

КЛАСТЕРНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Г.Г. Куликов, В.В. Антонов, К.А. Конев

Уфимский государственный авиационный технический университет, г. Уфа, Россия

Предлагается новый подход к созданию кластерной структуры программного обеспечения, обеспечивающей наибольшую защиту от получения злоумышленником контроля над системой поддержки принятия решений, а также высокую устойчивость к порче и взлому. Рассматривается подход к решению проблемы безопасности программного обеспечения информационных систем путём кластеризации его на модули. Показана возможность формального построения таких структур методами теории множеств и теории категорий. Доказано, что множество модулей удовлетворяет всем постулатам и аксиомам теории категорий, следовательно является категорией. Определены условия фрактального представления автоматизированной информационной системы от уровня структурных модулей системы до уровня микрокоманд с выделением конкретных объектов и морфизмов на уровнях иерархии.

Ключевые слова: кластеризация, фрактальная структура, теория множеств, теория категорий, объект, морфизм, автоматизированная информационная система, программное обеспечение, модуль.

Введение

Прогресс идёт семимильными шагами и комплексная автоматизация бизнес-процессов для крупных и даже средних по размеру организаций становится всё более актуальной проблемой, особенно на высоко конкурентных рынках, требующих минимизировать затраты, сроки, ошибки. Очевидно, что механизм управления в любой социально-экономической системе (команда, организация, служба, фонд и т. д.) является важнейшим компонентом. Существует множество подходов к повышению эффективности данного механизма (см. например [9, 11]), которые обычно предполагают его формализацию и автоматизацию. При этом автоматизация данного механизма сильно тормозится именно соображениями безопасности. Действительно, программа, способная стать источником команд для подразделений организации – прекрасная мишень для любых злоумышленников: от жуликов до террористов. Поэтому подходы к созданию проекта программного обеспечения (ПО) автоматизированной информационной системы (АИС) должны изначально отталкиваться от задач по максимальному противодействию любому злоумышленнику, изначально выстраивая структуры, в которых чрезвычайно сложно нарушить или перехватить управление. Одним из подходов по решению данной задачи является кластеризация программного обеспечения, позволяющая за счёт создания дополнительных механизмов управления модулями существенно затруднить перехват управления злоумышленником.

1. Концепция кластерного программного обеспечения для автоматизированной информационной системы

Кластерная структура – это система, объединяющая несколько однородных элементов, которые могут рассматриваться как самостоятельные единицы, обладающие определёнными свойствами [5]. В вычислительной технике кластерная структура – слабо связанная совокупность нескольких вычислительных систем, работающих совместно для выполнения общих приложений, и представляющихся пользователю единой системой [3]. В программировании такая структура предполагает возможность дублирования функций между компонентами так, чтобы имелась возможность незаметно для пользователя заменять отдельные аппаратные или программные ресурсы.

Концепция кластерной программной системы состоит в построении такой системы управления связями между модулями программного обеспечения, в которой некорректная работа любого модуля позволит его отключить без останова работы программы и переключить поток задач на его копию.

Структурно кластерная программная система должна иметь характер модульно-объектной, т. е. иметь над традиционными классами программного обеспечения дополнительный уровень категоризации, включающий критерии выделения компонентов программной системы в отдельные модули.

Необходимыми элементами такой программной системы являются:

- механизм самодиагностики;
- механизм подключения модулей, включающий адаптивное множество связей между модулями;
- механизм сокрытия резервных модулей.

Идея концепции в том, что конфигурация программного обеспечения может быть представлена в виде процесса, который имеет динамическую структуру. Конструктор вызывает только те модули, которые нужны для данной задачи, а при подозрении на некорректную работу их резервные копии, хранящиеся отдельно. Сам конструктор также может иметь сложную структуру и состоять из заменяемых модулей.

Введем следующие обозначения:

A – имя атрибута;

D – значение атрибута;

$\varphi: A \rightarrow D$ – отображение, ставящее в соответствие каждому атрибуту множество его возможных значений.

Тогда теоретико-множественное представление процесса $\{\langle A_i D_i \rangle, i = 1, \dots, n\}$ есть множество пар имён и значений атрибутов, взаимодействующих между собой на основе множества правил (обозначим π). Обозначим через $G = \{\bar{G}, \tilde{G}\}$ множество отношений атрибутов (\bar{G} – количественные отношения, \tilde{G} – качественные отношения).

