

МАТЕМАТИЧЕСКОЕ И КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ

УДК 004.45

**ДЕКОМПОЗИЦИЯ И АГРЕГАЦИЯ ФУНКЦИОНАЛЬНЫХ
МОДЕЛЕЙ СИСТЕМ**

В. А. Бочаров, ассистент кафедры «Системы автоматизированного проектирования» МГТУ им. Н.Э. Баумана, Москва, Россия;

orcid.org/0009-0006-7724-3862, e-mail: bocharov.v@bmstu.ru

Т. М. Волосатова, к.т.н., доцент кафедры «Системы автоматизированного проектирования» МГТУ им. Н.Э. Баумана, Москва, Россия;

orcid.org/0000-0002-2882-5464, e-mail: tamaravol@gmail.com

М. В. Филиппов, к.т.н., доцент кафедры «Программное обеспечение ЭВМ и информационные технологии» МГТУ им. Н.Э. Баумана, Москва, Россия;

e-mail: filippov.mire@mail.ru

Л. В. Чичаева, младший научный сотрудник лаборатории «Технологии цифрового проектирования и системное ПО» ФАУ «ГосНИИАС», Москва, Россия;

email: lvchichaeva@gosniias.ru

С. А. Продан, младший научный сотрудник лаборатории «Технологии цифрового проектирования и системное ПО» ФАУ «ГосНИИАС», Москва, Россия;

email: saprodan@gosniias.ru

*Статья посвящена поиску новых, более эффективных средств и методов разработки сложных технических систем. Область исследования охватывает функциональное моделирование, которое является источником важнейших свойств целевой системы для условий, при которых она работает. В рамках данного исследования решается задача сведения функции всей целевой системы к некоторому числу более простых функций, анализ и синтез которых укладывается в возможности имеющихся математических методов и вычислительных средств. Для решения поставленной задачи предлагается общая абстрактная метамодель функций системы и связей её частей, а также теория, дающая возможность «собирать» «разбирать» и модели систем на отдельные её смысловые составляющие, не нарушая функциональной целостности. **Цель работы** – повышение результативности и сокращение сроков НИОКР ответственных технологий и систем с помощью отечественной платформы системной инженерии, отвечающей современным и перспективным требованиям.*

Ключевые слова: функциональная модель, функциональная декомпозиция, язык моделирования, декомпозиция, семантика, математические методы

DOI: 10.21667/1995-4565-2024-89-85-93

Введение

Моделирование, будучи основным методом разработки систем, позволяет исследовать ещё несуществующую систему на её виртуальном аналоге, который при известных допущениях и упрощениях имеет свойства, схожие с материальной системой. Множество общих свойств системы и её модели отражает *адекватность* модели. Следует заметить, что физическая система, будучи материализованной, является моделью самой себя с адекватностью, равной единице. Разрабатывая или исследуя систему, инженерам приходится делить её на составные части и учитывать различие в результатах и выводах этого процесса, обусловленное дезинтеграцией (декомпозицией) или наоборот, интеграцией (сборкой) модели [1, 2, 4]. В связи с этим возникает вопрос: как, по каким принципам и правилам надо делить систему,

чтобы после проработки каждой её отдельной части не терять целевые свойства системы и не обретать ненужные? В данной статье приводится анализ свойств систем, формирующих соответствие их внутреннего строения внешним свойствам, которое можно исследовать и на основании которого можно производить функциональную декомпозицию (далее – ФД) функций до того уровня детализации, когда техническое решение становится очевидным. При этом должна обеспечиваться гарантия «собираемости» функций в единое целое.

Теоретическая часть

За отправную точку исследования были взяты следующие постулаты.

1. Все предметы и явления материального мира подчиняются одним и тем же естественным законам.

2. Системы с подобными структурами имеют подобные свойства.

3. Две системы, чей путь зарождения, развития и становления был идентичен, подобен или эквивалентен, после своего формирования идентичны, подобны или эквивалентны друг другу.

4. Все отношения внутри и вовне системы по своей природе являются функциональными и базируются на отношении *изменяемого, изменённого, изменения и условий изменения*.

5. Система с точки зрения взаимодействия с окружающей средой имеет три множества: а) то, что она принимает от внешней среды; б) то, что она отдаёт во внешнюю среду; в) то, что имеется внутри и образует связь между (а) и (б).

