

Общероссийский математический портал

А. Г. Жихарев, Системно-объектный подход к моделированию организационных знаний, учитывающий общесистемные закономерности, *Искусственный интеллект и принятие решений*, 2021, выпуск 3, 33–44

DOI: 10.14357/20718594210303

Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением

<http://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 3.148.108.174

26 декабря 2024 г., 20:31:20



Системно-объектный подход к моделированию организационных знаний, учитывающий общесистемные закономерности *

А. Г. Жихарев

Белгородский государственный технологический университет им. В.Г. Шухова, г. Белгород, Россия

Аннотация. В работе рассмотрены способы учета общесистемных принципов и закономерностей при построении системно-объектных моделей организационных знаний. Предложена модификация оригинального формального аппарата исчисления систем как функциональных объектов. Применение полученных результатов позволит, в перспективе, автоматизировать и усовершенствовать процедуры системно-объектного моделирования при проектировании сложных организационных и информационных систем с целью повышения эффективности управления ими.

Ключевые слова: общесистемные закономерности, системно-объектная модель, организационные знания, исчисление систем как функциональных объектов.

DOI 10.14357/20718594210303

Введение

Управление современными организационными системами требует наличия информационно-аналитического обеспечения процессов принятия решений. В работах [1, 2] показано, что снижение качества функционирования организационных и информационных систем является следствием несоблюдения общесистемных принципов и закономерностей при их проектировании и разработке. Для эффективного управления сложными системами необходимо информационно-аналитическое обеспечение, основанное на системном подходе [3].

В данной работе предлагается использовать системно-объектный подход «Узел-Функция-Объект» [4], технологию графоаналитического имитационного моделирования [5, 6] и системно-объектный метод представления организа-

ционных знаний [7]. Показано, что системно-объектный подход, в отличие от других системных подходов, учитывает общесистемные принципы и закономерности [8]. Программный инструмент UFOModeler позволяет автоматизировать процедуры визуального графоаналитического имитационного системно-объектного моделирования [9], но не гарантирует соответствие системно-объектной модели общесистемным закономерностям. Поэтому весьма актуальна разработка методов и алгоритмов моделирования систем, которые учитывают такие закономерности.

1. Учет принципа коммуникативности

Рассмотрим, как можно учесть при системно-объектном моделировании *принцип коммуникативности* [4]. Согласно этому принципу,

* Работа поддержана Российским фондом фундаментальных исследований (проекты 19-07-00111, 19-07-00290).

✉ Жихарев Александр Геннадиевич. E-mail: zhikharev@bsu.edu.ru

любая система должна быть связана множеством коммуникаций с окружающей средой (другими системами), иначе существование системы не имеет смысла. На Рис. 1 представлен фрагмент системно-объектной модели, которая не соответствует принципу коммуникативности. Здесь использованы графические формализмы, предложенные в работе [10].

Узловой объект (например, объект s_1 – показан слева) состоит из четырех областей. В прямоугольной области, слева и справа отображаются объектные характеристики узла (интерфейсные порты узла), касающиеся параметров входа и выхода. Из рисунка видно, что узловой объект s_1 не связан ни с одним другим узловым объектом, что, естественно, противоречит принципу коммуникативности. Прямоугольная область, расположенная сверху, содержит идентификатор узлового объекта и символ множества его функциональных (передаточных) объектных характеристик. Прямоугольная область, расположенная внизу, содержит описание функции узлового объекта. Поточковые объекты (например, связи I_1 и I_2) отображаются в виде ориентированных связующих линий с соответствующими идентификаторами. Идентифицировать такие узловые объекты в терминах исчисления систем как функциональных объектов [4] можно в автоматическом режиме, применяя оператор контекста, введенный в работе [11]. Этот оператор возвращает перечень связей системно-объектной модели, у которых рассматриваемый узловой объект, переданный оператору контекста в качестве параметра, является «входом» или «выходом». Таким образом, после применения оператора контекста, например, к узло-

вому объекту s_1 , получим пустое множество: $context(s_1)=\emptyset$.

В соответствии с правилами исчисления систем как функциональных объектов любая системно-объектная модель состоит из трех компонент:

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

где L – множество потоковых объектов модели (иерархия связей системы), где каждый потоковый объект представлен набором именованных полей, содержащих значения потокового объекта; S – множество узловых объектов модели (подсистем), где узловой объект представлен в виде:

$$s = [(L_?, L!); f(L_?)L!; (O_?, O!, O_f)]. \tag{2}$$

Узловой объект содержит ряд полей и единственный метод. Полями узлового объекта являются первый и третий компоненты из выражения (2). $L_?, L!$ – множества входящих и исходящих потоковых объектов (порты моделируемой системы). По аналогии с исчислением процессов [12], множества входящих потоковых объектов обозначаются идентификатором с индексом «?», а исходящих – «!». Причем множества входящих и исходящих потоковых объектов являются подмножествами L . Объектные характеристики системы представлены тремя множествами – $O_?, O!, O_f$, где первое содержит перечень параметров входов, второе – параметры выходов, третье – параметры, характеризующие объект, который участвует в реализации метода $f(L_?)L!$. Метод узлового объекта содержит описание процесса преобразования входящих потоковых объектов в исходящие. C – множество связей узловых объектов. Элемент множества характеризует единственную связь

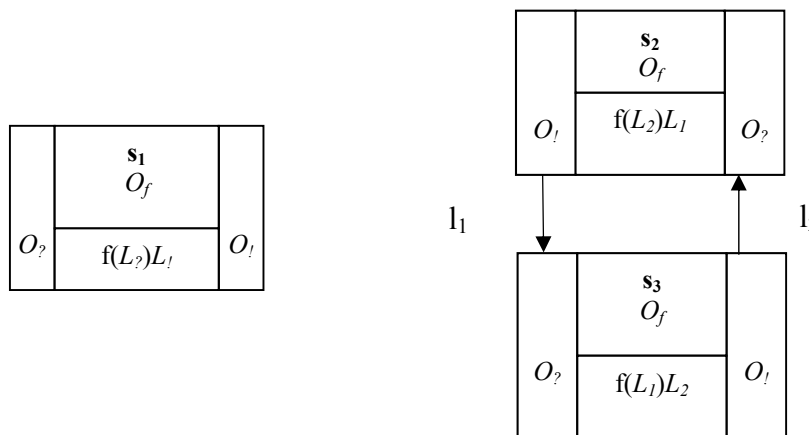


Рис. 1. Фрагмент абстрактной системно-объектной модели

двух узловых объектов. Он содержит узловой объект – источник связи s_{out} , узловой объект – получатель связи s_{in} и тип связи l . Проводя аналогию с теорией графов, элемент множества C есть направленное ребро, соединяющее две вершины графа. Необходимо отметить, что связь характеризует обмен конкретными ресурсами двух узловых объектов, когда структурные характеристики узлового объекта характеризуют его интерфейс для взаимодействия с внешней средой. Формально, связь двух узловых объектов будем представлять в форме трех связанных элементов:

$$c = \{s_{out}, s_{in}, l\} : s_{out} \in S, s_{in} \in S, l \in L. \quad (3)$$

Если интерфейс узлового объекта не содержит портов, ни входящих, ни исходящих, тогда он, по определению, не может быть связан с другими узловыми объектами. В таком случае принцип коммуникативности можно представить в следующем формальном виде:

$$\forall s \in S : \nexists s. U = \emptyset. \quad (4)$$

Далее при обращении к полям или методам узловых объектов будем использовать символ «.» по аналогии с правилами обращения к полям и методам в теории объектов [13]. Таким образом, запись вида « $s.U$ » есть идентификатор множества структурных характеристик узлового объекта (порты системы).

Связи узлового объекта с внешней средой существуют только при наличии соответствующих интерфейсных характеристик (портов системы) $U = (L_p, L_f)$. Тогда соответствие моделируемой системы принципу коммуникативности определяется отсутствием в модели узло-

вых объектов, для которых множество интерфейсных потоковых объектов является пустым. Однако возможна ситуация, когда, например, у узлового объекта имеются входные потоковые объекты и нет выходных потоковых объектов и наоборот. Такая ситуация также противоречит принципу коммуникативности. Поэтому критерий выполнения принципа коммуникативности можно формально уточнить и записать:

$$\forall s \in S : \nexists s. L_p = \emptyset, \nexists s. L_f = \emptyset. \quad (5)$$

Рассмотрим графоаналитическое представление системно-объектной модели, соответствующей принципу коммуникативности (Рис. 2).

Из рисунка видно, что все узлы модели соединены связями. Используем для формальной записи введенный в работе [14] оператор связывания узловых объектов: $s_i \xrightarrow{1} s_j$, согласно которому узловой объект s_i связывается с узловым объектом s_j с помощью потокового объекта l . Тем самым множество C пополняется новым элементом вида: $c = \{s_i, s_j, l\}$.

Таким образом, учет принципа коммуникативности в системно-объектной модели заключается в связывании имеющихся узловых объектов с другими. Отметим, однако, что процедура связывания узлового объекта с другими трудно поддается автоматизации. Поэтому программный инструментальный должен обеспечивать возможность привязки каждого конкретного узлового объекта к имеющимся свободным портам других узловых объектов. Алгоритм учета принципа коммуникативности в системно-объектной модели состоит из следующих шагов (Рис. 3).

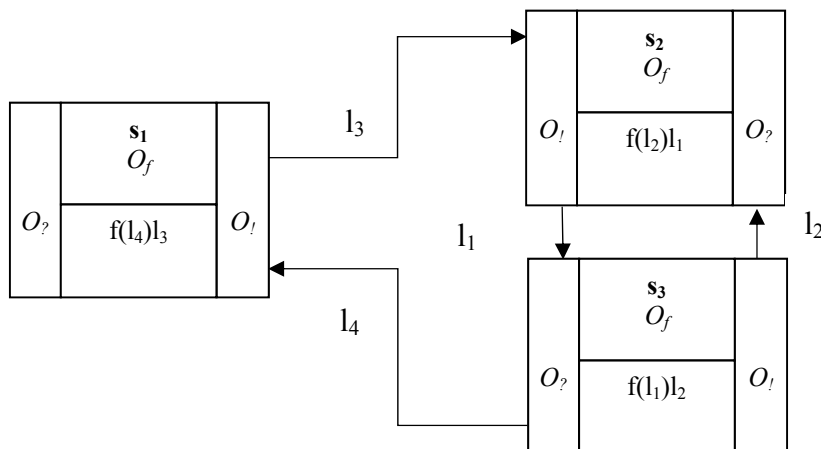


Рис. 2. Фрагмент системно-объектной модели, учитывающей принцип коммуникативности

