МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В. Ломоносова
Механико-математичекий факультет

На правах рукописи

Пирогов Михаил Викторович

РАЗРАБОТКА МЕТОДА ИНТЕЛЛЕКТУАЛИЗАЦИИ СЛОЖНЫХ СИСТЕМ НА ОСНОВЕ СРЕДЫ РАДИКАЛОВ

Специальность 05.13.17 –теоретические основы информатики

АВТОРЕФЕРАТ
диссертации на соискание ученой степени
кандидата физико-математических наук

Москва 2011

Работа выполнена на Федеральном государственном унитарном предприятии «Научно-производственное объединение им. С. А. Лавочкина»

Научный руководитель:	доктор физико-математических наук, профессор *Чечкин Александр Витальевич*
Официальные оппоненты:	доктор физико-математических наук, *Махортов Сергей Дмитриевич*
	кандидат физико-математических наук *Шундеев Александр Сергеевич*
Ведущая организация:	Московский институт новых информационных технологий ФСБ России

Зашита диссертации состоится 9 марта 2011 г. в 16 час. 45 мин. на заседании диссертационного совета Д 501.002.16 при Московском государственном университете имени М. В. Ломоносова по адресу: РФ, 119991, Москва, ГСП-1, Ленинские горы, дом 1, Московский государственный университет имени М. В. Ломоносова, Механико-математический факультет, аудитория 14-08

С диссертацией можно ознакомиться в библиотеке Механико-математического факультета МГУ (Главное здание, 14 этаж).

Автореферат разослан « 9 » февраля 2011 г.

Ученый секретарь
диссертационного совета,
доктор физико-математических наук А. А. Корнев

Общая характеристика работы

Актуальность темы. В данной работе будем рассматривать системы, для которых характерны: масштабные цели; продолжительный жизненный цикл; необходимость больших объемов разнообразных ресурсов; большое число составляющих, их свойств и связей; наличие в качестве составляющих СС специалистов и коллективов специалистов; недопустимость нарушений функционирования, которые во время эксплуатации невозможно было бы исправить; необходимость использования различных точек зрения на проблемную область и масштабирования при решении задач жизненного цикла СС. Такие системы будем называть *сложными системами (СС)*. Примером СС является АКПУ-Э - автоматизированный комплекс планирования и управления, входящий в состав космического комплекса «Электро» ФГУП НПО им. С.А. Лавочкина. АКПУ-Э входит в состав наземного комплекса, который, в свою очередь, входит в состав космического комплекса наряду с космическим аппаратом. АКПУ-Э предназначен для планирования работы космического аппарата и целевой аппаратуры на требуемом временном промежутке. Планирование осуществляется в соответствии с запросами потребителей и с учетом ограничений на работу бортовых и наземных средств. АКПУ-Э включает в себя математические, программные, информационные, технические, методические, лингвистические, организационные, правовые, эргономические и метрологические средства. Проблемная область АКПУ-Э (жизненного цикла АКПУ-Э) имеет сложную структуру, в ней много составляющих и связей разных типов. Множество внешних и внутренних связей динамически изменяется в течение всего жизненного цикла, включающего этапы создания и эксплуатации АКПУ-Э. Множество специалистов разных специальностей и коллективов специалистов принимает участие в образовании горизонтальных и вертикальных связей проблемной области в целях решения задач жизненного цикла АКПУ-Э. При этом используется имеющийся богатый опыт проектирования, изготовления, испытаний и эксплуатации космических систем ФГУП НПО им. С.А. Лавочкина. Необходимость СС очевидна. В то же время из разных источников постоянно появляется информация, свидетельствующая о многочисленных проблемах СС, об их потенциальной конфликтности, что приводит к разнообразным, масштабным потерям и даже катастрофам. Практика показывает, что проблема конфликтности СС остается, в целом, нерешенной. Таким образом, необходимо создание средств обеспечения бесконфликтности СС.

По нашему мнению, существует ключевая проблема СС – проблема информационно-системной безопасности (ИСБ) сложной системы, сформулированная А.В. Чечкиным, включающая в себя традиционное понятие информационной безопасности и имеющая следующие аспекты:

1. Информационная устойчивость СС: решение любой задачи жизненного цикла СС должно быть обеспечено независимо от формы и полноты исходной информации путем логического получения необходимой дополнительной информации. Обеспечение информационной устойчивости СС требует информационного подхода к проблемной области.

2. Системная безопасность СС: решение любой задачи жизненного цикла СС должно учитывать не только эту задачу, но и соответствующие последствия во всей проблемной области, системную целостность, требование бесконфликтности в течение всего жизненного цикла системы. Конфликтом в СС будем называть ситуацию, которая безусловно приведет к нарушению системной целостности СС. Обеспечение системной безопасности требует системного подхода к проблемной области.