Если ввести понятие характеристики процесса x_i , то её можно определить следующим образом:

$$x_i = \{\langle A_i, D_i \rangle, x \in \sigma\}, \quad (1)$$

где σ – совокупность сведений, характеризующих объекты, которая может быть представлена в виде векторов параметров $y_i = \{y_1^i, \dots, y_n^i\}$.

Будем считать, что для G определено множество типов оценки T . T – множество значений лингвистической переменной A (терм-множество), представляющих собой наименования нечетких переменных, областью определения каждой из которых является множество X . Множество T называется базовым *терм-множеством* лингвистической переменной A [12]. Значения разбиваются на классы объектов, которые взаимодействуют друг с другом на основе правил. Очевидно, что множество правил оценки определяется кортежем следующего вида:

$$\pi = \langle G, T \rangle. \quad (2)$$

Отсюда из (1) и (2) получаем формальное определение процесса в виде кортежа компонентов

$$Z = \{\langle A_i, D_i \rangle, \{\bar{G}, \tilde{G}\}, \{\langle \bar{G}, \tilde{G} \rangle, T\}\}, i \in N. \quad (3)$$

В частном случае с A можно сопоставить набор модулей, а с D – множество принимаемых значений их состояний («текущий», «активный», «резервный», «аварийный», «опасный»). Множество отношений между модулями может быть выражено количественно и качественно ($G = \{\bar{G}, \tilde{G}\}$). Правила π соотношений пар {модуль, состояние} будут определяться механизмом опроса, вызова, диагностики и отключения модулей, которые будут связаны с множеством типов выбора T («модуль работоспособен», «модуль скорее работоспособен», «модуль скорее не работоспособен», «модуль неработоспособен»).

Совокупность информационных характеристик процесса, установленных отношений и правил установления отношений можно использовать для формального определения процесса в виде кортежа компонентов, что с учётом (3) даст выражение:

$$z_i = \{x_i, G, \pi\}. \quad (4)$$

На основании (4) и учитывая, что конфигурация ПО будет квалифицироваться конечным количеством взаимодействий бизнес-процессов, можем представить её в виде автоматизированной информационной системы, управляемой и инвариантной по времени. Очевидно, что данная система будет характеризоваться конечным количеством состояний. Так как в теории категорий преобразования объектов (объекты – аналоги множеств, преобразования – аналоги отображений) входят в аксиоматическое определение наравне с объектами, наибольшее внимание может быть уделено различным формам их преобразований.

Состояние автоматизированной информационной системы Q определяется соотношением информационных характеристик предметной области S , множествами свойств, определяемых связями между элементами P и множеством отношений объектов G :

$$Q = \{S, P, G\}. \quad (5)$$

Таким образом, из (5) и по аналогии с [2] можно сделать вывод о наличии зависимости состояний информационной системы в целом от состояний программных модулей, структурно входящих в её реализацию. Более того, данная зависимость может быть описано формально.

Для того чтобы показать масштабируемость кластерного подхода, покажем, что можно построить схему, характеризующую отображение с уровня программных модулей на уровень АИС на множестве типов и состояний.

2. Кластерное программное обеспечение АИС

Принцип иерархии системного анализа [14] утверждает, что сложные системы, как правило, имеют уровневые структуры, в рамках которых возможно определение закономерностей их саморазвития. Более того, такие сложные системы, как общество, организация или АИС имеют сложную онтологию, определяющую основные понятия и принципы их функционирования [7, 8], что предполагает наличие сложных взаимосвязей и между уровнями композиции этих систем. Поскольку уровень программных модулей есть композиция АИС, докажем на уровне АИС, что кластеризация подобна уровню модуля.

Введем следующие обозначения:

$$T = \{t_i : i \in I\} \text{ – множество типов объектов;}$$

$\delta: T \rightarrow 2^A$ – отображение, ставящее в соответствие каждому типу объекта множество его имен атрибутов.

Тогда данное отображение будет связывать модули ПО с именами их атрибутов, т. е. демонстрировать структуру этих модулей.

$$U = \{x_i : i \in I\} \text{ – множество объектов;}$$

$$\gamma: U \rightarrow T \text{ – отображение, которое ставит в соответствие каждому объекту его тип.}$$

Данное отображение показывает соотношение модулей с их типами по шкале нечёткой оценки.