Введём некоторую дополнительную терминологию, необходимую для формализации функциональной модели системы. За основу берём общий принцип описания функциональности системы в нотации IDEF0 [5], но добавим к нему недостающие элементы.

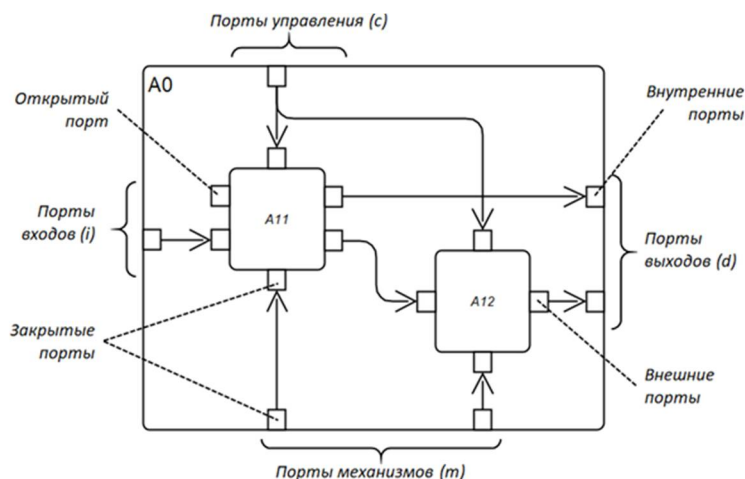


Рисунок 1 – Понятия, определяющие функциональную модель системы

Figure 1 – Concepts defining the functional model of the system

Система, как набор функций и функциональных связей, имеет формальный вид [6]:

$S = \{ \{ F_{i,k} \}, \{ J_{f,t} \} \}$, где:

S – (system) система;

F – (function) функции, номер i на уровне вложенности k ;

J – (joint) функциональные связи, от порта f (from) до порта t (to).

Функция как активный элемент системы имеет вид:

$F = \{ id, \{ P \} \}$, где:

F – функция;

id – идентификатор функции;

P – набор портов;

Порты как свойства функции имеют вид:

$P = \{ q, n, m \}$, где:

P – порты функции;

q – номер порта;

n – тип порта, где: $n = \{input, output, control, mechanism\}$;

m – сторона порта, где: $m = \{inner, outer\}$.

Типы функциональных связей:

$f-p(f, p)$ – связь функции с портом;

$j(f, t)$ – соединение порта *from* (откуда) с портом *to* (куда);

$part-of(f, f)$ – отношение части-целого для функций.

У функциональных связей, как показали исследования, имеются четыре вида ограничений, вытекающих из семантических свойств.

1. Связи не могут пересекать границу между внешней и внутренними функциями, а могут только перестыковываться через *транзитные порты*.

2. Внутренние порты с внешними могут быть связаны, только если они однородные (входы с выходами, управление с управлением и т.д.).

3. Исходящими могут быть только выходы, принимающими – только входы и управление.

4. Внутренние порты никогда не связываются между собой напрямую по очевидной причине: они являются таковыми ради наследования свойств функции от составной к набору составляющих.

Внутренние связи функции имеют следующую семантику [7].

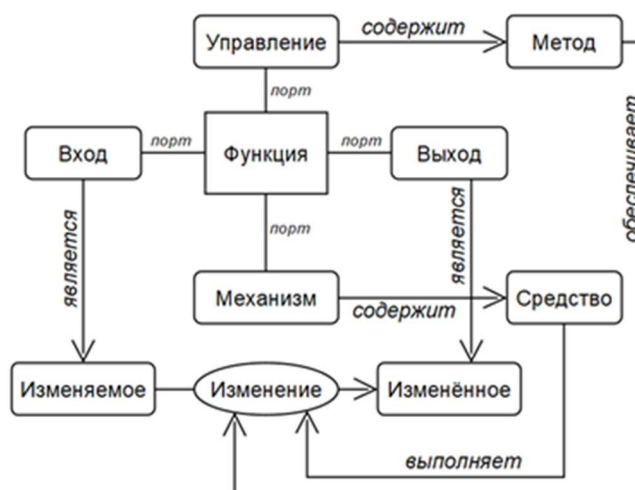


Рисунок 2 – Семантика внутренних связей функции

Figure 2 – Semantics of function internal connections

На языке пропозиционной логики связи функции и её портов будут выглядеть как набор высказываний вида: $\Phi = \{a, b, c, d, e, f, g, h\}$, где:

$a: \{has(f, p), \text{“функция имеет порты”}\}$

$b: \{has-type(xor\{i, d, c, m\}), \text{“типология портов”}\}$

$\{вход, выход, управление, механизм\}$

$c: \{in(["asset"], c), \text{“управление содержит метод”}\}$

$d: \{in(["unit"], m), \text{“механизм содержит средство”}\}$

$e: \{isA(["being_changed"], i), \text{“вход является изменяемым”}\}$

$f: \{isA(["changing"], d), \text{“выход является изменённым”}\}$

$g: \{does(["changing"], ["method"]), \text{“метод обеспечивает изменение”}\}$

$h: \{func(["being_changed"], ["changing"], ["method"]),$

$\text{“изменение превращает изменяемое в изменённое”}\}$.

Одновременно на том же языке пропозиционной логики функциональные связи выражают:

- отношение *части и целого* между функциями;
- *распределение*: разделение портов на части или их повторное применение;

- *суперпозицию*: тождество отправляемого и принимаемого порта;
- *обратную связь*: повтор обработки или коррекция метода.

Использование логики предикатов: модель, представленная таким образом, может обрабатываться процедурами логики предикатов (логики первого порядка) [8], такими как:

- *требование обязывающее*: $\{ "condition" \rightarrow "obligation" \}$;
- *требование запрещающее*: $\{ \neg "statement" \}$;
- *индукция*: $\{ "assumption" \& "inference rule" \rightarrow "conclusion" \}$;
- *дедукция*: $\{ "fact" \& "deductive rule" \rightarrow "reason" \}$.

Здесь требования отражают отношение к целевой системе со стороны её создателя и потребителя, которое сводится к описанию фактов, возникающих при заданных условиях, и фактов, которые не должны иметь место. Индукция образует отношение между явно заданными фактами и тем, что за ними следует, в соответствии с заданными правилами логического вывода. Дедукция, – напротив, восстанавливает причинность факта, опираясь на природу появления фактов данного вида [9].

Пользуясь этим инструментом и семантическими правилами, можно определять, верно ли построена модель, определена ли специализация входов и – управления и соответственно, верно ли они распределены по разным функциям. Также можно проверить, проложена ли связь по данным и по управлению и определены ли обратные связи.

Декомпозиция системы с известной внешней функциональностью и неизвестный внутренним устройством сводится: а) к сортировке и группировке функций по характеру действий; б) сортировке и группировке исходных и управляющих данных; в) распределению данных по функциям; г) соединению функций по входам и выходам; д) формированию обратных связей.

Критерием успешной декомпозиции составной функции и на группу составляющих является тождество списков *внутренних портов* составной функции списку *внешних портов* составляющих при соблюдении граничных условий семантической связности.

Предпосылками к декомпозиции являются пути упрощения системы, в числе которых – *длина цепочки преобразований, степени предикатов и цикломатическая сложность*.

Длина пути – максимальное число шагов, промежуточных состояний или преобразований от начальных входов до конечных выходов. Преобразования могут быть последовательные и параллельные, и во всей структуре преобразований можно выделить путь наибольшей длины. Сокращая число шагов наибольшего пути, мы сокращаем нагрузку на систему.

Степень (арность) предикатов отражает внутреннюю сложность выполняемых функцией действий.

$p(X)$ – *арность* = 1: присвоение X единичного семантического значения p .

$p(X, Y)$ – *арность* = 2: операция над X по формуле p , выводящая Y .

$p(X, Y, Z)$ – *арность* = 3: операция над X по формуле p , использующей условие Y , выводящая Z .

