае исходная модель из которой экспортируется элемент в библиотеку остается в неизменном виде, как показано на рисунке 4.

Далее рассмотрим операцию экспорта узлового объекта  $s_2$  из библиотеки  $L_M^*$  в системно-объектную модель  $M^*(\emptyset, S^*)$ . Сформулируем описание оператора экспорта в следующем виде:

$M^* = \text{export}(M, s_2, L_M^*) \rightarrow (L^* = \emptyset; S^* = [s_1 = (L^* = \emptyset, L^* = \{1!_1\}); f(L^*)L^*; O?, O!, Of), s_2 = (L^* = \{1?_1, 1?_3\}, L^* = \{1!_2\}); f(L^*)L^*; O?, O!, Of), s_3 = (L^* = \{1?_2\}, L^* = \{1!_3\}); f(L^*)L^*; O?, O!, Of]; L_M^* = [s_2 = (L^* = \{1?_1, 1?_3\}, L^* = \{1!_2\}); f(L^*)L^*; O?, O!, Of].$

Как видно из описания операции экспорта, соответствующий узловой объект был добавлен в модель, однако его необходимо соединить с существующими узловыми объектами модель. Для этого применим операцию соединения двух узловых объектов, описанных в работе [2]. Тогда получим модель в исходном виде, как представлено в выражении 10,11.

Для подбора элементов из библиотеки имеется возможность анализировать специальный количественный показатель «мера системности».

Для описания алгоритма расчета меры системности для узлового объекта с одним входом и одним выходом, как представлено на рисунке 1, введем следующие обозначения: FRFSs – (область) множество требуемых функциональных состояний узлового объекта [2] и FPSs – (область) множество возможных функциональных состояний [2]. Причем элементы данных множеств имеют вид:

$$a_s = [A^{11}, A^{12}], \quad (13)$$

где:

$A^{11}$  – состояние входного потокового объекта  $l_1$ ;

$A^{12}$  – состояние исходящего потокового объекта  $l_2$ .

Переменная MOS – представляет собой искомый коэффициент соответствия области возможных состояний и области требуемых состояний. Алгоритм расчета меры системности будет представлять собою поочередное сравнение элементов множества FRFSs с элементами множества FPSs.

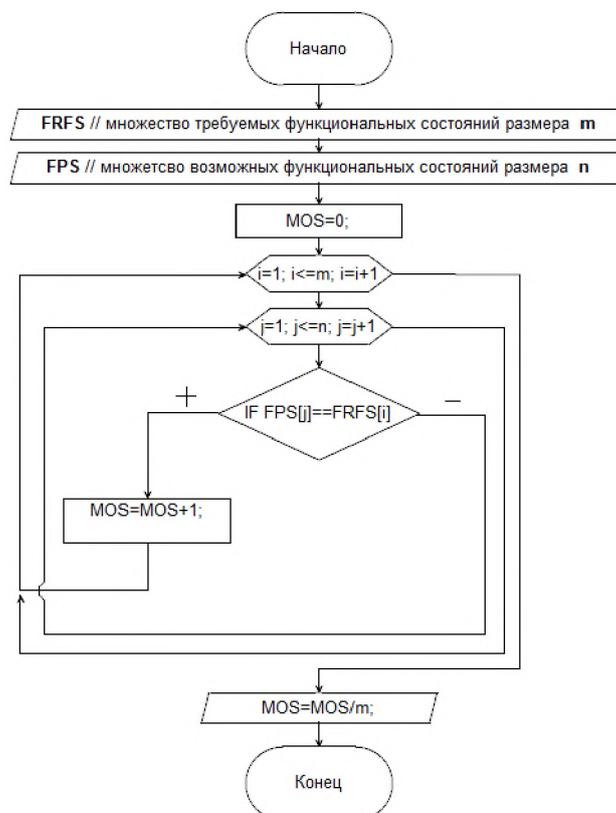


Рис. 6. Алгоритм расчета меры системности узлового объекта  
Fig. 6. The algorithm for calculating the measure of the systemic nodal object

Как видно из рисунка 6, элемент из множества требуемых состояний поочередно сравнивается с элементами множества возможных состояний. Если идентичное состояние найдено, тогда переменная MOS увеличивается на единицу и далее осуществляется переход к новому требуемому состоянию, так как далее сравнивать текущее требуемое состояние не имеет смысла, оно уже найдено. По истечении работы внешнего цикла, в переменной MOS будет содержаться количество найденных требуемых функциональных состояний из множества возможных, после чего поделив данное количество на общее количество требуемых состояний, получим значение коэффициента системности в промежутке от нуля до единицы, причем, чем ближе коэффициент к единице, тем система более соответствует запросу надсистемы. Данный алгоритм будет работать для всех видов узловых объектов, главная проблема будет состоять в адекватном формировании множества требуемых функциональных состояний узлового объекта.

Как видно из описания алгоритма, представленную числовую характеристику системы можно использовать для экспорта элементов библиотеки в модель и определения наиболее подходящего.

Таким образом, рассмотренные теоретические положения процедур формирования библиотек узловых объектов, позволяю разработать программную подсистему для реализации вышеупомянутых процессов. Это, в свою очередь, позволит в процессе системно-объектного имитационного моделирования использовать ранее смоделированные системы в текущем проекте, что существенно ускорит процесс разработки системно-объектной имитационной модели. В перспективе, планируется программная реализация описанных выше процедур и создание библиотек готовых узловых объектов в различных предметных областях.

