

На правах рукописи

Кольчугина

КОЛЬЧУГИНА Елена Анатольевна

**ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПОСТРОЕНИЯ
САМООРГАНИЗУЮЩИХСЯ ПРОГРАММНЫХ СИСТЕМ
С САМООРГАНИЗАЦИЕЙ КОНТИНУАЛЬНОГО ТИПА**

Специальность 05.13.17 – Теоретические основы информатики

А в т о р е ф е р а т
диссертации на соискание ученой степени
доктора технических наук

Пенза 2017

Работа выполнена в Федеральном государственном бюджетном образовательном учреждении высшего образования (ФГБОУ ВО) «Пензенский государственный университет» на кафедре «Математическое обеспечение и применение ЭВМ».

Научный консультант – доктор технических наук, профессор
Горбаченко Владимир Иванович

Официальные оппоненты: **Курейчик Виктор Михайлович**,
доктор технических наук, профессор,
ФГАОУ ВО «Южный федеральный университет», «Институт компьютерных технологий и информационной безопасности» (ИКТИБ ЮФУ), главный научный сотрудник, профессор кафедры «Системы автоматизированного проектирования»

Титов Виталий Семенович,
доктор технических наук, профессор, ФГБОУ
ВО «Юго-Западный государственный
университет», кафедра «Вычислительная
техника», заведующий кафедрой

Бабич Михаил Юрьевич
доктор технических наук, доцент, АО «НПП
"Рубин"», главный специалист

Ведущая организация: **АО «Пензенский научно-исследовательский электротехнический институт»**

Защита диссертации состоится 28 сентября 2017 г. в ___ часов на заседании диссертационного совета Д 212.186.01 на базе ФГБОУ ВО «Пензенский государственный университет» по адресу: 440026, г. Пенза, ул. Красная, 40.

С диссертацией и авторефератом можно ознакомиться в библиотеке ФГБОУ ВО «Пензенский государственный университет» и на сайте http://dissov.pnzgu.ru/ecspertiza/Tehnicheskie_nauki/kolchugina

Автореферат разослан «___» июня 2017 г.

Ученый секретарь
диссертационного совета

Гурин Евгений Иванович

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность. Сложность современного программного обеспечения объективно возрастает. Понятие сложности может быть рассмотрено в нескольких аспектах. Со структурно-функциональной точки зрения возрастание сложности означает расширение функциональных возможностей программного обеспечения и, как следствие, существенное увеличение объема программного кода. С концептуально-семантической точки зрения возрастание сложности означает непрерывное увеличение числа и разнообразия задач, решаемых с помощью программного обеспечения, а также возрастание сложности математических моделей, реализуемых программным обеспечением. Наконец, с технологической точки зрения возрастание сложности программного обеспечения означает рост ресурсоемкости процессов разработки, эксплуатации и сопровождения программного обеспечения. Важнейший вклад в возрастание ресурсоемкости вносит возрастание трудоемкости.

Применение человеческого труда при разработке программного обеспечения становится неэффективным в отсутствие автоматизации.

Концептуальная сложность программных проектов, необходимость выдерживать высокий темп разработки, недостаточно полное осмысление задачи разработчиком и человеческая невнимательность, проявленная на этапе реализации, приводят к недоработкам и ошибкам, которые трудно выявить. Устранение ошибок и совершенствование уже существующего программного обеспечения могут потребовать перепроектирования, что по стоимости может быть сопоставимо с разработкой нового проекта. Кроме того, любой программный продукт требует дополнительных трудозатрат на свое сопровождение в течение всего жизненного цикла.

В силу этих причин актуальной потребностью становится снижение влияния человеческого фактора при разработке программ. Процесс разработки программного обеспечения должен быть автоматизирован, а готовое программное обеспечение должно быть способно к самосовершенствованию в процессе эксплуатации, то есть быть способным к самостоятельному устранению ошибок или даже развитию новых функциональных возможностей под воздействием условий актуальной вычислительной среды. В идеальном случае программная система должна самостоятельно развиваться из некоторой универсальной зародышевой структуры в соответствии с поставленными требованиями, потоком задач пользователя и особенностями организации текущей среды выполнения.

Способности к саморазвитию, самовосстановлению и самосовершенствованию являются отличительными свойствами живых организмов и представляют собой проявления самоорганизации¹. Теория искусственной жизни

¹ Самоорганизация – в диссертации определяется как неравновесное внутреннее самоупорядочивание, проявляющееся в том числе в виде спонтанного нарушения симметрии среды и образования структур в ответ на неспецифическое внешнее воздействие.

является именно тем научным направлением, которое преследует цели исследования процессов самоорганизации в биологических системах с целью воспроизведения их в технической среде.

Теория искусственной жизни как научное направление сформировалась в 80–90-х гг. XX в. на основе результатов работ таких исследователей, как C. G. Langton, S. A. Kauffman, J. D. Farmer, M. Mitchell, L. J. Yaeger, T. S. Ray, W. Fontana, K. Sims, M. A. Bedau, J. S. McCaskill, M. Sipper, N. H. Packard, S. Rasmussen, C. Adami, K. Kaneko и других. В России исследования в данном направлении проводятся под руководством В. Г. Редько, близкими по тематике являются работы таких исследователей, как В. М. Курейчик, В. В. Курейчик, Л. А. Гладков, Н. П. Вашкевич, В. С. Титов, М. Ю. Бабич.

Среди моделей теории искусственной жизни выделяется класс моделей искусственной химии. Многие модели этого класса представляют собой аналоги автокаталитических систем. Автокаталитические системы живой природы и их организованные совокупности, например гиперциклы, являются примером достаточно хорошо изученных систем с континуальным типом самоорганизации.

Понятие континуальной самоорганизации, наряду с когерентной, было введено в работах А. П. Руденко. Более простая когерентная самоорганизация предполагает согласование действий однородных микросистем, результатом чего становится проявление системных эффектов на макроуровне. Такая самоорганизация наблюдается в классических мультиагентных системах. Возможности когерентной самоорганизации ограничены, прогрессивная эволюция здесь отсутствует. Напротив, отличительным и уникальным свойством систем с континуальной самоорганизацией является их способность к прогрессивной эволюции и образованию на микроуровне структур с качественно новыми, неизвестными ранее свойствами на основе динамического взаимодействия разнородных компонентов. В терминологии химии континуальная самоорганизация связана с образованием промежуточного функционально неделимого объекта в виде кинетического континуума веществ и процессов.

Многие модели искусственной химии используют в качестве метафоры уподобление программ биомолекулам, способным к автокатализу. Эти модели призваны исследовать эволюцию и образование качественно новых программных структур в результате функционирования и взаимодействия исходно заданного множества программ. Преобразования программ рассматриваются как аналоги химических реакций.

Данное направление основано на работах M. Eigen, P. Schuster, W. Fontana, L. W. Buss, S. Rasmussen, T. S. Ray, C. Adami, J. S. McCaskill, P. Dittrich, J. Ziegler, W. Banzhaf, P. Speroni di Fenizio и других.

Модели искусственной химии, таким образом, являются наиболее вероятными претендентами на роль прототипа практически полезных программных систем с самоорганизацией континуального типа. Однако известные модели этого класса, такие как «*Tierra*» (T. S. Ray и соавторы) и «*Avida*»

(С. Adami и соавторы), носят исключительно исследовательский характер и не могут использоваться в прикладных целях. Эти модели ориентированы на использование усеченных узкоспециальных версий языка ассемблера, а не на использование языков программирования высокого уровня.

Следовательно, проблема создания практически полезных самоорганизующихся программных систем с самоорганизацией континуального типа, подходящих для решения системных и прикладных задач, в том числе формулировка и обоснование основных теоретических принципов построения таких систем, а также разработка методов и средств их реализации, является актуальной.

Исследование основано на следующих исходных положениях и принципах.

1. Самоорганизующиеся программы рассматриваются как системы, состоящие из произвольного числа независимых программных единиц. В процессе функционирования количественный и качественный состав этих единиц и их взаимоотношения друг с другом могут изменяться.

2. Каждая независимая программная единица представляет собой набор функций, состав и структура которого динамически изменяются. Наличие ряда функций является обязательным (без них программа не сможет работать), но если семантика таких функций постоянна, их реализации могут изменяться. Некоторые функции являются опционными и могут присутствовать или отсутствовать. Реализации любых функций и их взаимодействия друг с другом могут изменяться.

3. Самоорганизующиеся программы функционируют и развиваются в условиях искусственно созданной замкнутой среды в соответствии с правилами и ограничениями, заданными для этой среды и функционирующих в ней программных единиц.

4. На всех уровнях организации самоорганизующихся программ присутствует и используется взаимная двойственность программного кода и данных (программа записывается в виде данных, которые могут быть обработаны; данные могут представлять собой запись программы, которая может быть выполнена). На уровне независимой программной единицы программный код и данные обобщенно рассматриваются как функции. Вся самоорганизующаяся программа может рассматриваться и как результат сборки на основе компонентов базы данных, и как данные, подлежащие сохранению в базе. В свою очередь база данных может рассматриваться как сжатая архивированная форма представления программы.

Объектом исследования данной диссертационной работы являются самоорганизующиеся программные системы с самоорганизацией континуального типа, функционирующие на основе распределенных вычислительных систем и сетей.

Предметом исследования являются:

– методы и средства построения самоорганизующихся программных систем с самоорганизацией континуального типа, позволяющие обеспечить

протекание процессов самоорганизации и возникновение новых программных структур, в том числе методы кодирования самоорганизующихся программных структур, а также способы и методы построения сред, в которых осуществляется самоорганизация программ;

– процессы самоорганизации программ, в том числе образование последовательных и параллельных программных структур, устойчивость поведения структур во времени, образование иерархических сообществ программных структур.

Целью исследования является формулировка и обоснование теоретических положений, основополагающих для построения самоорганизующихся практически значимых программных систем на основе бионических принципов (используются принципы молекулярной биологии).

Для достижения поставленной цели необходимо решить следующие основные задачи:

1) выполнить анализ и построить классификацию программ, способных к изменению своего поведения на стадии эксплуатации, и моделей, подходящих для создания самоорганизующихся программ с самоорганизацией континуального типа;

2) используя метафору молекулярного организма, построить функциональную модель самоорганизующейся программной единицы как множества взаимосвязанных процессов;

3) разработать структурную модель цифрового организма¹ как самоорганизующейся программной единицы; разработать структуру и способ кодирования цифровой ДНК²; определить операции над цифровой ДНК;

4) разработать способ кодирования параллельного алгоритма поведения цифрового организма с целью его размещения в цифровой ДНК; разработать операции над структурами, кодирующими параллельный алгоритм поведения, с целью получения новых параллельных алгоритмов;

5) исследовать устойчивость поведения цифрового организма как последовательность смены его состояний; с этой целью предложить *NK*-подобную автоматную модель, являющуюся модификацией *NK*-автоматов С. А. Кауффмана, в которой множество элементов будет разбито на два подмножества, представляющих данные и фрагменты программного кода, и будет присутствовать управляющий элемент, задающий программу поведения автомата;

6) с использованием методов теории нумераций разработать методы свертки цифровой ДНК с целью сохранения истории эволюции самоорганизующейся программной системы в базах данных;

¹ Цифровой организм – общепринятый в теории искусственной жизни термин; введен Т. С. Рэем (T. S. Ray) для самокопирующихся ассемблерных программ в модели «*Tierra*».

² Цифровая ДНК – общепринятый в теории искусственной жизни термин, также введен Т. С. Рэем; обозначает структуру, кодирующую основные свойства, в том числе и поведенческие, цифрового организма, подобно ДНК биологических организмов в живой природе.

7) разработать новую концепцию распределенного информационного поиска на основе распределенных баз данных и сообщества цифровых организмов;

8) разработать принципы организации замкнутых сред (искусственных миров), в которых существуют и развиваются цифровые организмы и их сообщества;

9) с точки зрения популяционной динамики исследовать различные типы отношений между видами цифровых организмов и различные модели сообществ, построенные на основе межвидовых отношений;

10) разработать структуру программного обеспечения, необходимого для реализации программной платформы, создающей замкнутую среду, в которой существуют организмы, а также для реализации отдельных цифровых организмов;

11) разработать рекомендации по проектированию программных систем в зависимости от требуемых свойств.

Методы исследования. Для решения поставленных задач использовались методы теории множеств, теории автоматов (*NK*- и *NK*-подобные автоматы), теории графов, теории формальных языков и грамматик, генетических алгоритмов, теории нумераций, математической статистики, математической биологии.

Научная новизна диссертации определяется новизной постановки задачи исследования и следующими основными результатами.

1. На основе бионических принципов предложена парадигма неравновесного программирования, *отличающаяся* тем, что в ее рамках программа рассматривается как имеющая изменяющуюся в ходе функционирования структуру совокупность процессов, связанных между собой по исходным данным и результатам вычислений, и *позволяющая* разрабатывать на языках высокого уровня самоорганизующиеся программы с самоорганизацией континуального типа.