ИСБ может быть обеспечена, по нашему мнению, на основе специального формализма СС и построения на его основе интеллектуальной моделирующей среды СС.

Проведенный в данной работе анализ имеющихся математических и программных средств СС показал их недостаточность для обеспечения ИСБ. В проблемной области СС выявлены противоречия, относящиеся как к теории, так и к практике. *Противоречие в теории*. С одной стороны – объективное требование единства и полноты формального описания проблемной области СС, налич[1] средств интеллектуализации проблемной области в целях обеспечения ИСБ-эволюции формальной модели СС. С другой – отсутствие достаточных средств обеспечения единства и полноты формального описания проблемной области СС, его ИСБ-эволюции. *Противоречие в практике*. С одной стороны – требование обеспечения бесконфликтности СС. С другой – ограниченные возможности современного математического и программного обеспечения СС. *Актуальность темы диссертации* определяется необходимостью ликвидации выявленных противоречий. Исходя из изложенного сформулирована научная задача исследования: разработка метода обеспечения ИСБ СС, основанного на концепции среды радикалов, а также методики решения задач в среде радикалов, обеспечивающей ИСБ. Представленные соображения свидетельствуют о том, что тема диссертации соответствует положениям Паспорта специальности 05.13.17 «Теоретические основы информатики».

Цель диссертационной работы состоит в создании и исследовании информационных моделей, моделей данных и знаний, методов работы со знаниями и методов обнаружения новых знаний для применения их в целях обеспечения ИСБ СС. Необходимо разработать принципы создания и основные конструкции языка представления знаний, интегрированные средства представления знаний СС, отображающие динамику процессов, протекающих в проблемной области. Для достижения этой цели сформулированы и решаются следующие задачи:

- разработка метода обеспечения ИСБ СС;
- разработка модели и методики разрешения конфликтов СС;
- разработка практических рекомендаций по применению метода обеспечения ИСБ СС.

Цель работы и перечисленные задачи соответствуют положениям паспорта специальности 05.13.17 – теоретические основы информатики.

На защиту выносятся следующие основные результаты.

- Метод обеспечения ИСБ СС, основанный на концепции среды радикалов и включающий радикал-активатор, модели запросов и ответов в среде радикалов.
- Модель и методики оценки конфликтов и средств их разрешения в среде радикалов.
- Интерфейс и практические рекомендации по применению метода обеспечения ИСБ СС при решении реальных задач жизненного цикла СС.

Методы исследования. Для формализации используемых понятий и проведения доказательств используются следующие методы:

методы математической логики, включая исчисление предикатов;

методы логического программирования;

методы дискретной математики, в том числе, теории графов, сетей Петри;

методы объектно-ориентированного моделирования;

методы математической информатики.

Научная новизна результатов диссертации состоит:

в разработке нового подхода к формализации проблемной области СС широкого класса на основе концепции среды радикалов в целях обеспечения ИСБ путем построения и развития полномасштабной моделирующей среды – среды радикалов СС, учитывающей как статический и динамический, так и эволюционный аспекты проблемной области;

в создании формальных моделей конфликтов проблемной области СС и методов их разрешения в среде радикалов;

в разработке интерфейса среды радикалов и практических рекомендаций по применению разработанных формальных средств обеспечения ИСБ для широкого класса СС.

Практическая значимость диссертационной работы заключается в следующем:

1. Результаты работы могут эффективно применяться при решении задач любого этапа жизненного цикла любой СС. На основе единого формализма предлагаемый метод упорядочивает, нормализует и делает прозрачной разработку и эксплуатацию математического, программного, информационного, технического, методического, лингвистического, организационного, правового, эргономического, метрологического и других видов обеспечения СС, решение задач всех этапов жизненного цикла таких систем в целях обеспечения ИСБ. Метод позволяет четко разграничить модели и методы СС и реализующие их программных средства, разрабатывать новый, стабильный в своей основе, пользовательский интерфейс.

2. Метод может применяться для независимой и объективной экспертизы СС и их составляющих с целью получения соответствующих оценок и рекомендаций.

3. Затраты на внедрение нового метода минимальны. Внедрение может осуществляться поэтапно, с учетом складывающейся ситуации. Метод не конфликтен по отношению к другим средствам СС, взаимодействие с которыми может быть, при необходимости, обеспечено.