*Исследования выполнены при финансовой поддержке грантов Российского фонда фундаментальных исследований № 18-07-00355, 16-07-00193 и 16-07-00460.*

#### Список литературы

1. Zhikharev, A.G., Matorin, S.I., Kuznetsov, A.V., Zherebtsov, S.V., Tchekanov, N.A. To The Problem of the Coefficient Calculus of the Nodal Object in the System-Object Models // Journal of Advanced Research in Dynamical & Control Systems, Vol. 10, 10-Special Issue, 2018. – P. 1813-1817.
2. 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, p. 182-191.
3. S.I. Matorin, A.G. Zhikharev and O.A. Zimovets Object Calculus in the System-Object Method of Knowledge Representation // Scientific and Technical Information Processing, Vol. 45, No. 5, pp. 1–10, 2018.
4. С.И. Маторин, А.Г. Жихарев Учет закономерностей при системно-объектном моделировании организационных знаний // Искусственный интеллект и принятие решений, № 03 - 2018, с. 57-68.
5. С.И. Маторин, А. Г. Жихарев Формализация системно-объектного подхода «Узел-Функция-Объект» // Прикладная информатика, том. 13, № 3(75), с. 124-135, 2018 г.
6. Егоров И.А., Жихарев А.Г., Маторин С.И. К вопросу оптимизации системно-объектных имитационных моделей // Тезисы XIX Всероссийской конференции молодых ученых по математическому моделированию и информационным технологиям, г. Кемерово, 29 октября-02 ноября 2018 г, с. 61-62.
7. Егоров И.А., Маторин С.И., Жихарев А.Г., Зайцев А.Н. Решение задач оптимизации в системно-объектных моделях // Научный результат. Информационные технологии, т. 3, № 2, с. 35-43, 2018 г.
8. С.И. Маторин, А.Г. Жихарев, О.А. Зимовец Обоснование взаимосвязей общесистемных принципов и закономерностей с позиции системно-объектного подхода // Труды Института системного анализа Российской академии наук. - 2017. - №3. - Том 67.
9. 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 2017, 12 (Special issue 6), p. 7859-7864.
10. А.Г. Жихарев, И.А. Егоров, М.Ю. Манзуланич, С.И. Маторин Системно-объектное имитационное моделирование систем массового обслуживания // Сетевой журнал «Научный результат», серия «Информационные технологии», - Вып. № 4(4), 2016 г.

### References

1. Zhikharev, A.G., Matorin, S.I., Kuznetsov, A.V., Zherebtsov, S.V., Tchekanov, N.A. To The Problem of the Coefficient Calculus of the Nodal Object in the System-Object Models // Journal of Advanced Research in Dynamical & Control Systems, Vol. 10, 10-Special Issue, 2018. – P. 1813-1817.
2. 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, p. 182-191.
3. S.I. Matorin, A.G. Zhikharev and O.A. Zimovets Object Calculus in the System–Object Method of Knowledge Representation // Scientific and Technical Information Processing, Vol. 45, No. 5, pp. 1–10, 2018.
4. S.I. Matorin, A.G. Zhikharev Uchet zakonomernostej pri sistemno-ob"ektnom modelirovanii organizacionnyh znanij // Iskusstvennyj intellekt i prinyatie reshenij, № 03 / 2018, s. 57-68.
5. S. I. Matorin, A. G. Zhikharev Formalizaciya sistemno-ob"ektnogo podhoda «Uzel-Funkciya-Ob"ekt» // Prikladnaya informatika, tom. 13, № 3(75), str. 124-135, 2018 g.
6. Egorov I.A., Zhikharev A.G., Matorin S.I. K voprosu optimizacii sistemno-ob"ektnyh imitacionnyh modelej // Tezisy XIX Vserossijskoj konferencii molodyh uchenyh po matematicheskomu modelirovaniyu i informacionnym tekhnologiyam, g. Kemerovo, 29 oktyabrya-02 noyabrya 2018 g, str. 61-62.
7. Egorov I.A., Matorin S.I., Zhikharev A.G., Zajcev A.N. Reshenie zadach optimizacii v sistemno-ob"ektnyh modelyah // Nauchnyj rezul'tat. Informacionnye tekhnologii, t. 3, № 2, str. 35-43, 2018 g.
8. S.I. Matorin, A.G. Zhikharev, O.A. Zimovec Obosnovanie vzaimosvyazej obshchesistemnyh principov i zakonomernostej s pozicii sistemno-ob"ektnogo podhoda // Trudy Instituta sistemnogo analiza Rossijskoj akademii nauk. - 2017. - №3. - Tom 67.
9. 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 2017, 12 (Special issue 6), p. 7859-7864.
10. A.G. Zhikharev, I.A. Egorov, M.YU. Manzulanich, S.I. Matorin Sistemno-ob"ektnoe imitacionnoe modelirovanie sistem massovogo obsluzhivaniya // Setevoj zhurnal «Nauchnyj rezul'tat», seriya «Informacionnye tekhnologii», - Vyp. № 4(4), 2016 g.

**Бузов Андрей Анатольевич**, аспирант

**Жихарев Александр Геннадиевич**, кандидат технических наук, доцент кафедры информационных и робототехнических систем

**Маторин Сергей Игоревич**, доктор технических наук, профессор, профессор кафедры информационных и робототехнических систем

**Buzov Andrey Anatolyevich**, graduate student

**Zhikharev Alexander Gennadievich**, Candidate of Technical Sciences, Associate Professor at the Department of Information and Robotic Systems

**Matorin Sergey Igorevich**, Doctor of Technical Sciences, Professor, Professor at the Department of Information and Robotic Systems