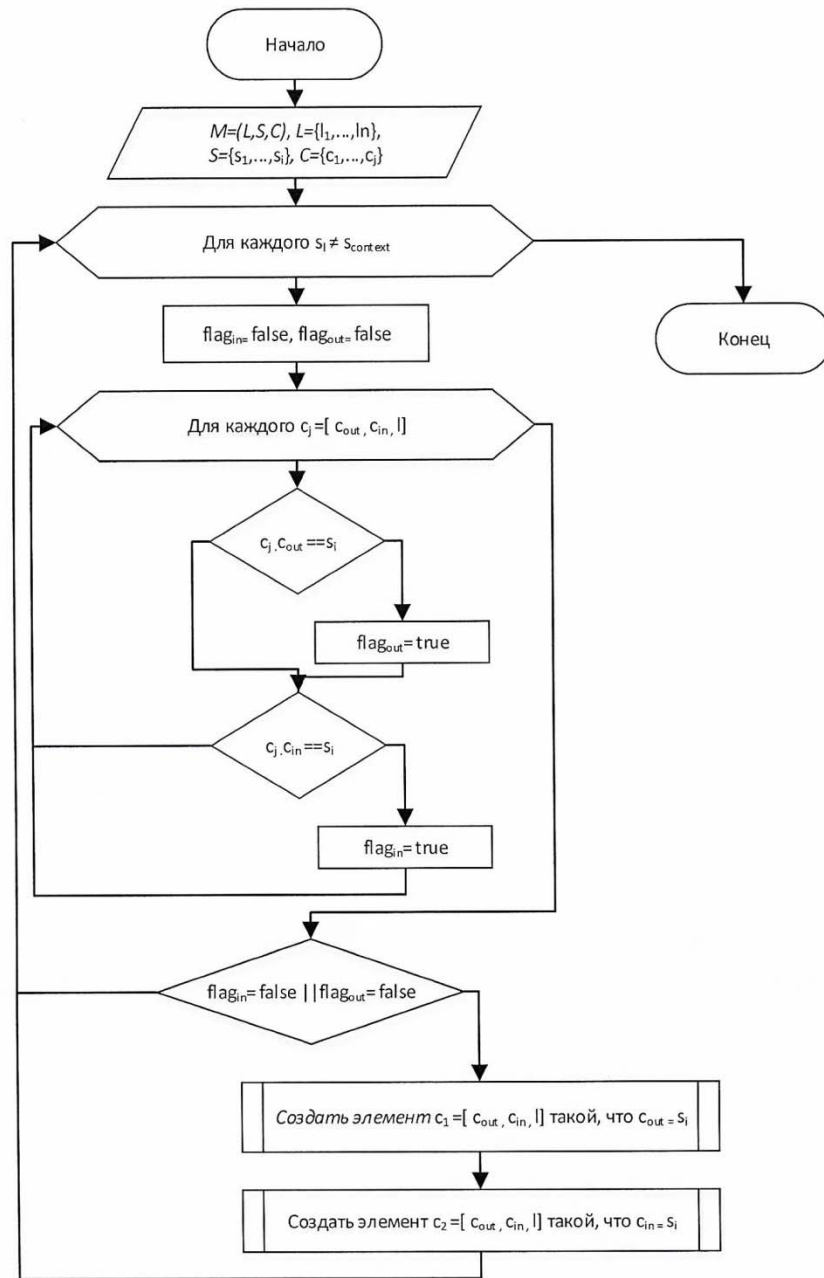


Рис. 3. Алгоритм учета принципа коммуникативности в системно-объектной модели

1. Ввести модель в форме $M=(L,S,C)$, где: $L=\{l_1, \dots, l_n\}$; $S=\{s_1, \dots, s_i\}$; $s_i = [(L?, L?); f(L?)L?; (O?, O?, O?)]$; $C=\{c_1, \dots, c_j\}$; $c_j=\{s_{out}, s_{in}, l\}$; $s_{out} \in S$, $s_{in} \in S$, $l \in L$.

2. Для каждого элемента множества S за исключением контекстного узлового объекта $s_{context}$ просмотреть все элементы множества связей модели C . Если узел, из которого выходит связь $c_j \cdot s_{out}$ соответствует анализируемому в текущий момент узлового объекту s_i , установить специальный счетчик $flag_{out}$ в значение истины, указы-

вающий, что рассматриваемый узел связан с другим исходящей связью. Если узел, в который входит связь $c_j \cdot s_{in}$ соответствует анализируемому в текущий момент узлового объекту s_i , установить специальный счетчик $flag_{in}$ в значение истины, указывающий, что рассматриваемый узел связан с другим входящей связью.

3. Проверить значения всех счетчиков. Если хотя бы один из них остался в значении ложь, то связать текущий узел с другими узлами, после чего перейти к следующему узлового объекту.

4. Алгоритм завершает свою работу, когда будут проанализированы все узловые объекты рассматриваемой модели.

2. Учет принципа иерархичности

Рассмотрим, как можно учесть при системно-объектном моделировании *принцип иерархичности* [4], который вытекает из принципа коммуникативности. Выделим два аспекта, соблюдение которых обеспечивает выполнение указанного принципа. Во-первых, любую систему необходимо рассматривать как иерархию подсистем, во-вторых, систему на любом уровне необходимо рассматривать как часть системы более высокого уровня (надсистемы).

С формальной точки зрения, «вложенность» узловых объектов учитывается вложенностью функций, т.е. когда функция узлового объекта реализуется за счет функций других (вложенных) узловых объектов. При системно-объектном имитационном моделировании [4] функция любого узлового объекта может быть представлена в двух вариантах:

1. Функция узлового объекта при достижении требуемого уровня детализации реализуется с помощью языка описания функциональных узлов УФО-скрипт [10], на котором прописываются процедуры преобразования полей входных потоковых объектов в поля выходных потоковых объектов.

2. Функция узлового объекта реализуется за счет функций других (вложенных) узловых объектов (реализация вложенности узловых объектов).

Положения системно-объектного подхода гарантируют соблюдение принципа иерархичности, так как в любой модели всегда существует узловой объект первого уровня, связанный с контекстом модели. Для автоматизации системно-объектного анализа и имитационного моделирования необходимо разработать алгоритм проверки модели на соответствие критерию выполнения принципа для последующей верификации модели.

Формальный критерий соответствия принципу иерархичности, используя обозначение вложенности функций узловых объектов [11], можно записать в виде:

$$\exists s_i \in S : s_i.f = \begin{cases} s_1.f \\ \dots \\ s_k.f \end{cases} \quad (6)$$

Выражение (6) показывает, что для соответствия принципу иерархичности необходимо существование такого узлового объекта s_i , функция которого реализуется за счет функций других узловых объектов – его подсистем.

3. Учет принципа моноцентризма

Далее рассмотрим механизм учета при системно-объектном моделировании *принципа моноцентризма* [4], являющегося следствием принципа иерархичности. По определению Богданова согласно данному принципу устойчивая система обладает одним центром, а полицентричность приводит к нарушению процессов координации, что в перспективе обуславливает потерю целостности.

Согласно положениям системно-объектного подхода «Узел-Функция-Объект», для любой графоаналитической модели применяется базовая иерархия связей, в нашем случае – потоковых объектов. При этом базовая иерархия связей имеет ряд predetermined связей следующих видов: класс связей V , по которым «текут» вещественные ресурсы; класс связей E , по которым «текут» энергетические ресурсы; класс связей D , по которым передаются данные; класс связей C , по которым передается управляющая информация. Обозначим predetermined потоковые объекты так: L^v – потоковый объект родитель, представляющий класс вещественных объектов; L^e – потоковый объект родитель, представляющий класс энергетических объектов; L^d – потоковый объект родитель, представляющий класс информационных объектов; L^c – потоковый объект родитель, представляющий класс информационных управляющих объектов.

Базовые потоковые объекты не имеют полей, следовательно, имеют вид:

$$L = []. \quad (7)$$

Все остальные элементы множества L , не являющиеся базовыми элементами, являются их наследниками, причем каждый дочерний потоковый объект наследует поля потокового объекта – родителя. Таким образом, для конкретной модели системы в терминах исчисления функциональных объектов, множество потоковых объектов L можно представить как иерархию (Рис. 4).