Тогда каждый объект может быть охарактеризован своей «информационной схемой», которая представима кортежем:

$$S = \left\langle \{A^j\}, \{D^j\}, \varphi, T, \delta \right\rangle, \quad (6)$$

где $\{A^j\}$ – множество имён атрибутов категории объекта с номером j ; $\{D^j\}$ – множество значений атрибутов категории объекта с номером j .

Данная схема (6) показывает связь всех модулей в ПО, всех их возможных состояний, а также отображения на множестве типов и атрибутов.

Тогда ПО объекта, сформированное по схеме S , представима кортежем

$$U^S = \left\langle \{A^j\}, \{D^j\}, \varphi, T, \delta, U, \gamma \right\rangle. \quad (7)$$

Тогда ПО всех объектов:

$$U^{\bar{S}} = \{U_i^{S_j} : i \in I\}, \quad (8)$$

где $\bar{S} = \{S_j : j \in I\}$.

Схемы отображения модулей ПО на множестве типов и атрибутов построены (см. (7) и (8)). Докажем корректность представления множества модулей, входящих в программное обеспечение АИС, в виде категории.

3. Построение категории на множестве модулей ПО АИС

Категория – наиболее общее или специальное априорное понятие, используемое при построении теорий [4]. В теории категорий – это класс объектов, такой, что при их взаимодействии между собой их структура не меняется, а преобразование происходит посредством морфизмов, т. е. преобразований подобия.

Интуитивно множество модулей ПО – есть категория, поскольку оперирование модулями в операционной системе осуществляется на уровне экземпляров, являющихся копиями (см. например [1]). Рассмотрим аналитическое доказательство данного утверждения.

Допустим A – множество модулей, являющееся категорией. Тогда данное множество должно соответствовать следующим постулатам и аксиомам теории категорий.

Постулаты:

1. Для каждой пары объектов A и B задано множество морфизмов (или стрелок $Hom_S(A, B)$), причём каждому морфизму соответствуют единственные A и B .

2. Для пары морфизмов $f \in Hom(A, B)$ и $g \in Hom(B, C)$ определена композиция $f \circ g \in Hom(A, C)$.

3. Для каждого объекта A задан тождественный морфизм $id_A \in Hom(A, A)$.

Аксиомы:

1. Операция композиции ассоциативна: $h \circ (g \circ f) = (h \circ g) \circ f$.

2. Тождественный морфизм действует тривиально: $f \circ id_A = id_B \circ f = f$ для $f \in Hom(A, B)$.

Докажем, что множество модулей в программном обеспечении автоматизированной информационной системы – категория.

Постулат 1. Учитывая, что в процессе выполнения ПО структура модулей не меняется, то для модулей A и B , между которыми есть отношение, существует морфизм соответствующий единственными A и B .

Постулат 2. Учитывая, что из модуля A можно перейти в модуль B , а оттуда в модуль C , то определена композиция морфизмов из A в C .

На практике это реализуется за счёт создания нового объекта (модуля), нацеленного на согласование модулей.

Постулат 3. Каждый модуль может вызывать собственные подпрограммы и переменные, т. е. обладает тождественным морфизмом.

Аксиома 1. Ассоциативность множества модулей программы обеспечивается наличием пути от A к C как напрямую, так и через B или D .

Ассоциативность также обеспечивается механизмом согласования модулей.

Аксиома 2. Если для модуля A создано тождественное отображение A и при условии, что существует морфизм из A в B , то для модуля B существует и тождественное отображение B (рис. 1). Иными словами, если модули включены в механизм согласования, то оба они обладают тождественным морфизмом.

Таким образом, множество модулей в рамках ПО, для которого задан механизм (модуль) согласования, является категорией.

Поскольку множество модулей программы – категория, то взаимодействие осуществляется на уровне морфизмов. Отношения между модулями не приводят к изменению их структуры и к возникновению новых. Следовательно, система (ПО) полна и непротиворечива.