2. Предложена функциональная модель цифрового организма, *отличающаяся* тем, что в ней цифровой организм определяется как совокупность взаимосвязанных функций, и *позволяющая* избежать возникновения проблемы хрупкости, или потери семантической значимости, программного кода в процессе эволюции программ.

3. Предложена структурная модель цифрового организма, *отличающаяся* тем, что цифровой организм рассматривается как совокупность отдельно представленных генотипа, заданного цифровой ДНК, и фенотипа, как совокупности данных и программного кода, и *позволяющая* описать программу в терминах унифицированных структурных элементов, а как следствие, предложить технологию программной реализации цифрового организма как самоорганизующейся программной единицы.

4. Разработан метод кодирования алгоритма поведения и свойств цифрового организма в виде цифровой ДНК, *отличающийся* тем, что с помощью

этого метода единообразно, в виде числовой последовательности, представляются как фрагменты программного кода, входящие в состав реализации алгоритма поведения, так и данные, представляющие свойства и текущее состояние цифрового организма, и *позволяющий* унифицировать выполнение генетических операций над данными и фрагментами программного кода, вследствие чего автоматизируется процесс разработки программ, обеспечивается эволюция программного обеспечения в процессе его эксплуатации, что приводит к снижению трудоемкости разработки и повышению ее качества благодаря снижению влияния человеческого фактора.

5. Разработан метод кодирования в составе цифровой ДНК управляющих структур, представляющих параллельные версии алгоритма поведения цифрового организма, *отличающийся* тем, что в его основе лежит использование матрицы, задающей подмножество возможных параллельных алгоритмов поведения, в сочетании с формулой, задающей правило прочтения разбиения матрицы, и *позволяющий* автоматизировать разработку параллельных алгоритмов при помощи генетических операций над цифровой ДНК и преобразований матрицы расписания, а также преобразований, порождающих новые правила прочтения матрицы; в результате автоматизируется процесс разработки параллельных программ, снижается его трудоемкость и повышается качество разработки программ благодаря снижению влияния человеческого фактора.

6. Предложена НК-подобная автоматная модель цифрового организма, *отличающаяся* от НК-модели С. А. Кауффмана тем, что: а) множество элементов автомата разбито на два непересекающихся подмножества; б) имеется управляющий элемент, задающий программу поведения автомата, которая может быть описана ярусно-параллельной формой алгоритма с высотой более одного яруса, и *позволяющая* исследовать динамические свойства цифрового организма с точки зрения устойчивости поведения, оцениваемой на основе длины периода последовательности смены состояний автомата.

7. Предложены методы кодирования ДНК-структур цифровых организмов при помощи функций свертки теории нумераций, *отличающиеся* тем, что числовая последовательность, представляющая ДНК цифрового организма, в результате применения функций свертки заменяется одним целым числом, и *позволяющие* компактно сохранять в базах данных информацию о последовательности генов цифрового организма, а также выполнять операции над цифровыми организмами путем преобразования сверток их ДНК-структур и восстановления последовательностей из сверток; это в свою очередь позволяет исследовать эволюцию программной системы на основании информации из баз данных, а также автоматизировать генерацию новых программ на основе кодированных представлений уже имеющихся программ.

8. Разработан новый метод организации распределенного информационного поиска в растущих доменно-ориентированных базах данных с покортежным пространственным распределением отношений, реализуемый само-

организуемым сообществом цифровых организмов, *отличающийся* ориентацией на использование уникальных внутрисистемных идентификаторов значений из заданных доменов, и *позволяющий* выполнять информационный поиск по таблицам отношений произвольной структуры вне зависимости от их пространственного расположения и структуры и даже в отсутствие информации о пространственном расположении данных и структуре отношений, что расширяет информационно-поисковые возможности распределенных систем.

9. Разработаны принципы организации замкнутых сред, или искусственных миров, для существования самоорганизующихся программных систем как сообществ цифровых организмов, *отличающиеся* послойной пространственной организацией, что *позволяет* создавать структурированные сообщества цифровых организмов со стратификацией по функциональному назначению. Стратификация обеспечивает прозрачность структуры и управляемость сообщества цифровых организмов.

10. Впервые разработана структура системного программного обеспечения, необходимого для реализации предложенной парадигмы неравновесного программирования и создания самоорганизующихся программ, *отличающегося* клиент-серверной организацией, использованием параллельных ветвлений и программных шаблонов, и *позволяющего* выполнять контроль свойств цифровых организмов, их периодическую пересборку (реактуализацию) и запуск на выполнение и тем самым не только динамически генерировать самоорганизующиеся программные системы, но и полностью контролировать их.

11. На основе результатов проведенных экспериментальных исследований даны рекомендации по проектированию самоорганизующихся программных систем, *отличающиеся* тем, что в рекомендациях перечисляются элементы поведения цифровых организмов, необходимые для создания программных систем с требуемыми свойствами, что *позволяет* снизить трудозатраты на проведение проектирования программного обеспечения и повысить качество проектирования.

Практическая ценность результатов диссертационной работы состоит в следующем.

1. Предложенная парадигма программирования позволяет унифицировать и стандартизировать процесс создания самоорганизующихся программных систем.

2. Предложенные функциональная и структурная модели цифрового организма определяют технологию создания самоорганизующейся программной единицы как совокупности, состоящей: а) из набора функций, для которых необходимо обеспечить набор возможных программных реализаций; б) из отношений между функциями, которые требуется задать при разработке программной единицы.

3. Предложенный метод кодирования ДНК цифрового организма позволяет применить по отношению к цифровой ДНК методы оптимизации для по-

лучения наилучших программных решений, то есть автоматизировать процесс программирования. Поиск наилучших решений может производиться как на этапе разработки самоорганизующейся программной системы, так и на этапе ее эксплуатации.

4. Предложенный метод кодирования ДНК цифрового организма позволяет быстро определять свойства цифровых организмов на основе анализа цифровой ДНК с целью удаления организмов с нежелательными свойствами.

5. Предложенный метод кодирования параллельного алгоритма поведения цифрового организма также позволяет применить методы оптимизации для получения наилучшего решения. Предложенный метод кодирования может быть применен как в рамках парадигмы неравновесного программирования, так и отдельно от нее для разработки параллельных программ. На основе предложенного способа кодирования введен новый метод разработки параллельных алгоритмов, применяемый для создания алгоритмов поведения цифровых организмов.

6. Предложенная *NK*-подобная автоматная модель позволяет исследовать динамические свойства цифровых организмов как на стадии формирования начальной популяции с целью выбора лучших исходных решений, так и на стадии анализа результатов работы самоорганизующейся программной системы.

7. Предложенный метод кодирования ДНК-структур позволяет, благодаря применению методов свертки теории нумераций, сохранять в базах данных цифровые организмы и проследивать историю их изменений. Таким образом, вся самоорганизующаяся программная система может быть сохранена в базе данных и впоследствии восстановлена. Это позволяет сохранять удачные решения и неограниченно их тиражировать, а также автоматически генерировать новые программные решения на основе имеющихся программ.

8. Предложенный метод организации растущих баз данных на основе доменно-ориентированной модели с покортежным и поддоменным пространственным распределением отношений позволяет обеспечить хранение неограниченно больших массивов данных на основе вычислительных сетей с автоматизацией распределения отношений по узлам вычислительных сетей. В сочетании с самоорганизующимися программами, реализованными как коллективы цифровых организмов, обслуживающих растущие базы данных, предложенный метод организации баз данных позволяет выполнять новый вид запросов – расширенные запросы, которые приближают поиск информации в базах данных к релевантному поиску в Интернете. Практическая ценность таких запросов состоит в том, что они предоставляют большее количество информации по сравнению с обычными запросами.

9. Разработанные принципы организации искусственных миров, являющихся средой функционирования и развития самоорганизующегося программного обеспечения, а также принципы организации сообществ цифровых организмов, позволяют стандартизировать и автоматизировать процедуру со-

здания искусственных миров и унифицировать правила генерации сообществ цифровых организмов, что способствует уменьшению временных затрат на разработку самоорганизующихся программ и повышению ее качества.

10. Предложенная структура программного обеспечения задает прототип, в соответствии с которым могут быть как построены серверные компоненты программного обеспечения, реализующие среду виртуальных ЭВМ, так и реализовано программное обеспечение, представляющее отдельные цифровые организмы. Таким образом, формализуются принципы построения соответствующего программного обеспечения, что позволяет обеспечить автоматизацию, унификацию и стандартизацию процесса разработки.

11. Рекомендации по проектированию самоорганизующихся программных систем позволяют проектировать программные системы с заранее известными, предсказуемыми и требуемыми свойствами, при этом снижается трудоемкость проектирования и повышается его качество. Для программной системы, реализованной в соответствии с парадигмой неравновесного программирования, отмечено снижение трудоемкости: для одного элементарного пакета работ в $\frac{E_{2i}}{E_{1i}} \approx 30,6$ раза; для всего проекта в $\frac{E_2^{95\%}}{E_1^{95\%}} \approx 79,73$ раза; количество строк исходного кода уменьшилось в 17 раз (по сравнению с реализацией с использованием парадигмы агент-ориентированного программирования).

Достоверность результатов исследования подтверждается:

– целостностью и внутренней непротиворечивостью введенных теоретических положений и их согласованностью с результатами современных исследований в области технических, физико-математических и естественных наук;

– логической обоснованностью и корректностью применения методов теории множеств, теории автоматов (*NK*- и *NK*-подобные автоматы), теории графов, теории формальных языков и грамматик, генетических алгоритмов, теории нумераций, математической статистики, математической биологии;

– воспроизводимостью, повторяемостью и непротиворечивостью данных, полученных в ходе экспериментальной проверки результатов исследований.

Основные положения, выносимые на защиту:

1. Парадигма неравновесного программирования, *позволяющая* разрабатывать самоорганизующиеся программы с самоорганизацией континуального типа.

2. Функциональная модель цифрового организма, *позволяющая* избежать возникновения проблемы хрупкости, или потери семантической значимости, программного кода в процессе эволюции программ.

3. Структурная модель цифрового организма, *позволяющая* предложить технологию программной реализации цифрового организма как самоорганизующейся программной единицы.

4. Метод кодирования алгоритма поведения и свойств цифрового организма в виде структуры цифровой ДНК, *позволяющий* унифицировать выпол-

нение генетических операций над данными и фрагментами программного кода, вследствие чего автоматизируется процесс разработки программ, обеспечивается эволюция программного обеспечения в процессе его эксплуатации, что приводит к снижению трудоемкости разработки и повышению ее качества благодаря снижению влияния человеческого фактора.

5. Метод кодирования в составе цифровой ДНК управляющих структур, представляющих параллельные версии алгоритма поведения цифрового организма, *позволяющий* автоматизировать разработку параллельных алгоритмов при помощи генетических операций над цифровой ДНК и преобразований матрицы расписания, а также преобразований, порождающих новые правила прочтения матрицы; в результате автоматизируется процесс разработки параллельных программ, снижается его трудоемкость и повышается качество разработки программ благодаря снижению влияния человеческого фактора.

6. НК-подобная автоматная модель цифрового организма, *позволяющая* исследовать динамические свойства цифрового организма с точки зрения устойчивости поведения, оцениваемой на основе длины периода последовательности смены состояний автомата, с учетом различных значений высоты управляющего алгоритма.

7. Методы кодирования ДНК-структур цифровых организмов при помощи функций свертки теории нумераций, *позволяющие* компактно сохранять в базах данных информацию о последовательности генов цифрового организма, а также выполнять операции над цифровыми организмами путем преобразования сверток их ДНК-структур и восстановления последовательностей из сверток, что позволяет исследовать эволюцию программной системы на основании информации из баз данных, а также автоматизировать генерацию новых программ на основе кодированных представлений уже имеющихся программ.

8. Метод организации распределенного информационного поиска в растущих доменно-ориентированных базах данных с покортежным пространственным распределением отношений, реализуемый самоорганизующимся сообществом цифровых организмов, *позволяющий* выполнять информационный поиск по таблицам отношений произвольной структуры и вне зависимости от их пространственного расположения и даже в отсутствие информации о пространственном расположении данных, что расширяет возможности информационного поиска в распределенных базах данных.

9. Принципы организации замкнутых сред, или искусственных миров, для существования самоорганизующихся программных систем как сообществ цифровых организмов, *позволяющие* создавать структурированные сообщества цифровых организмов со стратификацией по функциональному назначению. Стратификация обеспечивает прозрачность структуры и управляемость сообщества цифровых организмов.

10. Структура системного программного обеспечения, необходимого для реализации предложенной парадигмы неравновесного программирования и создания самоорганизующихся программ, *позволяющего* выполнять кон-

троль свойств цифровых организмов, их периодическую пересборку (реактуализацию) и запуск на выполнение, то есть динамически генерировать самоорганизующиеся программные системы и полностью контролировать их.

11. Рекомендации по проектированию самоорганизующихся программных систем, *позволяющие* снизить трудозатраты на проведение проектирования программного обеспечения и повысить качество проектирования.