Например, если есть предикат 3 степени и известны процедуры, возвращающие значения переменных, то выражение с этим предикатом может быть преобразовано в набор выражений с предикатами со степенью на единицу меньше:

$$f:\{a, b, c\} = \{P:\{\{f1, f2, f3\} \rightarrow f\}, f1:\{a, b\}, f2:\{a, c\}, f3:\{b, c\}\}$$

Применительно к предикату функции его переменная – значение функции выводится из аргумента и формулы преобразования аргумента. Если представить функцию в виде диаграммы Венна, то она выглядит как три множества с двумя областями их пересечения:

$A:\{a1, a2, \dots a(i)\}$ – множество фактов исходной реальности;

$B:\{b1, b2, \dots b(j)\}$ – множество фактов изменённой реальности;

$C:\{c1, c2, \dots c(k)\}$ – факты, управляющие различиями и изменениями;

$p:\{a, b\}$ – предикат, определяющий отношение элементов множеств A и B ;

$f:\{a, b, c\}$ – функция как управляемое отношение исходных фактов ко вновь порождённым.

Взаимодействие этих множеств можно отобразить, как показано на рисунке 3:

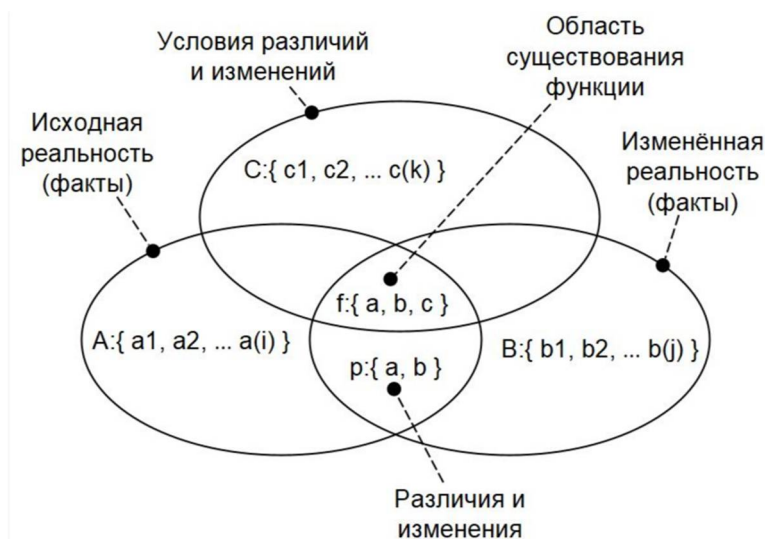


Рисунок 3 – Функция на языке логики предикатов
Figure 3 – Function in the language of predicate logic

При переходе от одной функции к связке из двух последовательных возникает ещё одна область пересечения – между выходами предшествующей функции и входами последующей. Таким образом, функциональную модель системы целиком можно представить как набор пересекающихся множеств, что даёт основание применять к модели известные теоретико-множественные и графо-структурные методы исследования.

Число циклов и их вложенность (цикломатическая сложность, характерная больше для программного обеспечения кибернетических систем). Она в ряде случаев не может быть понижена в силу естественных причин, но можно избегать её неоправданного повышения.

Функции информационно-управляющей системы (САПР), реализующей процесс функциональной декомпозиции и обеспечения семантической целостности, представлены на рисунке 4.

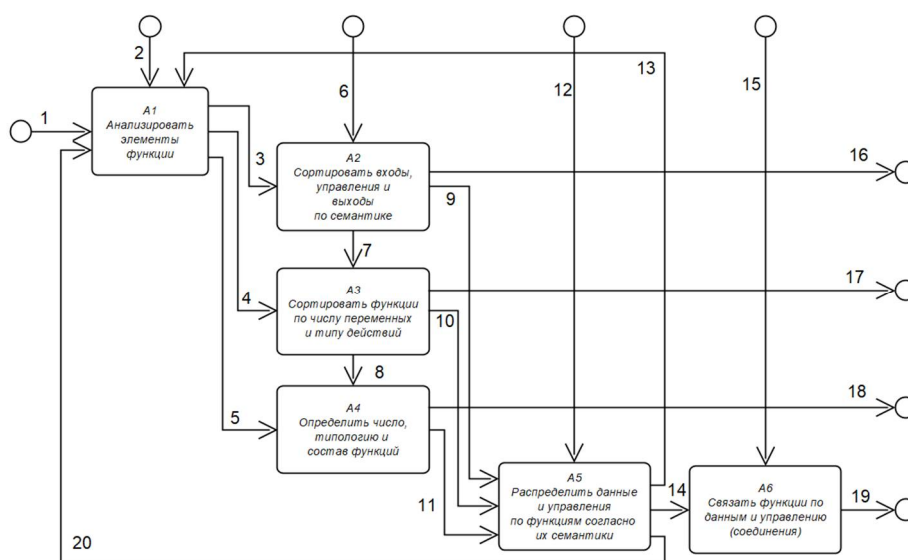


Рисунок 4 – Функциональная схема системы, выполняющей функциональную декомпозицию систем