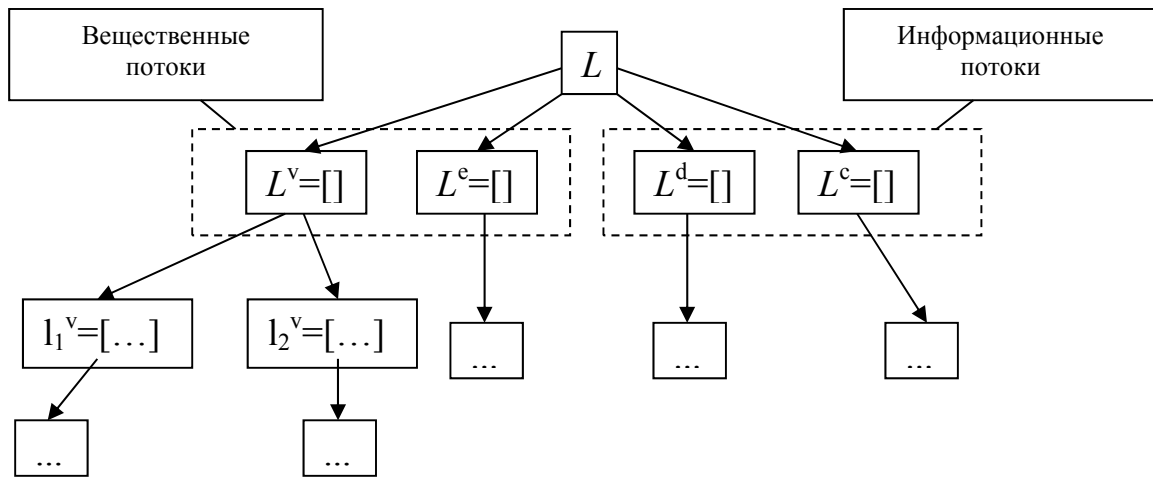


Рис. 4. Множество потоковых объектов системы в виде иерархии

Используя приведенные выше обозначения классов потоковых объектов, сформулируем критерий соответствия модели принципу моноцентризма: все узловые объекты модели должны иметь входящие управляющие потоковые объекты. Также должна быть исключена ситуация, когда узловым объектом управляют несколько других узловых объектов, то есть имеет место несколько входящих управляющих связей из разных источников. Принцип коммуникативности при этом выступает в качестве необходимого условия соблюдения рассматриваемого принципа. Критерий выполнения принципа моноцентризма можно формально записать так:

$$\forall s_i \in S \exists c = [s_{out}, s_{in}, l], s_i \equiv s_{in}, l \in L^c. \quad (8)$$

Выражение (8) показывает, что для соответствия принципу моноцентризма, каждый узловой объект должен обладать входящей связью управляющего типа. В то же время не должно быть ситуаций, когда одним узловым объектом управляют разные узловые объекты. Это ограничение можно представить в виде:

$$\forall s_i \in S \nexists c_1, c_2: c_1 \cdot s_{in} \equiv s_i, c_2 \cdot s_{in} \equiv s_i, c_1 \cdot s_{out} \neq c_2 \cdot s_{out}, c_1 \cdot l \in L^c, c_2 \cdot l \in L^c. \quad (9)$$

Алгоритм учета принципа моноцентризма заключается в устранении потоковых связей, которые нарушают процесс координации подчиненных объектов и включают в себя следующие шаги (Рис. 5):

1. Ввести модель в форме $M=(L,S,C)$, где: $L=\{l_1, \dots, l_n\}$; $S=\{s_1, \dots, s_i\}$; $s_i = [(L?, L?); f(L?)L?]$

$(O?, O?, O?]$; $C=\{c_1, \dots, c_j\}$; $c_j=\{s_{out}, s_{in}, l\}$; $s_{out} \in S, s_{in} \in S, l \in L$.

2. Для каждого элемента множества S за исключением контекстного узлового объекта $S_{context}$ просмотреть все элементы множества связей системно-объектной модели C . Каждую связь анализировать на предмет ее вида, если она является управляющей и входит в текущий рассматриваемый узловой объект. Тогда увеличить на единицу счетчик входящих управляющих связей узлового объекта $flag_{control}$.

3. Проверить два условия для каждого узлового объекта. Если у него нет входящих связей по управлению, т.е. счетчик равен нулю, тогда предложить пользователю добавить управляющую связь. Если же счетчик принял значение больше единицы и управляющие связи приходят от разных узлов, предложить удалить одну из управляющих связей. Перейти к следующему узловому объекту.

4. Алгоритм завершает свою работу, когда будут проанализированы все узловые объекты рассматриваемой модели.

4. Учет принципа обратной связи

Рассмотрим механизм учета *принципа обратной связи* [4], согласно которому устойчивость в сложных динамических системах достигается за счет замыкания петель обратных связей. Здесь речь идет об управляемых узловых объектах, от которых к субъекту управления должны быть обратные связи, причем они могут быть разных типов.