Соответствие диссертации паспорту научной специальности

Основная идея, развиваемая в данном исследовании, состоит в использовании бионических принципов, то есть принципов организации и функционирования, присущих биологическим объектам и их совокупностям, в технологиях распределенной обработки информации, а также технологиях создания программного обеспечения, предназначенного для распределенной обработки информации. Это соответствует п. 13 паспорта специальности 05.13.17.

Основная часть работы посвящена разработке теоретических основ создания программных систем, пригодных для реализации новой информационной технологии. Это соответствует п. 14 паспорта специальности 05.13.17.

В диссертационной работе предложена новая технология организации распределения и поиска информации в распределенных растущих базах данных, которая может быть реализована на базе сети Интернет. Это соответствует п. 9 паспорта специальности 05.13.17.

В диссертации были предложены метод кодирования алгоритма поведения и свойств цифрового организма в виде структуры цифровой ДНК, а также метод кодирования в составе цифровой ДНК управляющих структур, представляющих параллельные версии алгоритма поведения цифрового организма. Также был предложен метод кодирования ДНК-структур путем применения методов свертки теории нумераций, что позволяет сохранять в базах данных цифровые организмы и прослеживать историю их изменений. Разработка перечисленных методов кодирования соответствует п. 3 паспорта специальности 05.13.17.

В ходе исследований также была предложена автоматная модель нового типа, а именно НК-подобная модель, и исследованы ее свойства. Это соответствует п. 10 паспорта специальности 05.13.17.

Реализация и внедрение результатов диссертационной работы. Диссертационная работа выполнена на кафедре «Математическое обеспечение и применение ЭВМ» Пензенского государственного университета (ПГУ).

Результаты диссертации внедрены в Научно-исследовательском и конструкторском институте радиоэлектронной техники – филиале ФГУП ФНПЦ «ПО "Старт" им. М. В. Проценко» (НИКИРЭТ – филиал ФГУП ФНПЦ «ПО "Старт" им. М. В. Проценко»), ООО НТП «Криптософт», ООО НПП «Контех», АО «ПО "Электроприбор"». Все внедрения подтверждены соответствующими актами. Результаты докторской диссертации были использованы в деятельности предприятия АО «Научно-производственное предприятие "Рубин"». Акт реализации результатов прилагается к диссертации.

Теоретические и практические результаты диссертационной работы были внедрены при выполнении фундаментальной НИР № 430 «Развитие теории организации взаимодействия параллельных процессов в распределенных системах обработки данных» (2012–2013 гг., № гос. рег. 01201257166), выполненной в рамках государственного задания в ПГУ на кафедре «Вычислительная техника». Имеется акт внедрения.

На программное обеспечение, созданное в результате диссертационной работы и представляющее собой инструментальные средства для создания самоорганизующихся программных систем, получены свидетельства об официальной регистрации в ФИПС.

Научные и практические результаты работы используются при подготовке и проведении ряда учебных дисциплин на кафедре «Вычислительная техника» Пензенского государственного университета (подтверждено актом внедрения).

Апробация работы. Результаты исследования были доложены и получили одобрение на следующих научных конференциях: VI, VII, VIII, IX и XI Международных научно-технических конференциях «Новые информационные технологии и системы» (Пенза, 2004, 2006, 2008, 2010 и 2014); Международной научно-технической конференции «Современные информационные технологии (Computer-based conference)» (Пенза, 2004, весенняя и осенняя сессии; 2005, весенняя и осенняя сессии; 2006, весенняя и осенняя сессии); IX и X Международных научно-методических конференциях «Университетское образование» (Пенза, 2005 и 2006); I Международной конференции «Аналитические и численные методы моделирования естественнонаучных и социальных проблем» (Пенза, 2006); XIV Международной конференции студентов, аспирантов и молодых ученых «Ломоносов-2007» (Москва, 2007); XV, XVI, XVII и XVIII Международных конференциях «Математика. Компьютер. Образование» (Дубна Московской области, 2008, 2010 и 2011; Пушкино Московской области, 2009); XVII и XXIII Всероссийских семинарах «Нейроинформатика, ее приложения и анализ данных» (Красноярск, 2009 и 2015); Международной научно-практической конференции «Современные тенденции в образовании и науке» (Тамбов, 2014); Международной научно-практической конференции «Проблемы развития современной науки» (Уфа, 2015); Международной научно-практической конференции «Роль науки в развитии общества» (Казань, 2015); V Международной научной конференции «Технические науки в России и за рубежом» (Москва, 2016); XVI Международной научно-методической конференции «Информатика: проблемы, методология, технологии» (Воронеж, 2016); X European Conference on Innovations in Technical and Natural Sciences (Vienna, Austria, 2016); International Conference «Science and Practice: a New Level of Integration in the Modern World» (Sheffield, UK, 2016).

Публикации. По теме диссертации опубликовано 60 научных работ, в том числе статьи в журналах (из них одна статья опубликована в издании,

входящем в систему цитирования SCOPUS, 16 статей – в журналах, рекомендованных ВАК РФ), сборниках научных трудов, трудах конференций. Получено два свидетельства о регистрации программ для ЭВМ. В автореферате перечислены наиболее существенные публикации.

Все результаты, составляющие содержание диссертации, получены автором самостоятельно.

Структура и объем диссертационной работы. Общий объем диссертационной работы составляет 295 страниц. Работа состоит из введения, шести глав, заключения, списка литературы из 254 наименований, 2 приложений, 43 рисунков и 10 таблиц. В приложении А приведен список используемых математических обозначений. В приложении Б приведены акты внедрения и реализации результатов диссертации.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении обосновывается актуальность темы диссертации, определяются предмет и объект исследования, формулируются цели исследования, определяются новизна и практическая ценность диссертационной работы.

В первой главе вводится и рассматривается понятие изменяющихся программ, способных совершенствовать свое поведение в процессе эксплуатации в практически значимых и полезных целях. Вводится классификация изменяющихся программ, среди них выделен класс самоорганизующихся программ. Выделены свойства самоорганизующихся программ: 1) самоорганизующиеся программы являются системными объектами, обладающими структурой; 2) самоорганизующиеся программы способны к упорядочиванию и переупорядочиванию своих элементов и связей, то есть образованию и изменению структуры; 3) самоорганизующиеся программы являются открытыми системами; 4) самоорганизующиеся программы способны к отводу энтропии во внешнюю среду; 5) самоорганизующиеся программы в числе реализуемых функций выполняют работу, направленную на удаление от модельного аналога термодинамического равновесия и производство модельного аналога свободной энергии; 6) самоорганизующиеся программы функционируют до тех пор, пока обладают модельным аналогом свободной энергии.

Выделены подклассы самоорганизующихся программных систем: а) полностью самоорганизующиеся программные системы; б) программные системы, полученные путем самоорганизации.

Полностью самоорганизующаяся программная система образуется и существует благодаря процессам самоорганизации, проходящим в специальным образом устроенной среде; в такой системе процессы самоорганизации присутствуют на всех последующих этапах жизненного цикла программной системы. **Программная система, полученная путем самоорганизации** отличается тем, что по достижении ею какого-либо состояния, удовлетворительного с точки зрения проектировщиков или конечных пользователей, она извлекается из среды, в которой была получена, и в дальнейшем процессы самоорганизации в ней не происходят.

Согласно работе А. П. Руденко¹ прогрессивная эволюция, то есть появление объектов с принципиально новыми свойствами, возможна только в случае самоорганизации континуального типа. При этом образуется промежуточный объект в виде функционально неделимого кинетического континуума веществ и процессов. То есть программа, являющаяся самоорганизующейся в полном смысле этого слова, должна представлять собой функционально неделимую и динамически устойчивую совокупность данных, фрагментов программного кода и процессов их переноса, взаимодействия и взаимопревращения (потоков). Условно такую программу можно рассматривать как модельный аналог молекулярного организма. Подобного рода модели рассматриваются в теории искусственной жизни.

Далее в диссертации рассматриваются модели теории искусственной жизни, предлагается их классификация. Особое внимание уделяется моделям искусственной химии, и в особенности двум их подклассам: классическим моделям и моделям сообществ эволюционирующих ассемблерных программ. В случае последнего подкласса программа рассматривается как цифровой организм – модельный аналог молекулярного организма, обладающего запасом внутренней энергии и способного к саморепликации. Наиболее известными представителями данного подкласса являются модели «*Tierra*» и «*Avida*». С точки зрения моделирования биологических систем «*Tierra*» и «*Avida*» представляют собой модели «миров РНК» – сообществ самокопирующихся молекул РНК, существовавших на доклеточном биомолекулярном этапе развития живого вещества. Недостатками этих моделей, ограничивающими их применение для решения задач практического программирования, являются: медленный темп эволюции программ; проблема хрупкости (*brittleness*), или потери семантической значимости программного кода; проблема вырожденности кода.

В диссертации предлагается другой подход, при котором молекулярный цифровой организм представляет собой программу на языке высокого уровня. При этом, в отличие от цифровых организмов «*Tierra*» и «*Avida*», в модели генотип и фенотип цифрового организма будут разделены. Генотип будет представлен цифровой ДНК в виде последовательности целых неотрицательных чисел, а фенотип – совокупностью семантически законченных фрагментов программного кода и данных.

Предложенные принципы организации цифрового организма представлены функциональной моделью, которая рассматривает цифровой организм как множество минимально необходимых взаимодействий и использует в качестве метафоры сравнение с химическими реакциями. В это множество необходимых реакций входят: 1) базовая автокаталитическая реакция $A \xrightarrow{X, \varepsilon} M(M') + B$ (A – субстрат; B – продукты реакции; X – запись основного программного кода цифрового организма, выполняющая роль катализатора; ε – свободная энергия; M – матрица, или цифровая ДНК нового организма, возможно, мо-

¹ Руденко А. П. Самоорганизация и синергетика. – URL: <http://spkurdyumov.ru/what/samoorganizaciya-i-sinergetika-a-p-rudenko/> (дата обращения: 29.01.2017)

дифицированная мутацией (M'); 2) реакция расщепления с участием катализатора, освобождающая энергию $B_1 \xrightarrow{X, \varepsilon_1} \tilde{B}_1 + \varepsilon_2$ ($B_1 \in B$, \tilde{B}_1 – продукты расщепления, $\varepsilon_1 < \varepsilon_2$ – затрачиваемая и высвобождаемая энергия); 3) реакция матричного синтеза цифрового организма на основе цифровой ДНК $\{Dom_i\}_i + M \xrightarrow{P, \varepsilon} X$ ($\{Dom_i\}_i$ – множество фрагментов программного кода и данных, единообразно представленных как элементы некоторых доменов баз данных; P – процесс-построитель); 4) мутация цифровой ДНК $M \xrightarrow{X, \varepsilon} M'$.

Модельным аналогом свободной энергии является процессорное время. Описанные аналоги реакций позволяют не только описать процессы взаимодействий внутри цифровых организмов, но и задать взаимодействия как между отдельными организмами, так и видами.

Поскольку цифровой организм рассматривается в первую очередь как постоянная система отношений, а составляющие его элементы вводятся, выводятся или модифицируются со временем, цифровой организм можно рассматривать как автокатакинетическую¹ систему, занимающуюся самоперебором состояний и экспортирующую энтропию в окружающую среду. Отсюда можно предположить возрастание хаоса как результат постоянного распада и образования связей между цифровыми организмами или их составляющими. Под хаосом здесь понимается максимально неупорядоченное состояние, при котором отсутствует или не выявляется как пространственная, так и временная и функциональная упорядоченность, то есть отсутствуют или не выявляются какие-либо структуры. Цифровые организмы и структуры, возникающие в результате процессов самоорганизации в предложенной модели, будут замещать собой неупорядоченные области. С точки зрения компьютерной реализации это будет означать отвод энтропии во внешнюю среду реального мира и превращение информационной энтропии в термодинамическую (принцип Ландауэра).

Приведенные рассуждения позволяют перейти к формулировке новой парадигмы программирования, основанной на неравновесии. Неравновесное программирование – парадигма программирования, основанная на постоянстве протекания процессов образования-распада данных и программных единиц вследствие взаимодействий (аналогов реакций) друг с другом, а также неравновесии между этими процессами. Неравновесие понимается как совершение программами работы против аналогов химического и термодинамического равновесия (то есть против прекращения работы программной системы) с целью уменьшения неупорядоченности и, как следствие, образования структур, а также с целью увеличения доступного аналога свободной энергии (процессорного времени). В соответствии с парадигмой неравновесного программирования, процесс решения задачи формулируется в терминах

¹ Swenson R. Autocatakinetics, Yes – Autopoiesis, No: Steps Toward a Unified Theory of Evolutionary Ordering // International Journal of General Systems. – 1992. – Vol. 21. – P. 207–228.

множеств аналогов химических реакций, осуществляемых цифровыми организмами различных видов, и взаимоотношений между видами цифровых организмов, являющихся следствием взаимосвязей между множествами аналогов химических реакций.