Figure 4 – Functional diagram of the system performing functional decomposition of systems

Результатом выполнения данного действия будет тождество исходной функции системы и эквивалентного ей набора составляющих функций:

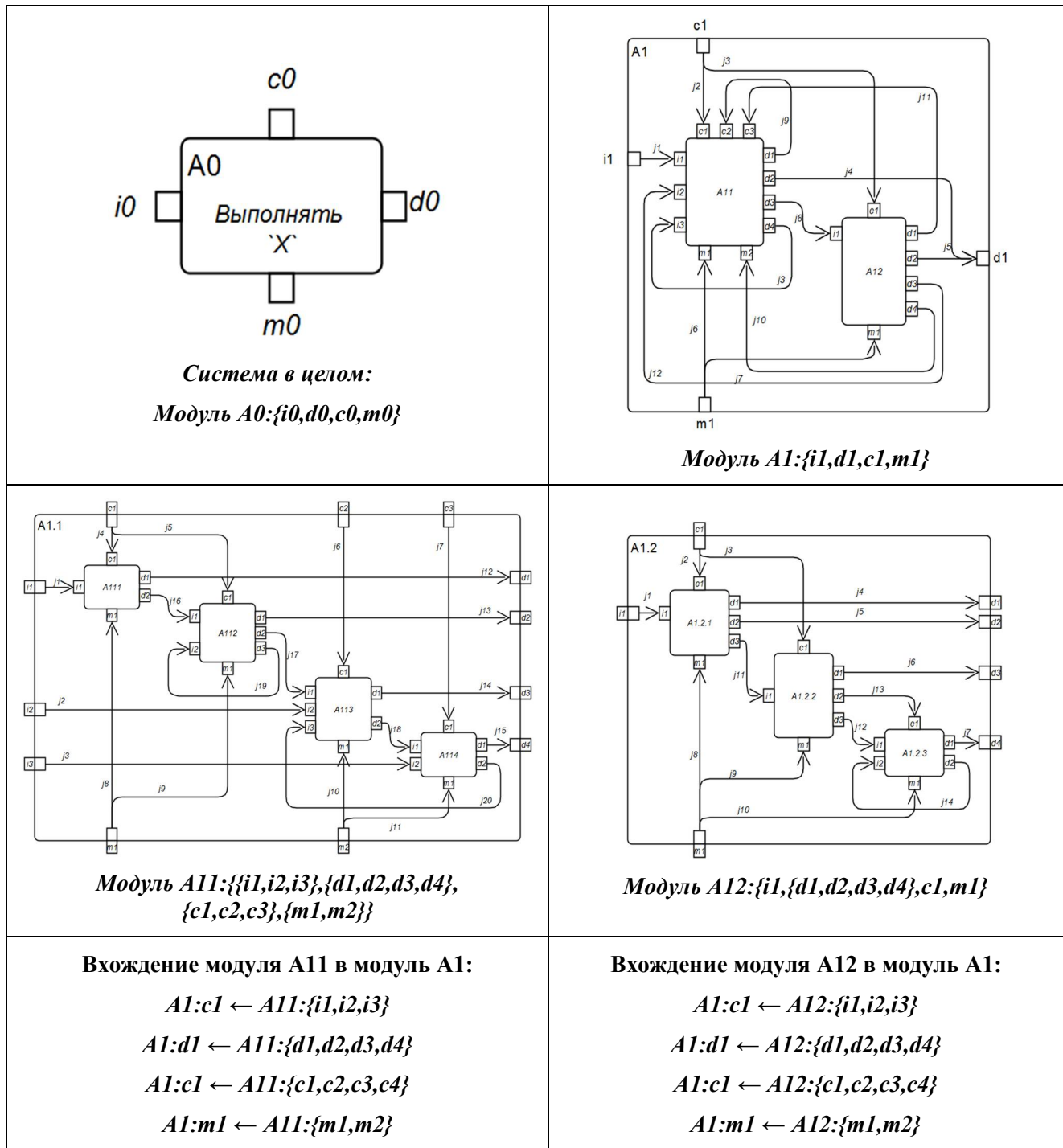


Рисунок 5 – Функциональная декомпозиция: графическое и формальное представление
Figure 5 – Functional decomposition: graphical and formal representation

Эквивалентность функциональности системы до и после декомпозиции может быть выражена как: $A0 \equiv \{A11, A12\}$, где:

$A11 \equiv \{A111, A112, A113, A114\}$ и $A12 \equiv \{A121, A122, A123\}$.