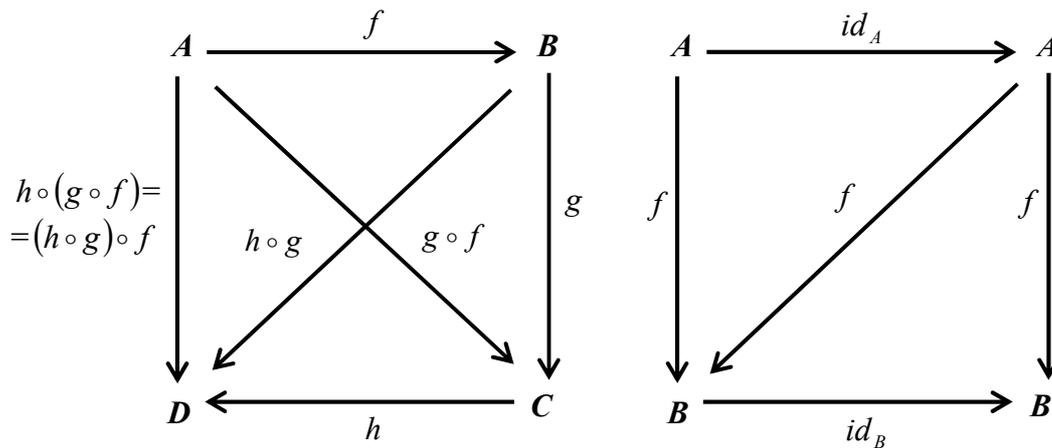


Рис. 1. Ассоциативность композиции и тождественный морфизм для категории «модуль»

Введём понятие «множество ресурсов вычислительной системы». Нетрудно показать, что данное множество также является категорией. Тогда АИС тоже будет категорией, для которой категории «множество модулей» и «множество ресурсов» будут декомпозицией. Очевидно, что для уровня АИС также имеется «множество ресурсов», к которым она имеет доступ с соответствующими правилами и ограничениями, т. е. выводы, сделанные для уровня ПО, будут справедливы и для уровня АИС.

4. Фрактальный подход к программной реализации информационной системы (ИС)

Фракталом [17] может называться предмет, обладающий, по крайней мере, одним из указанных ниже свойств:

- нетривиальной структурой на всех масштабах;
- самоподобие или приближённое самоподобие;
- дробная метрическая размерность или метрическая размерность, превосходящая топологическую.

Математически фрактал определяется как множество с нецелой размерностью [15]. В философском смысле, если представить всю совокупность знаний о предметной области в виде множества информационных миров $S = \{s_i\}, i = \overline{1, \dots, n}$, в котором присутствуют однотипные объекты $l_j = \{P_j\}, j = \overline{1, \dots, m}$ из информационного пространства L , где P_j – совокупность свойств этих объектов, то можно задать множество отображений из любого слоя в каждый слой:

$$F = \{F_j^i : s_i \rightarrow s_j\}, F_j^i = \{f_m^k : l_k \rightarrow l_m\}, \quad (9)$$

где $(s_i, s_j) \in S, l_k \in s_i, l_m \in s_j$ [15].

Аналогичным (9) образом на основе сделанных ранее выводов можно предположить, что программная реализация автоматизированной информационной системы также может быть представлена в виде многоуровневой фрактальной стратифицированной модели, аналогично тому, как это было ранее показано для структурной модели организационной системы [10, 13, 16] и её онтологического представления [6].

Представим АИС в виде иерархии уровней (слоёв) $R = \{R_1, R_2, R_3, R_4, R_5\}$ и рассмотрим композицию категории «Автоматизированная информационная система» на них (см. таблицу).