Принципы парадигмы неравновесного программирования: 1) открытость программных структур любого уровня; 2) диссипативность; 3) компонентная организация программ; 4) иерархичность; 5) динамическое развертывание; 6) избыточность и конкуренция цифровых организмов (а возможно, и сообществ); 7) ограниченность времени существования цифровых организмов и их сообществ; 8) неравновесность; 9) антиэнтропийная направленность внутрисистемных процессов; 10) поддержание внутренней упорядоченности путем понижения потенциалов окружающей среды благодаря потреблению ресурсов и экспорту энтропии; 11) образование и тактированная пересборка цифровых организмов; 12) распределенность ресурсов (исходных данных) и вычислений, выполняемых цифровыми организмами; 13) безопасность.

Во второй главе предлагается и рассматривается структурная модель цифрового организма, созданная на основе функциональной (рисунок 1). Данная модель определяет также правила сборки цифрового организма на основе его цифровой ДНК.

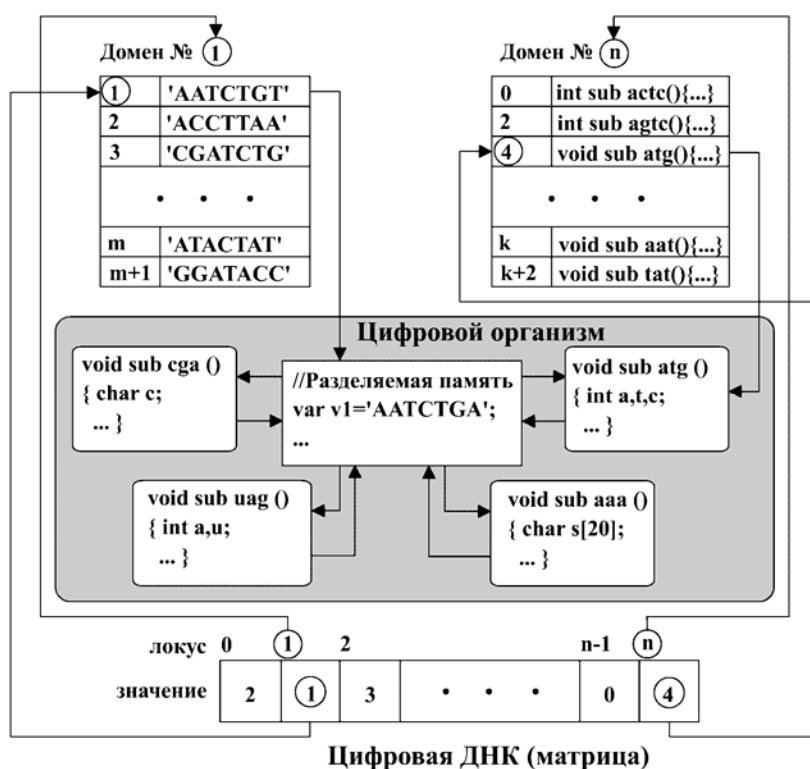


Рисунок 1 – Сборка цифрового организма на основе цифровой ДНК¹

¹ Кольчугина Е. А. Структура цифрового организма в самоорганизующихся программных системах // Программные продукты и системы. – Тверь : ЗАО НИИ ЦПС, 2012. – № 2. – С. 51–54.

На рисунке 1 показано, что цифровой организм представляет собой совокупность относительно независимых (и, возможно, параллельно исполняемых) фрагментов программного кода, модифицирующих переменные, которые описывают состояние цифрового организма. Эти переменные находятся в разделяемой памяти, доступной всем фрагментам программного кода внутри цифрового организма. И значения разделяемых переменных, и фрагменты программного кода являются результатом сборки цифрового организма на основе кодирующей последовательности (цифровой ДНК) и множества доменов, содержащих фрагменты программного кода и значения переменных.

Первичная структура цифровой ДНК задается последовательностью целых чисел, кодирующих алгоритм поведения и свойства цифрового организма. В диссертации вводятся операции над этой структурой, некоторые из них сходны с операциями генетических алгоритмов: 1) спонтанная мутация; 2) копирование значений выделенных локусов из одной цифровой ДНК в другую; 3) кроссинговер; 4) удаление (выключение) локуса; 5) обмен значениями локусов в пределах одного набора генов.

Далее выделяются и исследуются аномалии выполнения введенных операций. Аномалии связаны в первую очередь с тем, что локусам цифровой ДНК ставятся в соответствие значения, которые не определены для данного локуса, и это может препятствовать сборке цифрового организма на основе его ДНК. Предлагаются возможные способы устранения таких аномалий: 1) выключение локуса с некорректным значением; 2) дополнение домена, соответствующего локусу, новым значением.

Если выполнять фрагменты программного кода, входящие в состав фенотипа цифрового организма, в порядке возрастания номеров соответствующих им локусов, то первичную структуру цифровой ДНК можно рассматривать как закодированное представление линейного алгоритма поведения цифрового организма. На основе первичной линейной структуры цифровой ДНК путем ее сгибания (фолдинга) может быть получена матрица расписания, задающая структуру второго порядка цифровой ДНК. Элементами этой матрицы являются номера локусов цифровой ДНК, соответствующих фрагментам программного кода. Матрица описывает множество параллельных алгоритмов. Строки матрицы могут рассматриваться как ярусы параллельного алгоритма, а столбцы – как параллельные ветви. Конкретный экземпляр алгоритма задается рекурсивным разбиением исходной матрицы на подматрицы меньших порядков. Разбиение может быть задано также с помощью формулы либо древовидной структуры (рисунок 2). Приведенные на рисунке примитивы *par* означают параллельное выполнение заданных подмножеств операций алгоритма, примитивы *seq* – последовательное выполнение, а *exec* – выполнение фрагментов кода, соответствующих локусам первичной структуры цифровой ДНК.

Матрица и ее разбиение в совокупности кодируют параллельный алгоритм поведения цифрового организма. Они должны быть представлена также

и на уровне первичной структуры цифровой ДНК. Для получения новых параллельных алгоритмов на основе первичной структуры цифровой ДНК либо уже существующего параллельного алгоритма введены следующие операции: 1) операции над матрицей расписания: а) горизонтальный фолдинг; б) вертикальный фолдинг; в) добавление строки; г) добавление столбца; д) удаление строки; е) удаление столбца; ж) перестановка двух элементов матрицы; з) копирование элемента матрицы в произвольную позицию); 2) операции преобразования разбиения матрицы, или способа прочтения матрицы расписания (операции генетического программирования, за исключением делеции и дупликации, основанные на двух более простых операциях: а) построение древовидной структуры случайным образом на основе исходной заданной матрицы или подмножества ее элементов; б) замена всей древовидной структуры или ее части (поддерева) другой древовидной структурой).

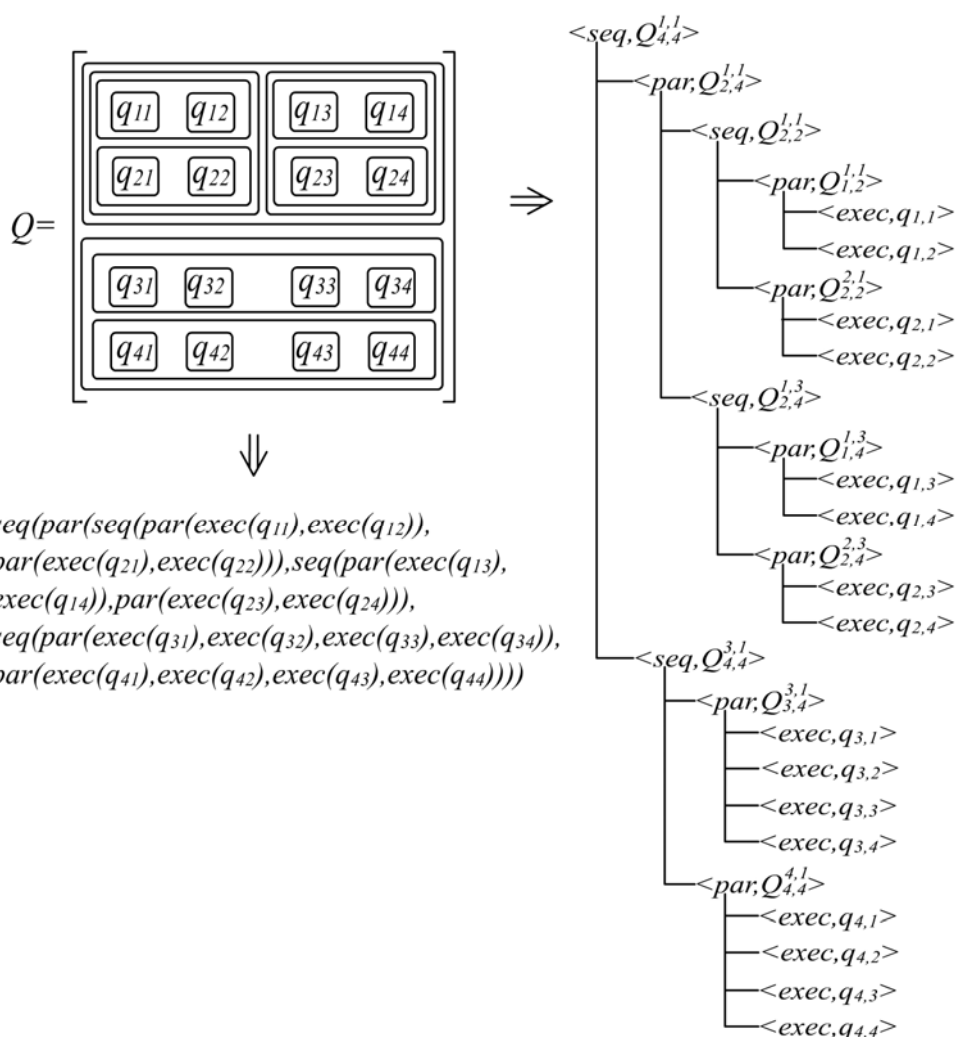


Рисунок 2 – Построение древовидной структуры на основе матрицы Q^1

¹ Кольчугина Е. А. Построение параллельных алгоритмов поведения программных агентов методами генетического программирования // Инфокоммуникационные технологии. – 2012. – Т. 10, № 3. – С. 40–44.

Введенные структуры ДНК и операции над ними позволяют обеспечить эволюцию алгоритма поведения цифрового организма не только на уровне реализации отдельных элементов алгоритма, но и на уровне параллелизма или изменения порядка выполнения элементов алгоритма. Можно выделить несколько типов эволюции, происходящих на уровне различных структур как в составе цифровой ДНК, так и вне ее: 1) эволюция на уровне первичной структуры цифровой ДНК; 2) эволюция доменов и их подмножеств (субдоменов); 3) эволюция матрицы расписания; 4) эволюция способов прочтения (или разбиений) матрицы расписания.

В третьей главе исследуется динамика поведения цифрового организма, представленная последовательностью смены его состояний. При этом исследуются свойства упорядоченности, устойчивости и сходимости последовательности состояний цифрового организма при сохранении внешних условий. Свойства упорядоченности, устойчивости и сходимости рассматриваются с двух точек зрения: 1) по отношению ко всему генотипу цифрового организма; 2) по отношению к циклам состояний, задаваемых глобальными параметрами.

В первом случае речь идет о кластеризации по генетическому признаку и об образовании квазивидов. Генотип каждого цифрового организма задается строкой $z_i(t) = \langle z_1^i(t), z_2^i(t), \dots, z_n^i(t) \rangle$, где $z_j^i(t)$ – элемент строки, описывающей состояние цифрового организма в момент времени t . Тогда процесс конвергенции, когда признаки организмов одного квазивида, живущих в одинаковых условиях, начинают сходиться, описывается соотношением

$$Z^m(t) = \frac{1}{m} \sum_{i=1}^m \frac{1}{n} \sum_{k=1}^n |z_k^i(t) - z_k^j| \rightarrow 0 \text{ при } t \rightarrow \infty, \text{ где } m - \text{ количество цифровых}$$

организмов, $z_j = \langle z_1^j, z_2^j, \dots, z_n^j \rangle$ – аттрактор в виде особой точки (усредненный генотип). То же самое будет верно и в отношении цифровых организмов разных видов, занимающих одну и ту же экологическую нишу и существующих в идентичных условиях. В этом случае для организмов двух разных видов с учетом необходимых видовых различий будет верно,

$$\text{что } Z^{m_1 m_2}(t) = \frac{1}{m_1} \sum_{j=1}^{m_1} \left(\frac{1}{m_2} \sum_{i=1}^{m_2} \left(\frac{1}{n} \sum_{k=1}^n |z_k^i(t) - z(t)_k^j| \right) \right) \rightarrow \alpha \text{ при } t \rightarrow \infty. \text{ Возможно, что}$$

на основе одного исходного вида образуется несколько видов в результате мутаций и дивергенции признаков. Это означает, что в исследуемом фазовом пространстве присутствовало несколько аттракторов, а исходный генотип принадлежал сепаратрису или был источником (неустойчивым узлом).