Таким образом, формализм абстрактной системы, отвечающий требованиям семантической целостности модели системы, сводится к вектору из множества элементов словаря (понятий и отношений), инвариантов (аксиом и постулатов) и процедур над словарём и инвариантами, выводящих одни свойства системы из других. Всякая частная модель, посвящённая тому или иному подмножеству свойств системы, использует тот словарь предметной области, те аксиомы и те процедуры, которые входят в это подмножество. Это даёт возможность

исследовать не одну модель, а каскад из разных моделей, посвящённых разным группам свойств целевой системы, при том не теряя их взаимосвязи.

Имея методологию функциональной декомпозиции, можно строить комплекс систем из целевой (разрабатываемой) и разрабатывающей, в котором будет отражён путь пошагового развития целевой системы: от декларации потребительских свойств и до детальной функциональной спецификации, заключающей в себе все принятые при разработке решения с возможностью получения их обоснования.

Экспериментальные исследования. В рамках проводимых в НИИ работ по развитию платформы системной инженерии был проведён ряд экспериментов по интерпретации хода разработки некоторых авиационных систем на языке ФРС, где был опробован метод функциональной декомпозиции и получены некоторые выводы о его эффективности. Результаты показали, что наибольший эффект даёт использование метода ФД в условиях высокой неопределённости конечного результата, распределения работ «широким фронтом» между большим числом интенсивно взаимодействующих участников и часто обновляющихся системных требований. Также были опробованы сохранение и повторное использование абстрактного знания на уровне класса систем.

Заключение

Полученный в результате исследования метод функциональной декомпозиции систем изменил представление о проектировании и конструировании систем следующим образом.

- Добавил семантики в формальное представление функций систем, позволив применять эффективные математические методы, не усложняя модели.
- Позволил отдельно оперировать данными (классами), предикатами и процедурами.
- Свёл процедуры разработки к минимальному числу элементарных операций и минимальному числу внутренних состояний разрабатываемой системы.
- Позволил строить системы, верные по построению – не нуждающиеся в последующей верификации.
- Позволил ввести ещё одну классификацию для функций – по роду выполняемого действия, безотносительно физики. Благодаря этому конечную функцию стало возможным определять как общую инстанцию взаимно ортогональных классов, начиная с данного (по роду действия), что расширило возможности дальнейшей автоматизации и привлечения к процессам НИОКР искусственного интеллекта.

Полученная методология ФД в целом вносит вклад в возможность создания информационно-управляющей системы, решающей задачи моделирования и исследования моделей с использованием различных стратегий и их комбинаций. В числе возможных стратегий могут быть декомпозиция и агрегация функций от составляющего к составному, от общего к частному, от требований к решениям, в порядке эволюции и в соответствии с идеями разработчика.

Библиографический список

1. Медведев Д., Шани У., Дори Д. Взгляд внутрь концептуальных моделей: Подход через теоретико-графовые запросы // Журнал Прикладные Науки. 2021.
2. Мино Г., Миссауи Р., Гордин Р. Концептуальное моделирование концептуальными графами // Интернет-журнал Researchgate.
3. Бентен И., Дитмарш Х., Эйк Я., Яспарс Я. Логика в действии // Журнал Прикладные Науки. 2016.
4. Elwert M., Manuel Ramsaier, Eisenbart B. Digital Function Modeling in Graph-Based Design Languages // Журнал Прикладные Науки, 2022.
5. Function Modeling Manual (IDEF0) // Стандарт ARWAL-TR-81-4023.
6. Jørgensen, Bay S., Morten L., Jensen N. Functional Modeling View on Product and Process Engineering in Design and Operations // DOI:10.1021/acs.iecr.8b06338

7. **Евстигнеев В.А., Касьянов В.Н.** Словарь по Графам в Информатике // Новосибирск: ООО «Сибирское Научное Издательство», 2009.