Информатика и вычислительная техника

Отображения АИС на уровнях иерархии

| Уровни иерархии | Иерархия объектов основных категорий | Основные отображения |
|---------------------------|--|---|
| 1. АИС (R_5) | Множество информационных систем $IS = \{IS_1, \dots, IS_{N_5}\}$, где N_5 – число информационных систем. Множество информационных ресурсов организации $R^{IR} = \{R_1^{IR}, \dots, R_{M_5}^{IR}\}$, где M_5 – число доступных информационных ресурсов и систем в организации | $F_4^5 : R_5 \rightarrow R_4$ – отображение АИС на уровень программ, реализуемое средствами операционной системы; $F_3^5 : R_5 \rightarrow R_3$ – отображение АИС на уровень модуля, реализуемое средствами операционной системы |
| 2. Программы (R_4) | Множество программ $P = \{P_1, \dots, P_{N_4}\}$, где N_4 – число программ. Множество ресурсов системы и локальной сети $R^{Nw} = \{R_1^{Nw}, \dots, R_{M_4}^{Nw}\}$, где M_4 – число доступных ресурсов системы и локальной сети | $F_3^4 : R_4 \rightarrow R_3$ – отображение программы на уровень модулей, реализуемое средствами языка программирования; $F_2^4 : R_4 \rightarrow R_2$ – отображение программы на уровень оператора, реализуемое средствами языка программирования; $F_1^4 : R_4 \rightarrow R_1$ – отображение программы на уровень микрокоманд, реализуемое интерпретаторами языка программирования |
| 3. Модуля (R_3) | Множество модулей $M = \{M_1, \dots, M_{N_3}\}$, где N_3 – число модулей в программе. Множество ресурсов вычислительной системы (компьютера) $R^C = \{R_1^C, \dots, R_{M_3}^C\}$, где M_3 – число доступных ресурсов вычислительной системы | $F_2^3 : R_3 \rightarrow R_2$ – отображение модуля на уровень операторов, реализуемое средствами языка программирования; $F_1^3 : R_3 \rightarrow R_1$ – отображение модуля на уровень микрокоманд, интерпретаторами языка программирования |
| 4. Оператора (R_2) | Множество операторов $Op = \{Op_1, \dots, Op_{N_2}\}$, где N_2 – число операторов, используемых в программе. Множество ресурсов программной среды (языка) $R^P = \{R_1^P, \dots, R_{M_2}^P\}$, где M_2 – число доступных операторов и операций | $F_1^2 : R_2 \rightarrow R_1$ – отображение операторов на уровне микрокоманд, реализуемое интерпретаторами языка программирования |
| 5. Микрокоманды (R_1) | Множество микрокоманд $K = \{K_1, \dots, K_{N_1}\}$, где N_1 – число прерываний. Множество ресурсов программно-аппаратной системы $R^T = \{R_1^T, \dots, R_{M_1}^T\}$, где M_1 – число доступных операторов и операций | – |

Очевидно, что на каждом уровне формируются подобные категории и морфизмы. Следовательно, АИС демонстрирует признаки фрактальной структуры, определяемой правилами подобия композиции.

Покажем эти правила в графической форме на рис. 2.