Во втором случае при кластеризации генотипы цифровых организмов могут различаться, но одинаковыми являются длины циклов состояний организмов. Длина цикла состояний позволяет судить об упорядоченности поведения организма, устойчивости по отношению к случайным мутациям, способности к адаптации и самоорганизации. Для исследования циклов сос-

тояний была предложена NK -подобная автоматная модель цифрового организма (рисунок 3). Классические NK -автоматы, введенные С. А. Кауффманом¹, состоят из N элементов, каждый из которых имеет в среднем $K \leq N$ входов, один выход и соединен с K другими попарно различными элементами. Наиболее часто рассматриваются булевские автоматы. При $K = 2$ автоматы функционируют «на границе порядка и хаоса», подобно живым системам, и наблюдается конвергенция: у автоматов с разными начальными состояниями последующие состояния постепенно сходятся, что позволяет говорить не только об упорядоченности, но и об устойчивости автомата к случайным мутациям. Аттрактором для последовательности смены состояний NK -автомата является предельный цикл, при $K = 2$ количество циклов определяется как \sqrt{N} , а медиана длины цикла также пропорциональна значению \sqrt{N} .

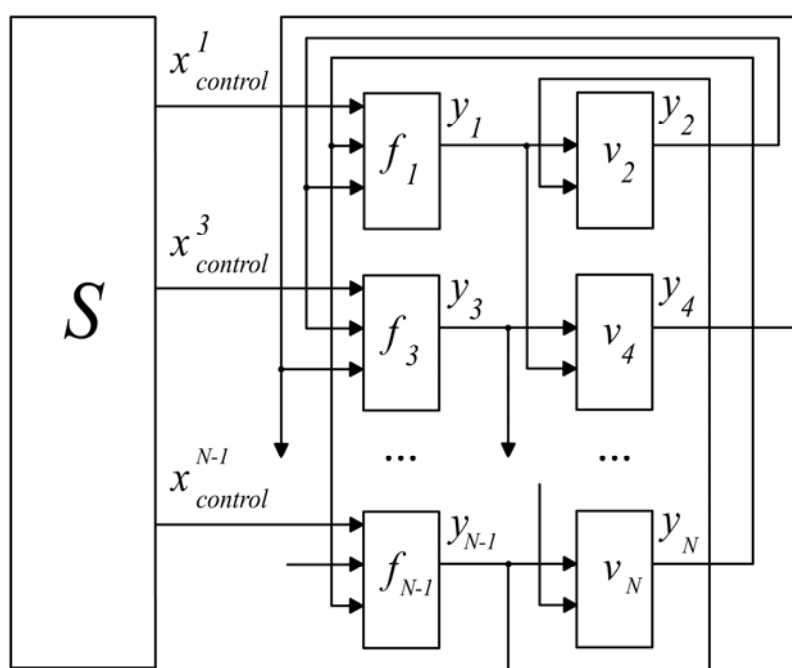


Рисунок 3 – NK -подобный автомат²

NK -подобный автомат отличается от классического NK -автомата тем, что: 1) все элементы NK -подобного автомата делятся на два непересекающихся подмножества: $F = \{f_{j_1}\}_{j_1=1}^{N_1}$ и $V = \{v_{j_2}\}_{j_2=1}^{N_2}$, $N_1 + N_2 = N$, все входы любого элемента из V соединены только с выходами элементов из F , обратное не верно; 2) вводится управляющий элемент S , задающий программу вычисле-

¹ Kauffman S. A. The Origin of Order: Self-Organization and Selection in Evolution. – New York-Oxford : Oxford University Press, 1993. – 710 p.

² Кольчугина Е. А. Исследование свойств цифровых организмов с помощью NK -подобных автоматов // Вестник компьютерных и информационных технологий. – 2011. – № 11. – С. 20–24.

ний автомата, все выходы которого являются входами $x_{control}^j$ для функциональных элементов из F . Среднее число входов автомата обозначим K' .

Для элементов f_j , моделирующих фрагменты программного кода, в каждый момент времени $t \in T$ выход определяется на основании множества входов $x_i^j(t)$ следующим образом:

$$y_j(t) = \begin{cases} y_j(t-1), & x_{control}^j(t) = 0, \\ f_j(x_1^j(t), \dots, x_{k+1}^j(t)), & x_{control}^j(t) = 1. \end{cases}$$

Для элементов v_j , моделирующих переменные состояния цифрового организма, в каждый момент времени $t \in T$ выход определяется на основании множества входов $x_i^j(t)$ следующим образом:

$$y_j(t) = \begin{cases} y_j(t-1), & \forall i \in \overline{1, K'}: x_i^j(t) = x_i^j(t-1), \\ x_i^j(t), & \exists i \in \overline{1, K'}: x_i^j(t) \neq x_i^j(t-1). \end{cases}$$

Чтобы реализовать элемент v_j , потребуется снабдить его памятью для хранения K' переменных, в которые будут записываться значения входов $x_i^j(t-1)$ в предыдущий момент времени. Для реализации такого свойства среди элементов f_j выделяется подмножество элементов-«замедлителей», которые пропускают на выход единственные входные значения, никак их не модифицируя, и тем самым отсрочивают на один такт потерю текущих значений выходов v_j .

Результатом предварительных расчетов является утверждение, что поведение NK -подобного автомата при $K' = 2$ будет аналогично поведению классического NK -автомата при $2 < K < 3$, $K \approx 2$ и близко к поведению «на границе порядка и хаоса». Это утверждение было подтверждено результатами экспериментов.

Также экспериментально было установлено, что высота параллельного алгоритма, то есть свойство программы, задаваемой элементом S , влияет на длину предельного цикла. Таким образом, для выживания цифрового организма важны не только значения переменных состояния и реализации действий, входящих в его алгоритм поведения, но также порядок и синхронность выполнения действий, модифицирующих значения переменных состояния.

В главе также рассматриваются вопросы применения $NKCS$ - и $NKCSW$ -моделей для исследования динамики эволюции сообществ.

В четвертой главе рассматриваются вопросы применения методов теории нумераций в проектировании самоорганизующихся программных систем. Предложенный в главе 2 способ кодирования цифрового организма с помощью последовательности целых чисел делает возможным применение к подобным последовательностям методов теории нумераций, в частности, операций свертки для кодирования последовательности одним числом, и раз-

вертки, или восстановления, последовательности чисел из свернутого представления.

Операция свертки $s^n(x_1, \dots, x_n) = s(s^{n-1}(x_1, \dots, x_{n-1}), x_n)$ и обратное ей восстановление последовательности при помощи n -развертки $s^n(r_{n,1}(x), \dots, r_{n,n}(x)) = x$, $r_{n,i}(s^n(x_1, \dots, x_n)) = x_i$ были предложены Ю. Л. Ершовым¹. В дальнейшем считается, что функция свертки строится на основе функции нумерации пар $f(x, y) = s$ и обладает следующими основными свойствами:

- 1) $\forall x, y \in \mathbf{N}_{0+} : (x \neq y) \leftrightarrow f(x, y) \neq f(y, x)$,
- 2) $\forall x_1, x_2, y_1, y_2 \in \mathbf{N}_{0+} : ((x_1 \neq y_1) \vee (x_2 \neq y_2)) \leftrightarrow f(x_1, x_2) \neq f(y_1, y_2)$,
- 3) $\forall x_1, x_2, x_3 \in \mathbf{N}_{0+} : (x_1 \neq x_2 \neq x_3) \rightarrow f(x_1, f(x_2, x_3)) \neq f(f(x_1, x_2), x_3)$,
- 4) $\forall z \in \mathbf{N}_{0+} \exists ! (x, y), x, y \in \mathbf{N}_{0+} : f(x, y) = z$.

В дополнение к классической функции свертки $s^n(x_1, \dots, x_n)$ в работе вводится функция свертки $sq^n(x_1, \dots, x_n) = f(s^n(x_1, \dots, x_n), n)$, учитывающая длину последовательности, что позволяет однозначно идентифицировать последовательности по результатам их свертки. С целью снижения роста функции свертки вводится также функция свертки $sv^{n+1}(x_1, \dots, x_n, n) = f(sv^n(x_1, \dots, x_n), n)$, учитывающая длину последовательности и вычисляемая по схеме сдваивания. На основе этих введенных функций и обратных к ним функций восстановления, а также классических функций свертки и восстановления вводятся обобщенные функции свертки и восстановления.

Введенные функции позволяют решать следующие задачи: 1) кодировать числовые последовательности при помощи целых чисел и использовать результаты кодирования для уникальной идентификации последовательностей; 2) ввести функции замены результатов свертки бинарными последовательностями произвольной длины, что позволяет кодировать исходные последовательности бинарными последовательностями; 3) предложить алгоритмы кодирования древовидных структур, представляющих параллельные алгоритмы поведения цифровых организмов, при помощи результатов вычисления функций сверток, а также предложить алгоритмы построения упомянутых древовидных структур путем применения операций восстановления последовательностей; 4) предложить алгоритм представления древовидных структур, задающих параллельные алгоритмы поведения цифровых организмов, как коллегиально-иерархических структур, рассматривая функцию свертки как П-функцию² подчинения вершин древовидной структуры.

¹ Ершов Ю. Л. Теория нумераций. – М. : Наука, Гл. ред. физ.-мат. лит., 1977. – 416 с.

² Айзерман М. А., Гусев Л. А., Петров С. В., Смирнова И. М., Тенебаум Л. А. Динамический подход к анализу структур, описываемых графами (основы графовой динамики).

Функции свертки в сочетании с доменно-ориентированным¹ подходом к организации баз данных позволяют выполнить свертку не только отдельных последовательностей, но также доменов и таблиц отношений баз данных. Это позволяет упростить: 1) идентификацию отношений; 2) сравнение отношений; 3) исследование эволюции отношений.

Ориентация на использование методов теории нумерации в сочетании с доменно-ориентированным подходом и покортежным распределением данных по вычислительным узлам позволила автору предложить новую концепцию распределенного информационного поиска на основе распределенных баз данных и сообщества цифровых организмов. В основе концепции лежит понятие расширенного запроса (рисунок 4).

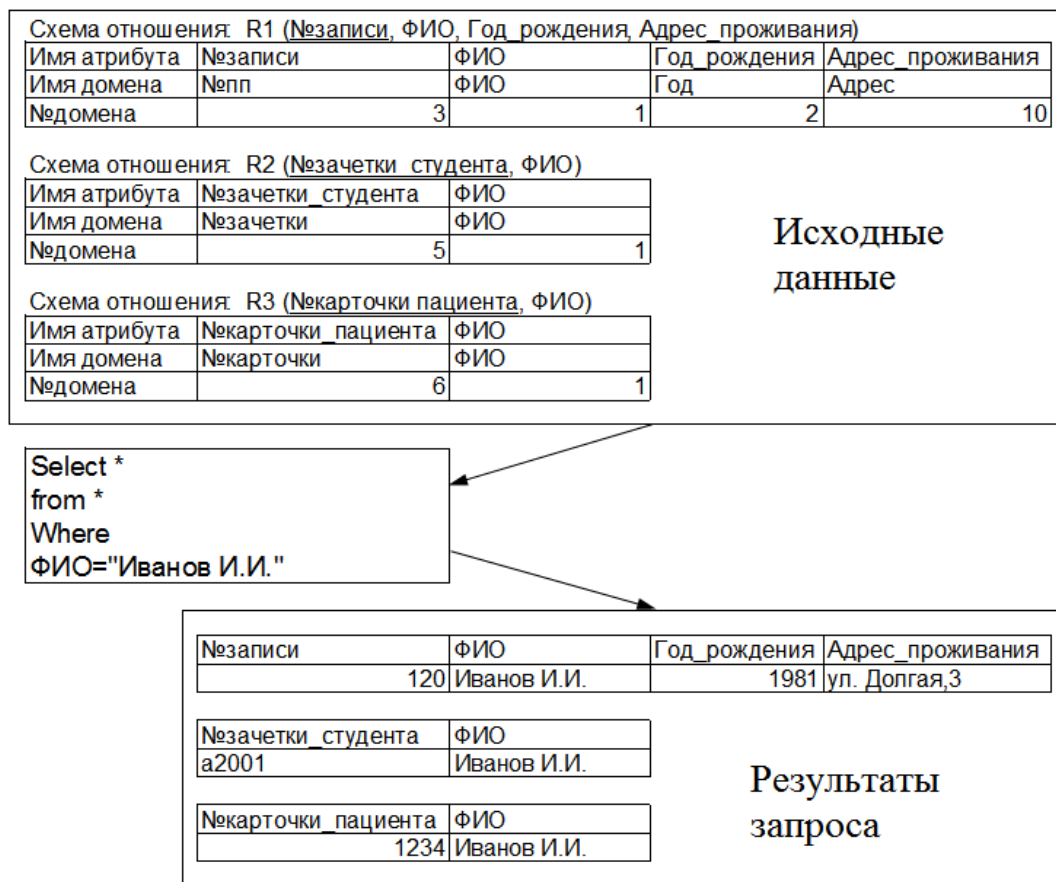


Рисунок 4 – Пример выполнения расширенного запроса²

ки) // Исследования по теории структур : сб. науч. тр. / АН СССР, Ин-т пробл. Управления ; отв. ред. М. А. Айзерман, Э. Р. Каянелло. – М. : Наука, 1988. – С. 5–75.