8. **Можайский А.Ф.** Методология гиперграфового моделирования структурно-технологических возможностей бортовых систем малых космических аппаратов. Труды Военно-Космической Академии имени А.Ф. Можайского. Выпуск 673.

9. **Топольский Н.Г., Трефилов Г.Б., Сатин А.П.** Алгоритмы многоуровневой иерархической декомпозиции гиперграфовых и графовых моделей причинно-следственных связей в АСУ безопасностью критически важных объектов // МГСУ. 2013.

UDC 004.45

DECOMPOSITION AND AGGREGATION OF FUNCTIONAL SYSTEM MODELS

V. A. Bocharov, BMSTU, assistant teacher, Moscow, Russia;
orcid.org/0009-0006-7724-3862, e-mail: bocharov.v@bmstu.ru

T. M. Volosatova, Ph.D. (Tech.), BMSTU Associate Professor, Moscow, Russia;
orcid.org/0000-0002-2882-5464, e-mail: tamaravol@gmail.com

M. V. Filippov, Ph.D. (Tech.), BMSTU Associate Professor, Moscow, Russia;
e-mail: filippov.mire@mail.ru

L. V. Chichayeva, junior researcher of Digital Design Technologies and System SW lab of Scientific Research Center «GosNIIAS», Moscow, Russia;
email: lvchichayeva@gosniias.ru

S. A. Prodan, junior researcher of Digital Design Technologies and System, SW lab of Scientific Research Center «GosNIIAS», Moscow, Russia;
email: saprodan@gosniias.ru

*The article is devoted to the search for new, more efficient tools and methods of model based system design and analysis. The field of the research includes functional modeling being the source of target system functional properties in the conditions in which it operates. System modeling is defined as using virtual analogues instead of physical things, which under certain assumptions and simplifications have properties similar to the material system. Within the framework of this study, the problem of reducing the functional complexity of entire target system to a certain number of simpler functions is solved, the analysis and synthesis of which fits within the capabilities of available mathematical methods and computing tools. To solve this problem, we propose a general abstract metamodel of system functions, the connections of its parts, as well as a theory that allows “disassembling” and “assembling” system models into its individual semantic parts, while functional integrity is kept. **The aim of this research** is to increase the effectiveness and to reduce R&D time of critical technologies and systems using a modern national system engineering platform.*

Keywords: functional model, functional decomposition, modeling language, semantics, mathematical method.

DOI: 10.21667/1995-4565-2024-89-85-93

References

1. **Medvedev D., Shani U., Dori D.** Vzglyad vnur' konceptual'nyh modelej: Podhod cherez teore-tiko-grafovye zaprosy. Zhurnal Prikladnye Nauki. 2021.

2. **Mino G., Missaui R., Godin R.** Konceptual'noe modelirovanie konceptual'nymi grafami. Internet-zhurnal Researchgate.

3. **Benten I., Ditmarsh X., Ejik Ya., Yaspars Ya.** Logika v dejstvii. Zhurnal Prikladnye Nauki, 2016.

4. **Michael Elwert, Manuel Ramsaier, Boris Eisenbart.** Digital Function Modeling in Graph-Based Design Languages. Applied Sciences. 2022.

5. **Function Modeling Manual (IDEF0). Standard ARWAL-TR-81-4023**

6. **Jørgensen, Sten Bay, Lind, Morten; Jensen, Niels.** Functional Modeling View on Product and Process Engineering in Design and Operations. DOI:10.1021/acs.iecr.8b06338

7. **Evstigneev V.A., Kas'yanov V.N.** Slovar' po Grafam V Informatike. Novosibirsk, ООО «Sibirskoe Nauchnoe Izdatel'stvo», 2009.

8. **Mozhaisky A.F.** *Aerospace & Defense Academy Works*, Issue 673. Methodology of graph-structural modeling of the small-sized orbiters onboard equipment.

9. **Topolsky N.G., Trefilov G.B., Satin A.H.** Algorithms of multi-level hierarchic decomposition of graph and hypergraph models of the Reason-Consequence relations in the safety critical systems. MGSU, 2013.