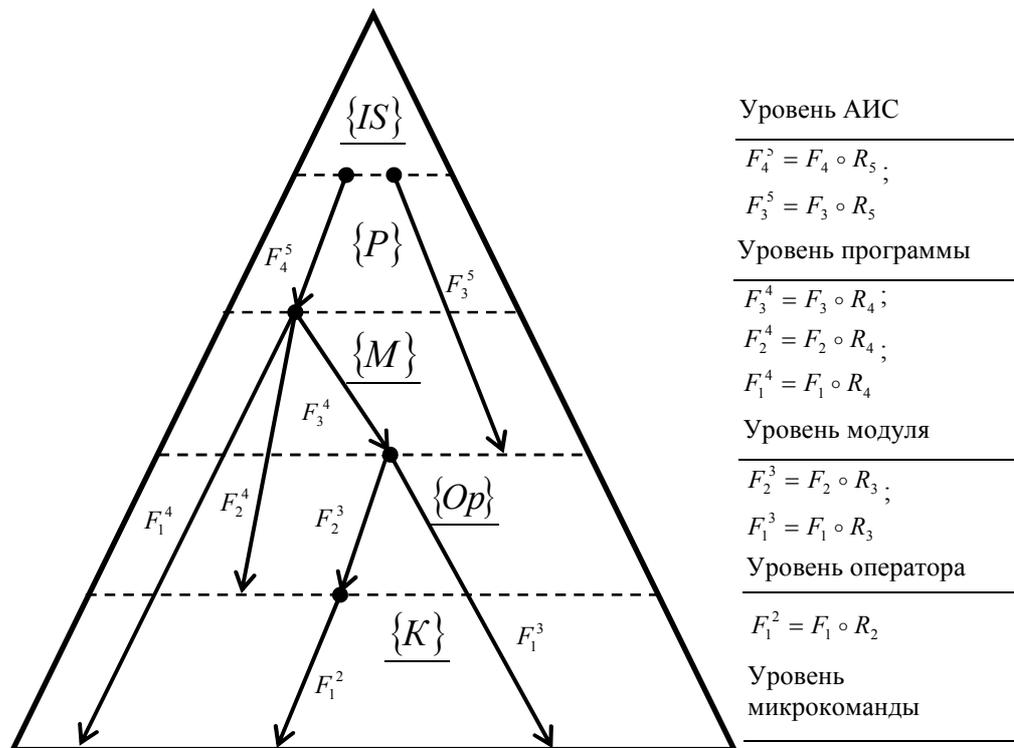


Рис. 2. Пирамида уровней реализации ПО АИС

Из рис. 2 видно, что каждое отображение представимо как композиция элементов своего уровня и отображения одного из нижестоящих уровней.

Допустим, что уровень АИС не крайний. Например, если конкретная АИС выступает компонентом в корпоративной информационной системе (КИС) организации, то можно построить новый уровень – уровень КИС, но при условии наличия сформированных правил взаимодействия разнотипных ИС, которые в неё входят.

Продолжая данную логику, можно предсказать возникновение уровня глобальных ИС, которые будут включать КИС организаций в качестве компонент. Но на нынешнем уровне развития это пока не реализуется по причине отсутствия правил взаимоотношений между КИС различных организаций. Однако в части отдельных КИС такой процесс уже идёт. Так, базы данных должников доступна нескольким банкам, база данных налоговых органов доступна МВД и т. д. Формализация категории «глобальная ИС» возможно будет реализована на основе технологий семантического веба (как платформы) и блокчейн (как заменителя категории «ресурсы»), но рассмотренные данного вопроса выходит за рамки данной статьи.

Заключение

Проблема безопасности сильно ограничивает процесс разработки и внедрения автоматизированных информационных систем в сфере организационного управления. Для решения данной проблемы рассмотрен подход к созданию программного обеспечения, имеющего кластерную структуру, т. е. механизм согласования модулей. В ходе исследования показано, что если представить модуль программного обеспечения в виде категории, то его можно представить как один из уровней изоморфного представления автоматизированной информационной системы. Более того, автоматизированная информационная система может быть представлена как фрактальная структура, последовательно разворачивающаяся от уровня микрокоманд до уровня системы.

Анализ показал, что уровень автоматизированной информационной системы не является крайним. В дальнейшем, по мере накопления общепринятых правил и стандартов, сформируется уровень корпоративных информационных систем как формальная структура из информационных систем разных классов, а затем и глобальные информационные системы.

В дальнейшем планируется подробнее изучить непосредственные механизмы, алгоритмы развёртывания фрактальной структуры, а также направления развития данного подхода и его взаимосвязи с перцептивными информационными технологиями, такими как блокчейн, семантический веб, datamining и т. д.