¹ Линьков В. М., Иванцов М. А. Разработка системы управления доменом // Сборник материалов научно-практического семинара «Применение баз данных» (Пенза, 26–27 ноября 1997). – Пенза : ПДЗ, 1997. – С. 6–7.

² Кольчугина Е. А. Организация баз данных и распределенный информационный поиск // Современные информационные технологии : тр. Междунар. науч.-техн. конф. (Computer-based conference). – Пенза : Пенз. гос. технол. академия, 2005. – Вып. 2. – С. 95–97.

В расширенном запросе не указываются имена таблиц, из которых необходимо выполнить выборку. Исходными данными служат имена атрибутов отношений и их значения. По имени атрибута определяется номер домена, с которым связан атрибут, а также номер значения атрибута в домене. Группа мобильных цифровых организмов просматривает все имеющиеся отношения, табличные представления которых хранятся на всех доступных вычислительных узлах, и одновременно выполняет выборку из тех отношений, схемы которых содержат атрибуты, связанные с теми же доменами, что и в запросе. Выборке подвергаются те кортежи, внутренние представления которых содержат те же номера значений из заданных доменов, что и в запросе. Затем, после того как выбраны все внутренние представления подходящих кортежей, происходит формирование внешнего представления кортежей на основе доменов значений. В отличие от классических программных агентов, цифровые организмы при выполнении информационного поиска не нуждаются в указании адреса ресурса. Расширенный запрос сближает поиск информации в базе данных с релевантным поиском в Интернете: в результате запроса на выборку можно получить все кортежи всех отношений, имеющих заданные значения атрибутов.

Применение операций свертки и восстановления позволяет решить задачу сохранения цифровой ДНК как числовой последовательности нерегулярной длины в таблицах баз данных реляционного типа. Тем самым упрощается решение задач сравнения и идентификации последовательностей, представленных в базах данных, отображения эволюции цифровых организмов и их сообществ в базах данных, а также исследования эволюции доменов. Для исследования эволюции цифровых организмов и доменов следует исходить из концепции темпоральных, или временных, баз данных. Благодаря применению операций свертки и восстановления, эволюция цифрового организма или домена может быть представлена последовательностью сверток, имеющих временную метку. В качестве временной метки используется свертка временной координаты или временного интервала.

Также в главе 4 предложена новая концепция программно-информационных систем, состоящих из баз данных и сообществ цифровых организмов, которые взаимно эволюционируют, самоорганизуясь и модифицируя друг друга.

В пятой главе рассматриваются вопросы организации замкнутых искусственных сред существования цифровых организмов – искусственных миров, а также рассматриваются вопросы организации сообществ цифровых организмов. Искусственный мир задается совокупностью вида $W = \langle S, T, R, \tilde{A}, \tilde{D} \rangle$, где S – пространство; T – время; R – правила и ограничения (законы), которым подчиняется существование искусственного мира; \tilde{A} – множество цифровых организмов («живая природа»); \tilde{D} – множество пассивных данных («неживая природа»).

Пространство искусственного мира предлагается рассматривать как дискретное клеточное, имеющее произвольную размерность. Особенностью модели является представление каждой клетки как совокупности бесконечного числа логических слоев. Каждому виду цифровых организмов соответствует единственный слой, но каждый слой может вмещать произвольно большое количество видов и цифровых организмов. Это соответствие позволяет упорядочить взаимоотношения между цифровыми организмами: каждый цифровой организм перемещается только вдоль соответствующего ему слоя; каждый вид может служить субстратом только для видов из вышележащих слоев и использовать виды только из нижележащих слоев в качестве субстрата. Далее в работе рассматриваются и предлагаются алгоритмы перемещения цифровых организмов в клеточном пространстве.

Время в моделях искусственных миров в основном рассматривается как линейное дискретное, неограниченное в отношении будущего, но возможно использование и других моделей, которые рассматриваются в работе в связи с возможностью применения темпоральных баз данных.

Продолжается начатая в главе 4 тема организации пространственно распределенных растущих баз данных с непрерывно увеличивающимся объемом хранения. В совокупности с сообществами цифровых организмов такие базы данных образуют самоорганизующиеся программно-информационные системы. Однако и сами базы данных могут быть способны хотя бы к самоменеджменту. Предлагается алгоритм пополнения пространственно распределенных растущих баз данных на основе доменно-ориентированной модели данных с покортежным распределением отношений, сопровождающегося активацией локальных СУБД в новых вычислительных узлах. Для исследования поведения подобной распределенной растущей базы данных предлагается модифицированная модель «кучи песка» (классическая модель «кучи песка» предложена П. Баком¹). Результаты моделирования показали отсутствие самоорганизованной критичности, но это и не было целью построения модели. Важно, что в процессе моделирования наблюдалось преобладание коротких событий, то есть данные окончательно размещались, как правило, вблизи от того вычислительного узла, в который изначально поступили. Это означает, что добавление данных даже в заполненную базу будет происходить достаточно быстро, с небольшим количеством пересылок данных между вычислительными узлами.

Далее вводится понятие вида цифрового организма, определяются основные отношения между видами, рассматриваются возможные варианты структуры многовидовых сообществ цифровых организмов. Вид цифрового организма определяется на основе уравнения $A \xrightarrow{X, \varepsilon} M(M') + B$, задающего

¹ Бак П. Как работает природа: Теория самоорганизованной критичности : пер. с англ. / вступ. ст. Г. Г. Малинецкого. – М. : УРСС : ЛИБРОКОМ, 2013. – 276 с.

основную полезную функцию, реализуемую организмом. Это же уравнение является основой для определения межвидовых отношений. В диссертационной работе основными видами межвидовых отношений считаются отношения использования (с сохранением ресурса) и истребления (с необратимыми затратами ресурса). На базе этих отношений могут быть промоделированы все межвидовые отношения, рассматриваемые в экологии, кроме аменсализма: нейтрализм; «хищник-жертва» или «паразит-хозяин»; комменсализм; симбиоз; конкуренция.

Представление структуры многовидового сообщества цифровых организмов в виде графа позволяет выделить два основных типа структур: ациклические иерархические сообщества без обратных связей; сообщества с циклической структурой (гиперциклы).

В результате рассмотрения достоинств и недостатков этих структур в качестве возможного варианта структуры самоорганизующихся программных систем прикладного назначения в виде многовидовых сообществ цифровых организмов предлагается ациклическая иерархическая структура сообщества, в которой виды с вышележащих уровней не истребляют полностью виды с нижележащих уровней, а лишь угнетают их.

На основе принципов эволюционного эмбриогенеза в главе предлагается рассматривать самоорганизующиеся программные системы прикладного назначения как результаты развития некоторой зародышевой структуры, возможно, единственной цифровой ДНК, помещенной в условия искусственного мира.

В шестой главе предложена структура системного программного обеспечения, необходимого для реализации парадигмы неравновесного программирования. Выделяются и рассматриваются особенности такого программного обеспечения. Описываются эксперименты с самоорганизующимися программными системами. Приводятся результаты подобных экспериментов. Приводится и рассматривается пример прикладной программной системы, построенной в соответствии с парадигмой неравновесного программирования

В первой части главы устанавливается, что программное обеспечение, необходимое для реализации парадигмы неравновесного программирования, должно строиться по клиент-серверной модели. В задачи серверной стороны входит организация искусственного мира, в котором существуют цифровые организмы (в частности: 1) задание пространственно-временных свойств искусственного мира; 2) сборка (реактуализация) цифровых организмов и запуск их на выполнение; 3) фиксация цифровых ДНК и информации о событиях в журналах событий и базах данных; 4) анализ цифровых ДНК на соответствие множеству заданных правил с целью предотвращения выполнения заведомо нежизнеспособных или потенциально опасных организмов; 5) ведение баз данных программных компонентов и правил. Клиентская часть представлена цифровыми организмами, которые можно рассматривать как простейшие неинтеллектуальные мобильные агенты.

Отдельно рассматриваются следующие вопросы, связанные с программной реализацией парадигмы неравновесного программирования: 1) обеспечение безопасности неравновесной программной системы; 2) принципы уникальной и информативной идентификации цифровых организмов; 3) средства обеспечения коммуникации между цифровыми организмами; 4) программное обеспечение визуализации результатов моделирования.

Кроме того, в первой части главы анализируются преимущества и пути развития предлагаемой в работе парадигмы неравновесного программирования, а также выделяются области возможного применения парадигмы неравновесного программирования.

Вторая часть главы посвящена описанию экспериментов с неравновесными самоорганизующимися программными системами и анализу их результатов. Все эксперименты проводились на локальной вычислительной сети, состоящей из восьми узлов, на базе которой моделировался искусственный мир с тороидальной топологией и дискретным временем. Проведенные эксперименты позволили установить следующее:

– в программных системах, построенных на основе парадигмы неравновесного программирования, присутствуют процессы самоорганизации: в ответ на неспецифические воздействия извне, выражающиеся в нарушении равномерности пространственного распределения источников исходных данных, нарушается симметрия поведения сообщества цифровых организмов – в сообществе начинают преобладать организмы с алгоритмом перемещения, гарантирующим наилучшие результаты поиска данных (на рисунке 5 показано изменение преобладания генотипов во времени в условиях неравномерного пространственного распределения источников данных); в условиях же повсеместного распределения источников данных все алгоритмы перемещения встречались с одинаковой частотой (на рисунке 6 показано изменение преобладания генотипов во времени в условиях повсеместного равномерного пространственного распределения источников данных) (результаты, представленные на рисунках 5 и 6, получены в ходе эксперимента, состоявшего из двух частей с различным пространственным распределением источников данных в каждой части эксперимента¹);

– на основе одного или двух исходных организмов одного вида может быть сгенерировано многовидовое сообщество, в котором происходят процессы внутривидовой и межвидовой конкуренции, наблюдаются отношения «хищник-жертва» и другие межвидовые отношения;

– виды и отдельные организмы, имеющие хотя бы малые конкурентные преимущества, в отсутствие регулирования оказываются монополистами и захватывают все доступные им ресурсы, тем самым истребляя прочие виды и организмы;

¹Кольчугина Е. А. Результаты эксперимента по созданию эволюционирующего программного обеспечения // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2007. – № 1. – С. 54–60.

– сосуществование видов хищника и жертвы при отсутствии убежищ и в условиях постоянного контакта возможно только в том случае, если хищник оказывает «мягкое» давление на жертву, то есть не истребляет вид полностью и не лишает его возможности размножаться.

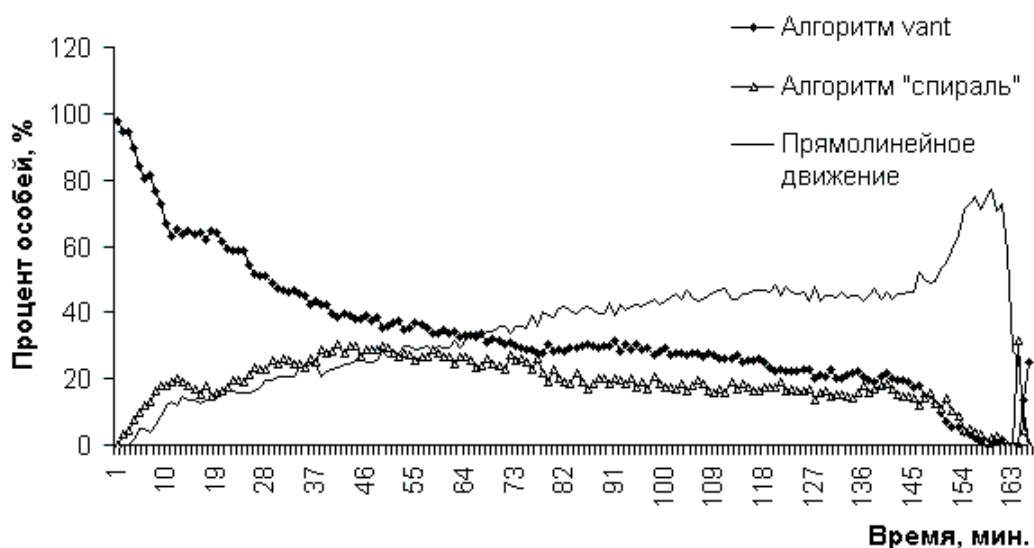


Рисунок 5 – Преобладающие генотипы (первая часть эксперимента)¹

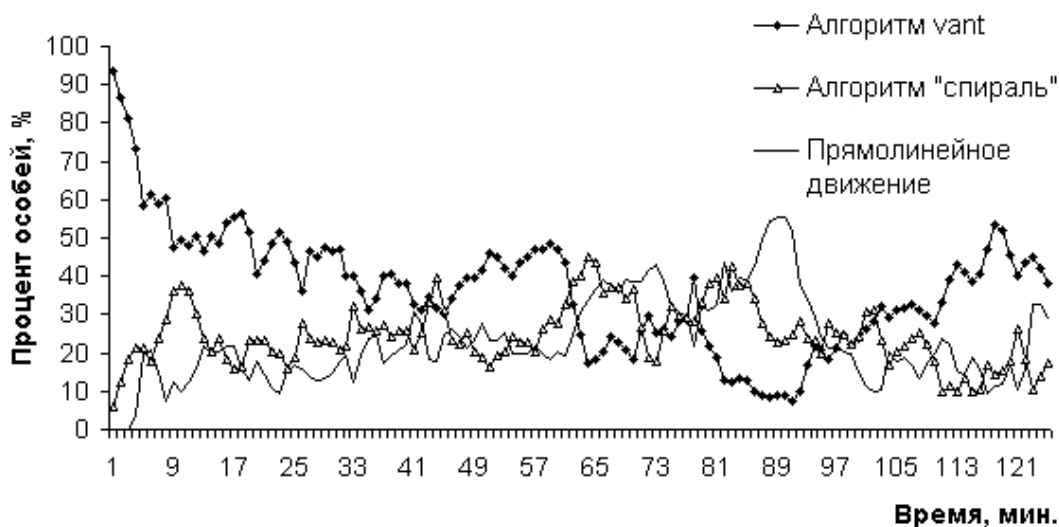


Рисунок 6 – Преобладающие генотипы (вторая часть эксперимента)²

¹ Рисунок иллюстрирует преобладание генотипов цифровых организмов с алгоритмом перемещения, гарантирующим наилучшие результаты поиска данных, в условиях неравномерного пространственного распределения источников данных (Кольчугина Е. А. Результаты эксперимента по созданию эволюционирующего программного обеспечения // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2007. – № 1. – С. 54–60.