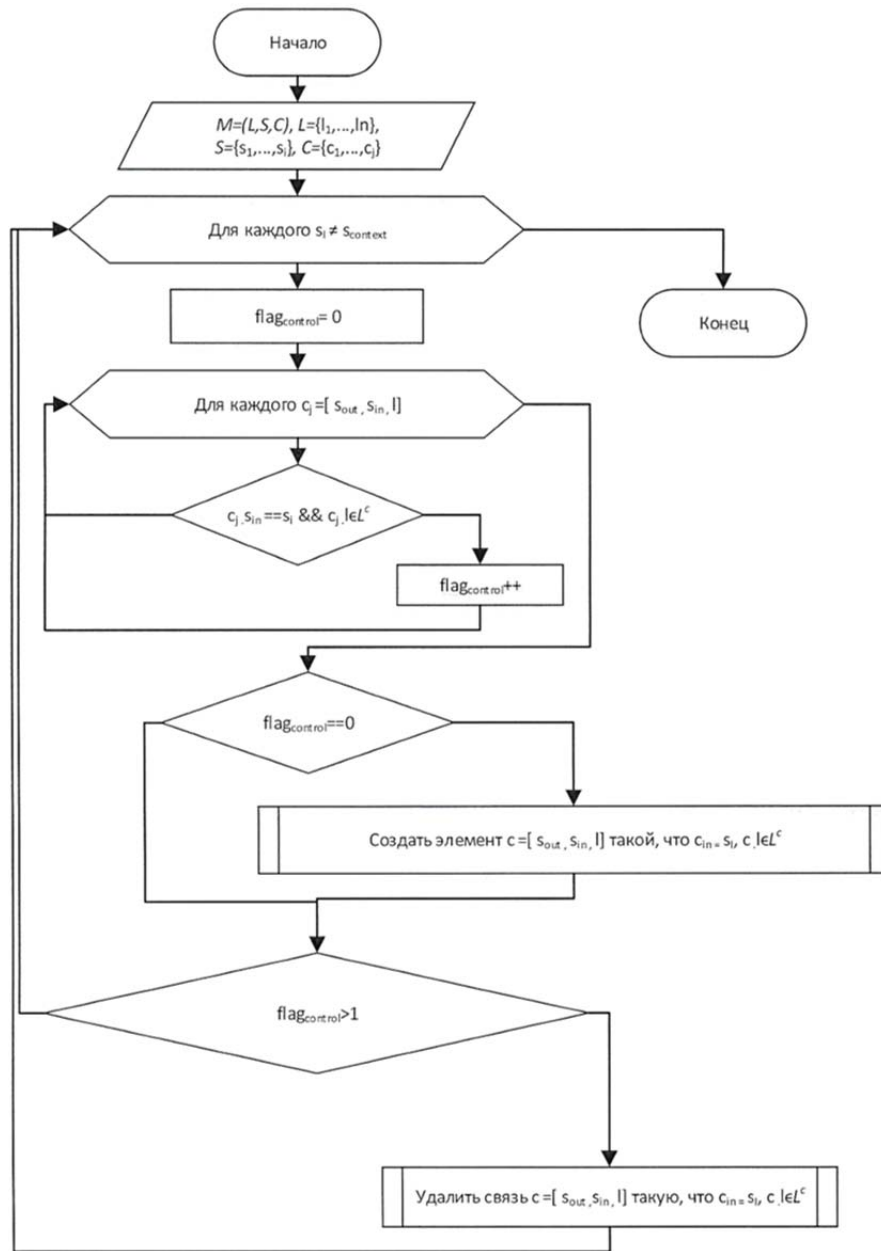


Рис. 5. Алгоритм учета принципа моноцентризма в системно-объектной модели

Формальный критерий данного принципа представим в виде выражения:

$$\exists s_i \in S, s_j \in S: l_m \in s_i \cdot L_?, l_m \in s_j \cdot L_?, l_n \in s_j \cdot L_?, l_n \in s_i \cdot L_? \quad (10)$$

Принцип обратной связи можно также выразить через множество связей систем. Для каждого узлового объекта, у которого есть входящая связь по управлению, должна существовать обратная связь к субъекту управления. Это утверждение можно записать в виде:

$$\forall s_i \in S \exists c = [s_{out}, s_{in}, l], s_i \equiv s_{in}, l \in L^c \exists c = [s_{out}, s_{in}, l], s_i \equiv s_{out} \quad (11)$$

Следовательно, механизм учета принципа обратной связи заключается в присоединении соответствующими видами связей узловых объектов, используя оператор связывания [14].

Для реализации алгоритма учета принципа обратной связи, необходимо выполнить алгоритм учета принципа моноцентризма, который гарантирует наличие управляющих связей для

каждого узлового объекта. Алгоритм модификации модели с учетом принципа обратной связи можно представить в виде следующих шагов (Рис. 6):

1. Ввести модель в форме $M=(L,S,C)$, где: $L=\{l_1, \dots, l_n\}$; $S=\{s_1, \dots, s_i\}$; $s_i = [L_i, L_i]; f(L_i) L_i;$ (O_i, O_i, O_i) ; $C=\{c_1, \dots, c_j\}$; $c_j=\{s_{out}, s_{in}, I\}$; $s_{out} \in S$, $s_{in} \in S$, $I \in L$.

2. Для каждого элемента множества S за исключением контекстного узлового объекта $S_{context}$ просмотреть все элементы множества связей системно-объектной модели C . Каждую

связь проанализировать на предмет ее вида, если связь является управляющей и входит в текущий рассматриваемый узловой объект, тогда запомнить текущую связь и продолжить поиск по множеству C обратных связей.

2. Если обратная связь, соответствующая ранее сохраненной управляющей связи, найдена, перейти к следующей связи узлового объекта, иначе предложить создать обратную связь.

3. Алгоритм завершает свою работу, когда будут проанализированы все узловые объекты рассматриваемой модели.