Литература

1. Антонов, В.В. Метод проектирования адаптивного программного комплекса на основе методологии категорийной формальной модели открытой предметной области / В.В. Антонов // Вестник УГАТУ. – 2015. – Т. 19, № 1. – С. 258–263.
2. Антонов, В.В. Теоретико-множественная модель ИС для многомерного аналитического анализа, отвечающая требованиям хранилища данных / В.В. Антонов, Г.Г. Куликов, Д.В. Антонов // Вестник УГАТУ. – 2012. – Т. 16, № 6 (51). – С. 189–201.
3. Архитектуры и топологии многопроцессорных вычислительных систем / А.В. Богданов, В.В. Корхов, В.В. Мареев, Е.Н. Станкова. – М.: ИНТУИТ.РУ, 2004. – 176 с.
4. Голдблатт, Р. Топосы. Категорный анализ логики / Р. Голдблатт. – М.: Мир, 1983. – 488 с.
5. Дранев, Я.Н. Кластерный подход к экономическому развитию территорий / Я.Н. Дранев // Практика экономического развития территорий: опыт ЕС и России. – М.: Сканрус, 2001. – С. 21–39.
6. Конев, К.А. Концептуальная модель автоматизации предприятия авиационного приборостроения на основе актуализируемой многослойной таксономии / К.А. Конев // Вестник УГАТУ. – 2013. – Т. 17, № 5. – С. 70–77.
7. Конев, К.А. Онтологическая концепция решения задач в прикладных научных исследованиях / К.А. Конев // Информатизация образования и науки. – 2016. – № 1 (29). – С. 147–155.
8. Конев, К.А. Онтологическая специализация области научного знания / К.А. Конев // Информатизация образования и науки. – 2017. – № 3 (36). – С. 95–105.
9. Конев, К.А. Ситуационный подход к управлению рисками-возможностями при обеспечении качества на предприятии авиационного приборостроения / К.А. Конев // Методы менеджмента качества. – 2017. – № 1. – С. 22–28.
10. Конев, К.А. Совершенствование функции управления выпускающей кафедры / К.А. Конев // Методы менеджмента качества. – 2009. – № 3. – С. 26–30.
11. Конев, К.А. Метаситуационные модели: концепция, архитектура, применение в задачах поддержки принятия решений / К.А. Конев, Г.Р. Шакирова // Вестник УГАТУ. – 2014. – Т. 18, № 1 (62). – С. 131–140.
12. Круглов, В.В. Искусственные нейронные сети. Теория и практика / В.В. Круглов, В.В. Борисов. – 2-е изд., стереотип. – М.: Горячая линия – Телеком, 2002. – 382 с.
13. Куликов, Г.Г. Методология управления машиностроительным предприятием на основе интеграции его бизнес-процессов / Г.Г. Куликов, К.А. Конев // Вестник УГАТУ. – 2006. – Т. 7, № 2. – С. 82–91.
14. Куликов, Г.Г. Теория систем и системный анализ: учеб. пособие / Г.Г. Куликов, К.А. Конев, В.А. Суворова. – Уфа: УГАТУ, 2012. – 159 с.
15. Массель, Л.В. Фрактальный подход к структурированию знаний и примеры его применения / Л.В. Массель // Онтология проектирования. – 2016. – Т. 6, № 2 (20). – С. 149–161. DOI: 10.18287/2223-9537-2016-6-2-149-161
16. Управление предприятием с применением системной модели / Е.В. Распопов, Г.И. Погорелов, К.А. Конев, Г.Г. Куликов // Методы менеджмента качества. – 2006. – № 2. – С. 26–30.
17. Федер, Е. Фракталы / Е. Федер. – М.: Мир, 1991. – 254 с.

Куликов Геннадий Григорьевич, д-р техн. наук, профессор, профессор кафедры автоматизированных систем управления, Уфимский государственный авиационный технический университет, г. Уфа; gennadyg_98@yahoo.com.

Антонов Вячеслав Викторович, д-р техн. наук, профессор, заведующий кафедрой автоматизированных систем управления, Уфимский государственный авиационный технический университет, г. Уфа; antonov.v@bashkortostan.ru.

Конев Константин Анатольевич, канд. техн. наук, доцент, доцент кафедры автоматизированных систем управления, Уфимский государственный авиационный технический университет, г. Уфа; sireo@rambler.ru.

Поступила в редакцию 14 декабря 2017 г.

DOI: 10.14529/ctcr180202

CLUSTER SOFTWARE OF AUTOMATED INFORMATION SYSTEM

G.G. Kulikov, gennadyg_98@yahoo.com,
V.V. Antonov, antonov.v@bashkortostan.ru,
K.A. Konev, sireo@rambler.ru

Ufa State Aviation Technical University, Ufa, Russian Federation

In the article, we propose a new approach to creating a clustered software structure that provides the greatest protection against gaining an attacker's control over system support for decision-making, as well as high resistance to corruption and burglary. An approach is considered to solve the problem of software information system security by clustering it into modules. The possibility of formal construction of such structures by methods of set theory and category theory is shown. It is proved that the set of modules satisfies all postulates and axioms of category theory, hence is a category. The conditions for the fractal representation of the automated information system from the level of structural modules of the system to the level of microinstructions with the identification of specific objects and morphisms at hierarchy levels are determined.

Keywords: clustering, fractal structure, set theory, category theory, object, morphism, automated information system, software, module.