² Рисунок иллюстрирует отсутствие преобладания какого-либо алгоритма перемещения в условиях повсеместного равномерного пространственного распределения источников данных. (Кольчугина Е. А. Результаты эксперимента по созданию эволюционирующего программного обеспечения // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2007. – № 1. – С. 54–60.

В третьей части главы приведен пример реализации прикладной программной системы, построенной в соответствии с парадигмой неравновесного программирования. С использованием методики *PERT* проведена оценка снижения трудоемкости разработки прикладной программной системы по сравнению с применением агент-ориентированного подхода. Для реализованной программной системы снижение трудоемкости составляет: для одного элементарного пакета работ в $\frac{E_{2i}}{E_{1i}} \approx 30,6$ раза; для всего проекта

в $\frac{E_2^{95\%}}{E_1^{95\%}} \approx 79,73$ раза. Кроме того, применение парадигмы неравновесного про-

граммирования приводит к снижению объема программного кода (в 17 раз), статистически возможного количества ошибок, затрат времени на передачу программ по сети, снижению отрицательного влияния человеческого фактора благодаря автоматизации программирования, а также снижению трудоемкости сопровождения.

В заключении сформулированы основные научные результаты, полученные в диссертационной работе.

В приложениях приводятся: математические обозначения, используемые в работе; акты внедрения и реализации результатов исследования.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ И ВЫВОДЫ

1. Предложена парадигма неравновесного программирования, в основу которой положены принципы молекулярной биологии. Предложенная парадигма является примером бионического подхода в программировании. Парадигма неравновесного программирования отличается тем, что программа рассматривается как имеющая изменяющуюся в ходе функционирования структуру совокупность процессов, связанных по исходным данным и результатам вычислений. Целью и основным назначением парадигмы является создание концептуальной основы разработки самоорганизующихся программ с самоорганизацией континуального типа. В основе парадигмы лежит понятие независимой самоорганизующейся программной единицы, или цифрового организма.

2. Предложена функциональная модель цифрового организма как аналога молекулярного организма, позволяющая выделить основные процессы, необходимые для существования цифрового организма, и выявить взаимосвязи между процессами. Модель отличается тем, что в ней в качестве основного строительного элемента рассматривается функция, что позволяет избежать появления проблемы хрупкости программного кода в процессе эволюции программ.

3. Предложена структурная модель цифрового организма, которая позволяет предложить технологию программной реализации цифрового организма. Модель отличается тем, что структурно цифровой организм состоит из раздельно представленных генотипа, заданного цифровой ДНК, и фенотипа,

заданного в виде совокупности, состоящей: а) из семантически целостных и законченных фрагментов программного кода; б) данных, представленных глобальными параметрами состояния цифрового организма. Обобщенно и данные, и фрагменты программного кода рассматриваются как функции, что позволяет описать программу в терминах унифицированных структурных элементов и предложить технологию программной реализации цифрового организма как самоорганизующейся программной единицы.

4. Предложен метод кодирования алгоритма поведения и свойств цифрового организма в составе структуры цифровой ДНК, заданной как линейная последовательность целых чисел. Метод отличается тем, что с его помощью унифицировано и единообразно, посредством целочисленной нумерации представляются как фрагменты программного кода, входящие в состав реализации алгоритма поведения, так и данные, представляющие свойства и текущее состояние цифрового организма. Это позволяет унифицировать выполнение генетических операций над всеми элементами цифровой ДНК, безотносительно к тому, представляют ли они данные или фрагменты программного кода. В силу этого становится возможным автоматизировать разработку программ и обеспечить эволюцию программного обеспечения в процессе его эксплуатации, в результате чего снижается трудоемкость разработки и повышается ее качество благодаря снижению влияния человеческого фактора.

5. Предложен метод кодирования управляющих структур в составе цифровой ДНК, задающих параллельные версии алгоритма поведения цифрового организма. В основе метода кодирования лежит представление параллельного алгоритма в виде сочетания матрицы элементов цифровой ДНК (матрицы расписания) и формулы (правила) прочтения разбиения матрицы. Матрица задает множество возможных вариантов параллельных алгоритмов поведения цифрового организма, а формула определяет конкретный экземпляр параллельного алгоритма. Данный метод кодирования позволяет порождать множество различных параллельных алгоритмов при помощи генетических операций над цифровой ДНК и преобразований матрицы расписания, а также преобразований, порождающих новые правила прочтения матрицы. В результате автоматизируется процесс разработки параллельных программ, снижается его трудоемкость и повышается качество разработки программ благодаря снижению влияния человеческого фактора.

6. Предложена НК-подобная автоматная модель цифрового организма, отличающаяся от классической НК-модели С. А. Кауффмана тем, что: а) элементы автомата разбиты на два непересекающихся подмножества; б) имеется управляющий элемент, задающий программу поведения автомата, которая может быть описана ярусно-параллельной формой алгоритма с высотой более одного яруса. Модель позволяет исследовать динамические свойства цифрового организма с точки зрения устойчивости поведения, оцениваемой на основе длины периода последовательности смены состояний автомата.

7. Предложены методы кодирования цифровой ДНК путем применения функций свертки теории нумераций по отношению к исходной числовой по-

следовательности, задающей цифровую ДНК. Методы отличаются тем, что в результате применения функций свертки заменяют кодируемую числовую последовательность одним целым числом. Это позволяет компактно сохранять цифровые ДНК в базах данных, а затем исследовать эволюцию программной системы, а также выполнять операции над цифровыми организмами путем преобразования сверток их ДНК-структур. Методы преобразования сверток ДНК-структур позволяют автоматизировать генерацию новых программ на основе кодированных представлений уже имеющихся.

8. Предложен новый метод организации распределенного информационного поиска в растущих доменно-ориентированных базах данных с покортежным пространственным распределением отношений, реализуемый самоорганизующимся сообществом цифровых организмов. Метод отличается тем, что при поиске согласно этому методу отыскиваются не данные, а их уникальные внутрисистемные идентификаторы, нумерующие значения в некоторых доменах. Метод позволяет выполнять информационный поиск по таблицам отношений произвольной структуры и вне зависимости от их пространственного расположения и делает поиск в базах данных сходным с релевантным поиском информации в Интернете, что расширяет возможности информационного поиска в распределенных базах.

9. Предложены принципы организации замкнутых сред, или искусственных миров, в которых существуют и развиваются сообщества самоорганизующихся цифровых организмов. Отличительной особенностью предлагаемой модели искусственных миров является послойная организация пространства модели, которое в общем случае является клеточным. Послойная организация позволяет упорядочить взаимоотношения различных видов организмов, назначая каждому виду слой с уникальным номером. Таким образом, взаимоотношения между видами могут быть выражены через отношения между номерами слоев. Это обеспечивает стратификацию видов по уровням, прозрачность структуры и управляемость сообщества цифровых организмов.

10. Впервые разработана структура системного программного обеспечения, необходимого для реализации предложенной парадигмы неравновесного программирования и создания самоорганизующихся программ. Программное обеспечение имеет клиент-серверную архитектуру и позволяет выполнять контроль свойств цифровых организмов, их периодическую переборку (реактуализацию) и запуск на выполнение, то есть динамически генерировать самоорганизующиеся программные системы и полностью контролировать их функционирование и развитие.

11. На основе проведенных экспериментальных исследований даны рекомендации по проектированию самоорганизующихся программных систем с требуемыми свойствами. Проведена оценка снижения объема программного кода, трудозатрат на разработку и сопровождение программного обеспечения, статистически возможного количества ошибок и времени передачи программного кода по сети. Для программной системы, реализованной в соот-

ветствии с парадигмой неравновесного программирования, снижение трудоемкости составляет: для одного элементарного пакета работ в $\frac{E_{2i}}{E_{1i}} \approx 30,6$ раза;

для всего проекта в $\frac{E_2^{95\%}}{E_1^{95\%}} \approx 79,73$ раза; количество строк исходного кода уменьшилось в 17 раз (по сравнению с реализацией с использованием парадигмы агент-ориентированного программирования).

Результаты, полученные в диссертационной работе, являются теоретической основой для построения самоорганизующихся программных систем с самоорганизацией континуального типа, обладающих практической полезностью.

Полученные и приведенные в диссертации результаты теоретических исследований, обладающие целостностью, логической стройностью, внутренней непротиворечивостью и согласованностью с результатами предыдущих исследований, а также результаты проведенных и представленных в диссертации экспериментальных исследований, воспроизводимые, устойчиво повторяемые, непротиворечивые, подкрепленные результатами расчетов по общепринятым методикам и актами внедрения и реализации, позволяют сделать вывод, что задачи исследования полностью решены, а цель исследования достигнута.

ОСНОВНЫЕ ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

Статьи в изданиях, входящих в систему цитирования SCOPUS

1. Kol'chugina, E. A. Self-organizing software systems with distributed artificial intelligence / E. A. Kol'chugina // Automatic Control and Computer Sciences. – 2015. – Vol. 49, № 4. – P. 216–220. (© Allerton Press, Inc., 2015). – DOI: 10.3103/S0146411615040069¹

Статьи в изданиях, рекомендованных ВАК РФ

2. Кольчугина, Е. А. Модель эволюционирующего программного обеспечения / Е. А. Кольчугина // Известия высших учебных заведений. Поволжский регион. Сер.: Технические науки. – 2006. – № 6 (27). – С. 78–86.

3. Кольчугина, Е. А. Эволюция расписаний как средство разработки параллельного алгоритма поведения цифрового организма / Е. А. Кольчугина //

¹ Перевод на английский язык, опубликованный Allerton Press, Inc.
Оригинальный текст в основной русскоязычной версии журнала:
Кольчугина, Е. А. Самоорганизующиеся программные системы с распределенным искусственным интеллектом / Е. А. Кольчугина // Автоматика и вычислительная техника. – 2015. – № 4. – С. 39–44.

Известия высших учебных заведений. Поволжский регион. Сер.: Технические науки. – 2008. – № 1 (5). – С. 45–52.

4. Кольчугина, Е. А. Неравновесное программирование / Е. А. Кольчугина // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2009. – № 3 (11). – С. 25–31.

5. Кольчугина, Е. А. Новые парадигмы распределенных вычислений и неравновесное программирование / Е. А. Кольчугина // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2009. – № 4 (12). – С. 3–11.

6. Кольчугина, Е. А. Исследование свойств цифровых организмов с помощью НК-подобных автоматов / Е. А. Кольчугина // Вестник компьютерных и информационных технологий. – 2011. – № 11. – С. 20–24.

7. Кольчугина, Е. А. Существование групп в составе сообществ программных агентов / Е. А. Кольчугина // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2011. – № 3 (19). – С. 37–43.

8. Кольчугина, Е. А. Структура цифрового организма в самоорганизующихся программных системах / Е. А. Кольчугина // Программные продукты и системы. – 2012. – № 2. – С. 51–54.

9. Кольчугина, Е. А. Применение методов теории нумераций для представления эволюции кода программных агентов в темпоральных базах данных / Е. А. Кольчугина // Программная инженерия. – 2012. – № 4. – С. 19–22.

10. Кольчугина, Е. А. Построение параллельных алгоритмов поведения программных агентов методами генетического программирования / Е. А. Кольчугина // Инфокоммуникационные технологии. – 2012. – Т. 10, № 3. – С. 40–44.