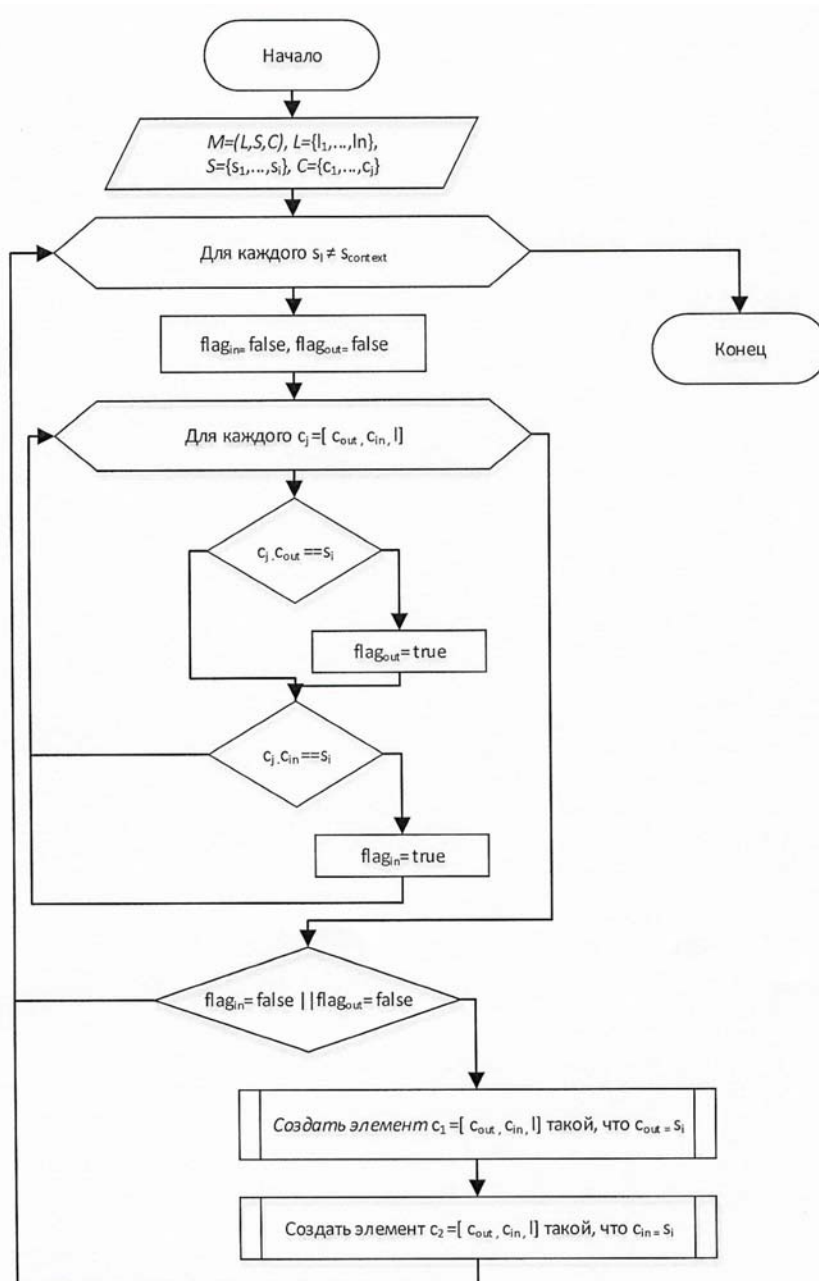


Рис. 6. Алгоритм учета принципа обратной связи в системно-объектной модели

5. Учет закона необходимого разнообразия

В работах [4, 10] предложены основные положения исчисления систем как функциональных объектов, которые позволили разработать алгоритм расчета меры системности MOS. Для любого узлового объекта мера системности вычисляется путем сравнения массива его возможных функциональных состояний (FPS) и массива требуемых функциональных состояний (FRFS): $MOS = FRFS / FPS$. Мера системности равна единице в случае, когда для любого требуемого функционального состояния существует возможное состояние узлового объекта. Чем больше таких совпадений, тем ближе мера системности к единице. В свою очередь функциональные состояния, как требуемые, так и возможные, могут быть представлены в виде векторов значений полей потоковых объектов в заданный момент времени. Из векторов состояний формируются матрицы требуемых возможных функциональных состояний соответственно.

Рассмотрим, как можно учесть при системно-объектном моделировании **закон необходимого разнообразия** [4]. Названный закон будет выполняться в том случае, когда число требуемых функциональных состояний меньше чем число возможных, т.е. $n > k$, где n – число возможных функциональных состояний (FPS), k – число требуемых функциональных состояний (FRFS). Когда коэффициент системности равен единице, в матрице возможных состояний содержатся все столбцы из матрицы требуемых функциональных состояний. Это гарантирует выполнение закона необходимого разнообразия. Поэтому учитывать этот закон необходимо, когда коэффициент системности меньше единицы. В таком случае необходимо предложить пользователю добавить потоковые объекты для анализируемого узлового объекта, что приведет к модификации его функции и сформируются новые возможные функциональные состояния системы.

Алгоритм механизма учета закона необходимого разнообразия представлен на Рис. 7. Его можно описать следующими шагами:

1. Ввести модель в форме $M=(L,S,C)$, где $L=\{l_1, \dots, l_n\}$; $S=\{s_1, \dots, s_i\}$; $s_i = [(L_i, L_i); f(L_i)L_i; (O_i, O_i, O_i)]$; $C=\{c_1, \dots, c_j\}$; $c_j=\{s_{out}, s_{in}, l\}$: $s_{out} \in S$, $s_{in} \in S$, $l \in L$.

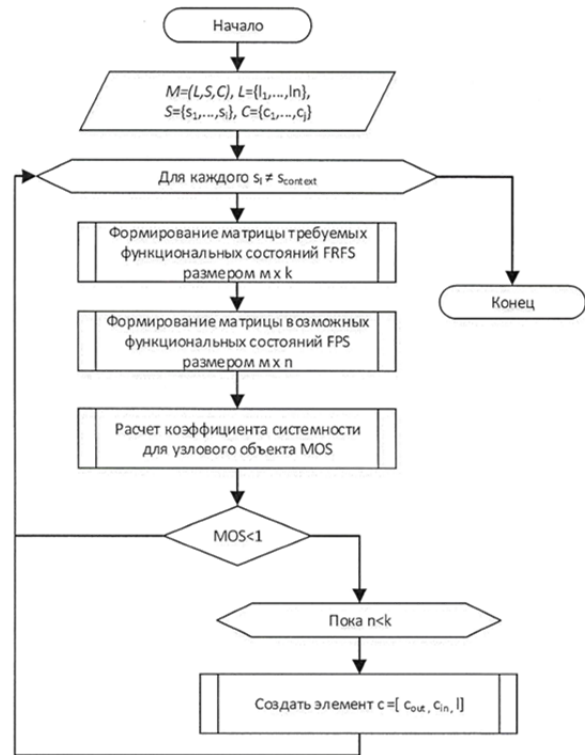


Рис. 7. Алгоритм учета закона необходимого разнообразия в системно-объектной модели

2. Для каждого элемента множества S за исключением контекстного узлового объекта $S_{context}$ сформировать матрицы требуемых и возможных функциональных состояний и рассчитать коэффициент системности.

3. Если коэффициент системности равен единице – перейти к анализу следующего узлового объекта, иначе добавить связи для анализируемого узлового объекта до тех пор, пока количество возможных функциональных состояний не превысит количество требуемых функциональных состояний.

4. Алгоритм выполняется, пока не будут проанализированы все узловые объекты системно-объектной модели.