References

1. Antonov V. [The Method of Designing an Adaptive Software Package on the Basis of the Methodology of the Category Formal Model of an Open Subject Domain]. *Bulletin of USATU*, 2015, vol. 19, no. 1, pp. 258–263. (in Russ.)
2. Antonov V., Kulikov G., Antonov D. [The Set-Theoretical IP Model for Multidimensional Analytical Analysis, which Meets the Requirements of the Data Warehouse]. *Bulletin of USATU*, 2012, vol. 16, no. 6 (51), pp. 189–201. (in Russ.)
3. Bogdanov A., Korkhov V., Mareev V., Stankova E. *Arkhitektury i topologii mnogoprotsessornykh vychislitel'nykh sistem* [Architecture and Topology of Multiprocessor Computer Systems]. Moscow, INTUIT.RU Publ., 2004. 176 p.
4. Goldblatt R. *Toposy. Kategornyj analiz logiki* [Topos. Categorical Analysis of Logic]. Moscow, Mir Publ., 1983. 488 p.
5. Dranev Y. *Klasternyy podkhod k ekonomicheskomu razvitiyu territoriy* [Cluster Approach to the Economic Development of Territories]. *Praktika ekonomicheskogo razvitiya territoriy: opyt ES i Rossii* [Practice of Economic Development of Territories: the Experience of the EU and Russia]. Moscow, Scanrus Publ., 2001, pp. 21–39.
6. Konev K. [Conceptual Model of Automation of the Enterprise of Aviation Instrument Making

on the Basis of the Actualized Multi-Layer Taxonomy]. *Bulletin of the USATU*, 2013, vol. 17, no. 5, pp. 70–77. (in Russ.)

7. Konev K. [Ontological Concept of Solving Problems in Applied Scientific Research]. *Informatization of Education and Science*, 2016, no. 1 (29), pp. 147–155. (in Russ.)

8. Konev K. [Ontological Specialization of the Field of Scientific Knowledge]. *Informatization of Education and Science*, 2017, no. 3 (36), pp. 95–105. (in Russ.)

9. Konev K. [A Situational Approach to Managing Risk-opportunities while Ensuring Quality at the Enterprise of Aircraft Instrument Making]. *Methods of Quality Management*, 2017, no. 1, pp. 22–28. (in Russ.)

10. Konev K. [Perfection of the Management Function of the Graduating Department]. *Quality Management Methods*, 2009, no. 3, pp. 26–30. (in Russ.)

11. Konev K., Shakirova G. [Metasituational Models: Concept, Architecture, Application in Decision Support Tasks]. *Bulletin of USATU*, 2014, vol 18, no. 1 (62), pp. 131–140. (in Russ.)

12. Kruglov V., Borisov V. *Iskusstvennye neyronnye seti. Teoriya i praktika* [Artificial Neural Networks. Theory and Practice]. Moscow, Hot Line-Telecom Publ., 2002. 382 p.

13. Kulikov G., Konev K. [Methodology of the Management of a Machine-Building Enterprise on the Basis of Integration of its Business Processes]. *Bulletin of USATU*, 2006, vol. 7, no. 2, pp. 82–91. (in Russ.)

14. Kulikov G., Konev K., Suvorova V. *Teoriya sistem i sistemnyy analiz: ucheb. posobie* [Theory of Systems and Systems Analysis: Textbook]. Ufa, USATU Publ., 2012. 159 p.

15. Mussel L. [Fractal Approach to the Structuring of Knowledge and Examples of its Application]. *Ontology of Design*, 2016, vol. 6, no. 2 (20), pp. 149–161. (in Russ.) DOI: 10.18287/2223-9537-2016-6-2-149-161

16. Raspopov E., Pogorelov G., Konev K., Kulikov G. [Management of the Enterprise Using the System Model]. *Methods of Quality Management*, 2006, no. 2, pp. 26–30. (in Russ.)

17. Feder E. *Fraktaly* [Fractals]. Moscow, Mir Publ., 1991. 254 p.

Received 14 December 2017

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Куликов, Г.Г. Кластерное программное обеспечение автоматизированной информационной системы / Г.Г. Куликов, В.В. Антонов, К.А. Конеv // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». – 2018. – Т. 18, № 2. – С. 19–28. DOI: 10.14529/ctcr180202

FOR CITATION

Kulikov G.G., Antonov V.V., Konev K.A. Cluster Software of Automated Information System. *Bulletin of the South Ural State University. Ser. Computer Technologies, Automatic Control, Radio Electronics*, 2018, vol. 18, no. 2, pp. 19–28. (in Russ.) DOI: 10.14529/ctcr180202