11. Кольчугина, Е. А. Применение методов генетического программирования к разработке web-интерфейсов / Е. А. Кольчугина, К. В. Заваровский // Прикладная информатика. – 2012. – № 5 (41). – С. 64–74.

12. Кольчугина, Е. А. Искусственные миры: пространственная организация / Е. А. Кольчугина // Программные продукты и системы. – 2013. – № 1. – С. 16–20.

13. Кольчугина, Е. А. Программные системы с самоорганизацией континуального типа / Е. А. Кольчугина // Программная инженерия. – 2013. – № 3. – С. 15–20.

14. Кольчугина, Е. А. Искусственные миры: темпоральная организация / Е. А. Кольчугина // Программные продукты и системы. – 2013. – № 2. – С. 212–215.

15. Кольчугина, Е. А. Искусственные миры: распределение данных / Е. А. Кольчугина // Программные продукты и системы. – 2013. – № 3. – С. 185–189.

16. Кольчугина, Е. А. Модели теории искусственной жизни и их классификация / Е. А. Кольчугина // Искусственный интеллект и принятие решений. – 2013. – № 4. – С. 3–13.

17. Кольчугина, Е. А. Самоорганизация программных систем на основе моделей искусственной химии / Е. А. Кольчугина // Нейрокомпьютеры: разработка, применение. – 2016. – № 2. – С. 77–83.

Статьи в прочих изданиях

18. Кольчугина, Е. А. Моделирование биоценозов / Е. А. Кольчугина // Новые информационные технологии и системы : тр. VI Междунар. науч.-техн. конф. – Пенза : Изд-во ПГУ, 2004. – С. 240–246.

19. Кольчугина, Е. А. Моделирование перемещений особи / Е. А. Кольчугина // Новые информационные технологии и системы : тр. VI Междунар. науч.-техн. конф. – Пенза : Изд-во ПГУ, 2004. – С. 246–249.

20. Kolchugina, E. A. Towards software design technology based on biosociosis model / E. A. Kolchugina // Современные информационные технологии : тр. Междунар. науч.-техн. конф. (Computer-based conference). – Пенза : Пенз. гос. технол. академия, 2004. – С. 184–188.

21. Кольчугина, Е. А. Программное обеспечение для реализации модели биоценоза / Е. А. Кольчугина // Современные информационные технологии : тр. Междунар. науч.-техн. конф. (Computer-based conference). – Пенза : Пенз. гос. технол. академия, 2005. – С. 135–139.

22. Кольчугина, Е. А. Модели функционирования распределенного программного обеспечения / Е. А. Кольчугина // Специальная техника средств связи : науч.-техн. сб. – Пенза : ФГУП ПНИЭИ, 2005. – С. 80–87.

23. Кольчугина, Е. А. Представление эволюции особей в одной модели теории искусственной жизни / Е. А. Кольчугина // Современные информационные технологии : тр. Междунар. науч.-техн. конф. (Computer-based conference). – Пенза : Пенз. гос. технол. академия, 2005. – Вып. 2. – С. 92–94.

24. Кольчугина, Е. А. Организация баз данных и распределенный информационный поиск / Е. А. Кольчугина // Современные информационные технологии : тр. Междунар. науч.-техн. конф. (Computer-based conference). – Пенза : Пенз. гос. технол. академия, 2005. – Вып. 2. – С. 95–97.

25. Кольчугина, Е. А. Влияние алгоритмов порождения процессов-особей на свойства модели поведения программного обеспечения / Е. А. Кольчугина // Современные информационные технологии : тр. Междунар. науч.-техн. конф. (Computer-based conference) / под ред. В. Б. Моисеева, Л. Г. Когельмана, С. В. Трубицкова. – Пенза : Изд-во Пенз. гос. технол. академии, 2006. – Вып. 3. – С. 47–51.

26. Кольчугина, Е. А. Применение моделей математической биологии для описания сообществ эволюционирующих программ / Е. А. Кольчугина // Аналитические и численные методы моделирования естественно-научных и социальных проблем : сб. ст. I Междунар. конф. – Пенза : ПДЗ, 2006. – С. 114–116.

27. Кольчугина, Е. А. Программные системы как аналог биоценозов живой природы / Е. А. Кольчугина // Новые информационные технологии и системы : тр. VII Междунар. науч.-техн. конф. – Пенза : Изд-во ПГУ, 2006. – Ч. 2. – С. 19–26.

28. Кольчугина, Е. А. Некоторые вопросы представления ДНК-структур цифрового организма / Е. А. Кольчугина // Новые информационные технологии и системы : тр. VII Междунар. науч.-техн. конф. – Пенза : Изд-во ПГУ, 2006. – Ч. 2. – С. 26–31.

29. Кольчугина, Е. А. Особенности современных технологий разработки программного обеспечения / Е. А. Кольчугина // Новые информационные техно-

логии и системы : тр. VII Междунар. науч.-техн. конф. – Пенза : Изд-во ПГУ, 2006. – Ч. 2. – С. 129–132.

30. Кольчугина, Е. А. Алгоритмы воспроизводства и видообразования в модели биоценоза / Е. А. Кольчугина // Вычислительные системы и технологии обработки информации : Межвуз. сб. науч. тр. – Пенза : Инф.-изд. центр ПГУ, 2006. – Вып. 6 (30). – С. 12–18.

31. Кольчугина, Е. А. Разработка программного обеспечения на основе модели биоценоза / Е. А. Кольчугина // Вычислительные системы и технологии обработки информации : Межвуз. сб. науч. тр. – Пенза : Инф.-изд. центр ПГУ, 2006. – Вып. 6 (30). – С. 28–34.

32. Кольчугина, Е. А. Алгоритмы обработки данных на основе методов биокибернетики / Е. А. Кольчугина // Современные информационные технологии : тр. Междунар. науч.-техн. конф. (Computer-based conference) / под ред. В. Б. Моисеева, Л. Г. Когельмана, С. В. Трубицкова. – Пенза : Изд-во Пенз. гос. технол. академии, 2006. – Вып. 4. – С. 68–71.

33. Кольчугина, Е. А. Результаты эксперимента по созданию эволюционирующего программного обеспечения / Е. А. Кольчугина // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2007. – № 1. – С. 54–60.

34. Кольчугина, Е. А. Генерация многовидовых иерархий эволюционирующих программ / Е. А. Кольчугина // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2007. – № 2. – С. 48–55.

35. Кольчугина, Е. А. Генерация параллельных алгоритмов на основе матрицы расписаний / Е. А. Кольчугина // Новые информационные технологии и системы : тр. VIII Междунар. науч.-техн. конф. – Пенза : Изд-во ПГУ, 2008. – Ч. 1. – С. 106–110.

36. Кольчугина, Е. А. НК-подобная автоматная модель цифрового организма / Е. А. Кольчугина // Новые информационные технологии и системы : тр. IX Междунар. науч.-техн. конф. – Пенза : Изд-во ПГУ, 2010. – Ч. 2. – С. 63–67.

37. Кольчугина, Е. А. Параллельная обработка информации коллективами эволюционирующих программных агентов / Е. А. Кольчугина // Нейроинформатика, ее приложения и анализ данных : материалы XVII Всерос. семинара (г. Красноярск, 2–4 октября 2009 г.) / под ред. А. Н. Горбаня, Е. М. Миркеса ; ИВМ СО РАН, отв. за вып. Г. М. Садовская. – Красноярск : ИПК СФУ, 2009. – С. 69–70.

38. Кольчугина, Е. А. Парадигма неравновесного программирования и другие современные парадигмы / Е. А. Кольчугина // Новые информационные технологии и системы : сб. науч. ст. XI Междунар. науч.-техн. конф. (г. Пенза, 25–27 ноября 2014 г.). – Пенза : Изд-во ПГУ, 2014. – С. 426–429.

39. Кольчугина, Е. А. Парадигма неравновесного программирования и индустрия разработки программ / Е. А. Кольчугина // Новые информационные технологии и системы : сб. науч. ст. XI Междунар. науч.-техн. конф. (г. Пенза, 25–27 ноября 2014 г.). – Пенза : Изд-во ПГУ, 2014. – С. 429–431.

40. Кольчугина, Е. А. Идентификация цифровых организмов / Е. А. Кольчугина // Новые информационные технологии и системы : сб. науч. ст. XI Меж-

дунар. науч.-техн. конф. (г. Пенза, 25–27 ноября 2014 г.). – Пенза : Изд-во ПГУ, 2014. – С. 431–432.

41. Кольчугина, Е. А. Алгоритм выполнения расширенного информационного запроса коллективом цифровых организмов / Е. А. Кольчугина // Современные тенденции в образовании и науке : сб. науч. тр. по материалам Международ. науч.-практ. конф. (28 ноября 2014 г.) : в 14 ч. – Тамбов : Консалтинговая компания Юком, 2014. – Ч. 3. – С. 98–99.

42. Кольчугина, Е. А. Разработка параллельных программ эволюционными методами / Е. А. Кольчугина // Нейроинформатика, ее приложения и анализ данных : материалы XXIII Всерос. семинара (г. Красноярск, 25–27 сентября 2015 г.) / под ред. А. Н. Горбаня ; ИВМ СО РАН, отв. за вып. М. Ю. Сенашова. – Красноярск : Институт вычислительного моделирования СО РАН, 2015. – С. 27–31.

43. Кольчугина, Е. А. Перспективы развития парадигм программирования / Е. А. Кольчугина // Проблемы развития современной науки : сб. ст. Международ. науч.-практ. конф. (1 декабря 2015 г., г. Уфа). : в 4 ч. – Уфа : АЭТЕРНА, 2015. – Ч. 3. – С. 55–58.

44. Кольчугина, Е. А. Передовые технологии программирования как средство преодоления цифрового неравенства / Е. А. Кольчугина // Роль науки в развитии общества : сб. ст. Международ. науч.-практич. конф. (20 декабря 2015 г., г. Казань) : в 3 ч. – Уфа : АЭТЕРНА, 2015. – Ч. 2. – С. 66–69.

45. Кольчугина, Е. А. Влияние стратегий поведения цифровых организмов на их выживание и устойчивость существования сообществ / Е. А. Кольчугина // Технические науки в России и за рубежом : материалы V междунар. науч. конф. (г. Москва, январь 2016 г.). – М. : Буки-Веди, 2016. – С. 1–3.

46. Кольчугина, Е. А. Биоинспирированные методы алгоритмизации и программирования / Е. А. Кольчугина // Информатика: проблемы, методология, технологии : сб. материалов XVI междунар. науч.-метод. конф. (Воронеж, 11–12 февраля 2016 г.) Секция 1. Проблемы информатики как науки. – Воронеж : Научно-исследовательские публикации, 2016. – С. 63–68.

47. Kolchugina, E. A. Adaptive Software Systems / E. A. Kolchugina // European Conference on Innovations in Technical and Natural Sciences. The 10th International scientific conference proceedings (February 02, 2016), Vienna. Ed. Busch P. – Vienna : «East West» Association for Advanced Studies and Higher Education GmbH, 2016. – P. 17–20.

48. Kol'chugina E. Modern Automated Software Development Methods Inspired by Biology / Kol'chugina E. // International Conference «Science and Practice: a New Level of Integration in the Modern World». Conference Proceedings. Scope Academic House, May 05, 2016, Sheffield, UK. – San Francisco : B&M Publishing, 2016. – P. 160–162.

Свидетельства о государственной регистрации программ для ЭВМ

1. Кольчугина, Е. А. Свидетельство об официальной регистрации программы для ЭВМ в Федеральной службе по интеллектуальной собственности, патентам и товарным знакам (Роспатент) № 2009613158 от 18.06.2009, Россия

«Программная платформа для сборки и выполнения эволюционирующих программных агентов Padme Server Yule v.1.4» / Кольчугина Е. А.

2. Кольчугина, Е. А. Свидетельство об официальной регистрации программы для ЭВМ в Федеральной службе по интеллектуальной собственности, патентам и товарным знакам (Роспатент) № 2009615700 от 13.10.2009, Россия «Программный шаблон эволюционирующего агента с параллельным алгоритмом поведения» / Кольчугина Е. А.

Научное издание

КОЛЬЧУГИНА Елена Анатольевна

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПОСТРОЕНИЯ
САМООРГАНИЗУЮЩИХСЯ ПРОГРАММНЫХ СИСТЕМ
С САМООРГАНИЗАЦИЕЙ КОНТИНУАЛЬНОГО ТИПА

Специальность 05.13.17 – Теоретические основы информатики

Редактор *А. Г. Темникова*
Технический редактор *М. Б. Жучкова*
Компьютерная верстка *М. Б. Жучковой*

Распоряжение № 12/124-2017 от 19.06.2017.

Подписано в печать 21.06.2017. Формат 60×84¹/₁₆.
Усл. печ. л. 2,33. Заказ № 328. Тираж 100.

Издательство ПГУ.
440026, Пенза, Красная, 40.
Тел./факс: (8412) 56-47-33; e-mail: iic@pnzgu.ru