6. Учет принципа совместимости

Рассмотрим механизм учета **принципа совместимости** [4] в процессе системно-объектного моделирования. Условием взаимодействия между системами является наличие у них относительной совместимости, то есть относительной качественной и организационной однородности. Используя терминологию

исчисления систем как функциональных объектов [4, 10], данный принцип можно представить в виде следующих выражений, например, для двух узловых объектов: $s_1 = [(L_1, L_1); f(L_1)L_1; (O_1, O_1, O_1)]$; $s_2 = [(L_2, L_2); f(L_2)L_2; (O_2, O_2, O_2)]$.

Узловые объекты считаются совместимыми, т.е. можно связать первый элемент со вторым если $s_2.L_2 \subset s_1.L_1$ и наоборот, $s_1.L_1 \subset s_2.L_2$. Проще говоря, связать два узловых объекта представляется возможным в случае наличия соответствующих портов у обоих. Для решения проблемы совместимости двух узловых объектов, необходимо добавить порты требуемых типов.

Алгоритм учета принципа совместимости на примере двух узловых объектов показан на Рис. 8. Алгоритм состоит из следующих этапов:

1. Инициализировать узловые объекты s_1 и s_2 .
2. Для каждого порта входящей связи $l_2 \in s_2.L_2$ проверить наличие портов того же вида для исходящих связей s_1 .
3. Если соответствующий порт найден, перейти к следующему порту второго узлового объекта, если порт не найден, создать порт для связывания данных узловых объектов.
4. Алгоритм работает, пока не будут проанализированы все порты второго узлового объекта.

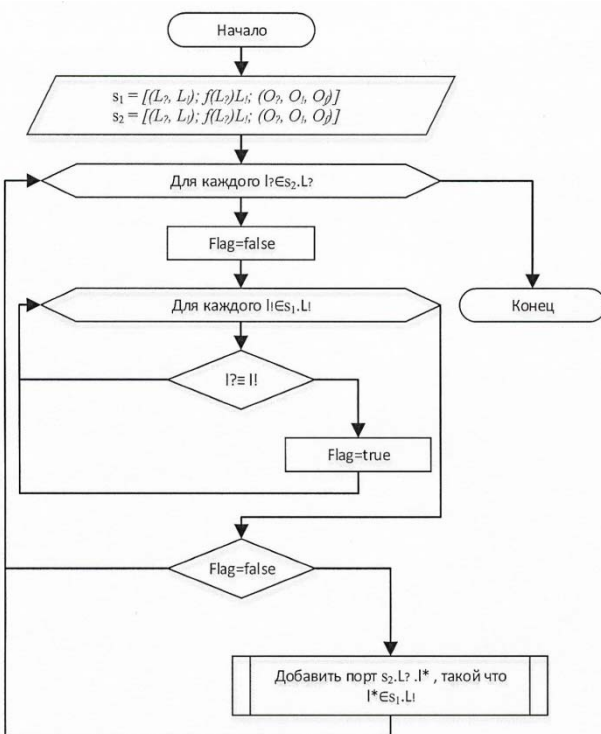


Рис. 8. Алгоритм учета принципа совместимости в системно-объектной модели

Заключение

Приведенные в работе алгоритмы модификации систем позволяют учесть некоторые общесистемные закономерности при построении моделей знаний об организационно-деловых и производственно-технологических процессах. Выполнение других общесистемных принципов и закономерностей при системно-объектном моделировании организационных знаний является предметом дальнейшего исследования автора.

При разработке алгоритмов использовано исчисление систем как функциональных объектов, предложенное в работе [4,10]. Применение алгоритмов будет способствовать повышению качества и адекватности системно-объектного моделирования за счет частичной автоматизации процессов верификации моделей с позиции соответствия общесистемным принципам и законам. Это в свою очередь обеспечит повышение эффективности проектирования организационных и информационных систем, результативность управления ими.

Литература

1. Садовский В. Н. Основания общей теории систем. Логико методологический анализ. М. 1974. 106 с.
2. Гиг Дж. ванн. Прикладная общая теория систем. М.: Мир. 1981. Т. 1. 336 с.
3. Косяков А., Уильям Н. Свит, Сэмюэль Дж. Сеймур, Стивен М. Бимер. Системная инженерия: принципы и практика. М.: ДМК Пресс. 2017. 623 с.
4. Маторин С.И., Жихарев А.Г., Зимовец О.А. и др. Теория систем и системный анализ: учебник. Под ред. С.И. Маторина. Москва; Берлин: Директмедиа Паблишинг. 2020. 509 с.
5. Маторин С.И., Жихарев А.Г., Зайцева Н.О. Имитационное моделирование с использованием системно-объектного подхода // Прикладная информатика. 2015. Т. 10. № 6 (60). С. 91-104.
6. Жихарев А.Г., Маторин С.И., Зайцева Н.О. Системно-объектный инструмент для имитационного моделирования технологических процессов и транспортных потоков // Искусственный интеллект и принятие решений. 2015. №4. С. 95-103.
7. Жихарев А.Г., Маторин С.И., Маматов Е.М., Смородина Н.Н. О системно-объектном методе представления организационных знаний // Научные ведомости БелГУ. Сер. Информатика. 2013. №8(151). Вып. №26/1. С. 137-146.
8. Маторин С. И., Жихарев А. Г. Учет общесистемных закономерностей при системно-объектном моделировании организационных знаний // Искусственный интеллект и принятие решений. 2018. № 3. С. 115–126.
9. Маторин С.И., Жихарев А.Г., Зайцева Н.О., Бузов П.А., Терещенко В.Г. UFOModeler // свидетельство о

- регистрации программы для ЭВМ 2015663240 / право-обладатели: С.И. Маторин, А.Г. Жихарев, Н.О. Зайцева [и др.]. – № 2015619898; заявл. 20.10.2015; зарегистрировано 14.12.2015 в Реестре программ для ЭВМ ФИПС.
10. Zhikharev A., Matorin S., Egorov I. Formal principles of system-object simulation modeling of technological and production processes // *Journal of Advanced Research in Dynamical and Control Systems*. 2018. Special is. 10. P. 1806-1812.
 11. Жихарев А.Г., Маторин С.И., Бузов П.А., Комарова В.А. Алгоритмические и программные средства имитационного моделирования процессов эволюции сложных динамических систем // *Информационные технологии в науке, образовании и производстве (ИТНОП-2020): сборник материалов VIII Международной научно-технической конференции (г. Белгород, 24–25 сентября, 2020 г.)*. Белгород: ИД «БелГУ». 2020. С. 69-75.
 12. Robin Milner A *Calculus of Communicating Systems*. New York: Springer-Verlag. 1980. 174 p.
 13. Abadi Martin and Luca Cardelli A *Theory of Objects*. New York: Springer-Verlag, 1996. 397 p.
 14. Matorin, S.I., Zhikharev, A.G. Calculation of the function objects as the systems formal theory basis // *Advances in Intelligent Systems and Computing*. 2018. Vol. 679. P. 182-191.

Жихарев Александр Геннадиевич Кандидат технических наук, доцент. Доцент кафедры программного обеспечения вычислительной техники и автоматизированных систем Белгородского государственного технологического университета им. В.Г. Шухова. Области исследований: системный подход, теория систем, системный анализ, CASE-технология, управление знаниями, бизнес-моделирование. E-mail: zhikharev@bsu.edu.ru

System-Object Approach to Organizational Knowledge Modeling Taking into Account System-Wide Patterns

A. G. Zhikharev

V.G. Shukhov Belgorod State Technological University, Belgorod, Russia

Abstract. The paper describes ways of taking into account system-wide principles and patterns in the construction of system-object models of organizational knowledge. The modification of the original formal apparatus of calculating systems as functional objects is proposed. The application of the results obtained will allow, in the future, to automate and improve the procedures of system-object modeling in the design of complex organizational and information systems in order to increase the efficiency of systems management.

Keywords: general system laws, system-object model, organizational knowledge, calculus of systems as functional objects.

DOI 10.14357/20718594210303

References

1. Sadovskiy V. N. *Osnovaniya obshchey teorii sistem. Logiko metodologicheskii analiz [Foundations of the general theory of systems. Logical methodological analysis]*. M. 1974. 106 p.
2. Gig Dzh. vann. *Prikladnaya obshchaya teoriya system [Applied General Systems Theory]*. M.: Mir. 1981. 1. 336 p.
3. Aleksandr Kosyakov, Uil'yam N. Svit, Semyuel' Dzh. Seymur, Stiven M. Bimer. *Sistemnaya inzheneriya: printsipy i praktika [Systems Engineering: Principles and Practice]*. Moskva: DMK Press. 2017. 623p.
4. Matorin S.I., Zhikharev A.G., Zimovets O.A. i dr. *Teoriya sistem i sistemnyy analiz: uchebnik [Theory of systems and system analysis: textbook]*. Pod red. S.I. Matorina. Moskva; Berlin: Direktmedia Publishing. 2020. 509p.
5. Matorin S.I., Zhikharev A.G., Zaytseva N.O. *Imitatsionnoye modelirovaniye s ispol'zovaniyem sistemno-ob'yektnogo podkhoda [Simulation modeling using the system-object approach] // Prikladnaya informatika [Applied Informatics]*. 2015. 6 (60). P. 91-104.
6. Zhikharev A.G., Matorin S.I., Zaytseva N.O. *Sistemno-ob'yektnyy instrumentariy dlya imitatsionnogo modelirovaniya tekhnologicheskikh protsessov i transportnykh potokov [System-object toolkit for simulation of technological processes and traffic flows] // Iskusstvennyy intellekt i prinyatiye resheniy [Artificial intelligence and decision-making]*. 2015. 4. P. 95-103.
7. Zhikharev A.G., Matorin S.I., Mamatov Ye.M., Smorodina N.N. *O sistemno-ob'yektnom metode pred-stavleniya organizatsionnykh znaniy [On the system-object method of representing organizational knowledge] // Nauchnyye vedomosti BelGU. Ser. Informatika [Scientific Bulletin of BelGU. Informatics]*. 2013 8(151). P.137-146.
8. Matorin S. I., Zhikharev A. G. *Uchet obshchesistemnykh zakonostey pri sistemno-ob'yektnom modelirovanii organizatsionnykh znaniy [Accounting for system-wide patterns in system-object modeling of organiza-*

- tional knowledge] // *Iskusstvennyy intellekt i prinyatiye resheniy* [Artificial intelligence and decision-making]. 2018. 3. P. 115–126.
9. Matorin S.I., Zhikharev A.G., Zajceva N.O., Buzov P.A., Tereshchenko V.G. UFOModeler [UFOModeler] // *svidetel'stvo o registracii programmy dlya EVM 2015663240* [certificate of registration of the computer program 2015663240] / pravoobladateli: S.I. Matorin, A.G. Zhikharev, N.O. Zajceva [i dr.]. – № 2015619898; zayavl. 20.10.2015; zaregistrovano 14.12.2015 v Reestre programm dlya EVM FIPS.
 10. Zhikharev A., Matorin S., Egorov I. Formal principles of system-object simulation modeling of technological and production processes // *Journal of Advanced Research in Dynamical and Control Systems*. 2018. 10. P.1806-1812.
 11. Zhikharev A.G., Matorin S.I., Buzov P.A., Komarova V.A. 2020. Algoritmicheskiye i programmnyye sredstva imitatsionnogo modelirovaniya protsessov evolyutsii slozhnykh dinamicheskikh system [Algorithmic and software tools for simulating the evolution of complex dynamic systems] // *Informatsionnyye tekhnologii v nauke, obrazovanii i proizvodstve (ITNOP-2020): sbornik materialov VIII Mezhdunarod-noy nauchno-tekhnicheskoy konferentsii* [Information technologies in science, education and production (ITSEP-2020): collection of materials of the VIII International Scientific and Technical Conference] (g. Belgorod, 24–25 sentyabrya, 2020 g.). Belgorod: ID «BelGU». P. 69-75.
 12. Robin Milner. *A Calculus of Communicating Systems*. New York: Springer Verlag. 1980.174p.
 13. Abadi Martin and Luca Cardelli. *A Theory of Objects*. New York: Springer-Verlag. 1996. 397p.
 14. Matorin S.I., Zhikharev A.G. Calculation of the function objects as the systems formal theory basis // *Advances in Intelligent Systems and Computing*. 2018. 679. P. 182-191.

Zhikharev Aleksandr G. Candidate of technical sciences, docent. Docent of the department of software for computer technology and automated systems, Belgorod State Technological University named after V.G. Shukhov. Research areas: systems approach, systems theory, systems analysis, CASE technology, knowledge management, business modeling. E-mail: zhikharev@bsu.edu.ru