Внедрение результатов работы. Выполнение работы, реализация и внедрение ее результатов осуществлялось во ФГУП НПО им. С.А. Лавочкина. Основные результаты исследования с 2003 г. применяются при создании и эксплуатации автоматизированных комплексов планирования и управления (АКПУ), предназначенных для планирования работы космического аппарата и его целевой аппаратуры. В том числе, внедрение и применение теоретических положений данной диссертации осуществлялось для решения задач жизненного цикла комплекса АКПУ-Э, входящего в состав космического комплекса «Электро» с космическим аппаратом «Электро-Л». Практическое использование полученных результатов показало их работоспособность и эффективность на всех этапах жизненного цикла АКПУ. Разработана методика использования схем радикалов для АКПУ, которая применяется в настоящее время и развивается во ФГУП НПО им. С.А. Лавочкина для решения текущих задач жизненного цикла АКПУ. Также с 2003 г. результаты исследования используются в учебном процессе Государственного технического университета МАИ при прохождении студентами производственной практики во ФГУП НПО им. С.А. Лавочкина.

Апробация работы. Основные результаты диссертационной работы докладывались и обсуждались на: VII конференции «Обратные и некорректно поставленные задачи», посвященной памяти акад. А. Н. Тихонова в связи с 95-летием со дня рождения, МГУ им. М. В. Ломоносова, Факультет вычислительной математики и кибернетики (г. Москва, 2001 г.); VIII конференции «Обратные и некорректно поставленные задачи», МГУ им. М. В. Ломоносова, Факультет вычислительной математики и кибернетики (г. Москва, 2003 г.); IX Международной конференции «Интеллектуальные системы и компьютерные науки», МГУ им. М. В. Ломоносова, Механико-математический факультет (г. Москва, 2006 г.); Российской конференции «Математика в современном мире», посвященной 50-летию Института математики им. С. Л. Соболева СО РАН, Институт математики (г. Новосибирск, 2007 г.); Научном семинаре «Проблемы современных информационно-вычислительных систем», МГУ им. М. В. Ломоносова, Механико-математический факультет (г. Москва, 2007 г., 2010 г.); Научном семинаре в Институте точной механики и вычислительной техники (ИТМ и ВТ) им. С. А. Лебедева РАН (г. Москва, 2007 г.); Научном семинаре во Всероссийском институте научной и технической информации (ВИНИТИ) (г. Москва, 2007 г.); Научном семинаре в Институте системного программирования (ИСП) РАН (г. Москва, 2007 г.); Научном семинаре в Московском институте информационных технологий ФСБ России (г. Москва, 2007 г.); Всероссийской научно-технической конференции «Актуальные проблемы ракетно-космического приборостроения и информационных технологий», Федеральное государственное унитарное предприятие «Российский научно-исследовательский институт космического приборостроения» (ФГУП «РНИИ КП») (г. Москва, 2008 г.); Международной конференции «Современные проблемы математики, механики и их приложений»,

посвященной 70-летию ректора МГУ академика В.А. Садовничего, МГУ им. М.В. Ломоносова (г. Москва, 2009 г.); Научном семинаре в Ставропольском государственном университете (г. Ставрополь, 2009 г.); Научном семинаре в ВА РВСН им. Петра Великого (г. Москва, 2009г.); II Всероссийской научно-технической конференции «Актуальные проблемы ракетно-космического приборостроения и информационных технологий», посвященной 100-летию со дня рождения М.С. Рязанского, Федеральное государственное унитарное предприятие «Российский научно-исследовательский институт космического приборостроения» (ФГУП «РНИИ КП») (г. Москва, 2009 г.)

Публикации. По результатам работы опубликовано 13 научных статей, в том числе 4 статьи – в изданиях из перечня ВАК, и одна монография (коллективная).

Структура работы. Работа состоит из введения, трех глав, заключения и списка литературы (60 наименований). Общий объем диссертации – 141 страница.

Содержание работы

Во введении обоснована актуальность диссертационной работы, сформулирована ее цель, аргументирована научная новизна, представлены выносимые на защиту результаты исследований, а также аргументирована их практическая значимость.

Первая глава диссертации посвящена анализу проблемной области СС и постановке задачи разработки метода обеспечения ИСБ СС.

В первом параграфе первой главы приведены результаты анализа проблемной области СС, используемых в ней критериев и показателей. Сделан вывод о том, что проблема конфликтности СС остается, в целом, нерешенной. Таким образом, необходимо создание средств обеспечения бесконфликтности СС, которые соответствовали бы возрастающему уровню сложности и влиянию на окружающую среду таких систем. По мнению автора, ключевая проблема СС – это проблема ИСБ, которая может быть обеспечена на основе специального формализма СС и построения на его основе интеллектуальной моделирующей среды СС. Необходим метод обеспечения ИСБ СС. Сформулированы требования к такому методу.

Во втором параграфе первой главы приведены результаты подробного анализа современных математических и программных средств СС, проведенного с точки зрения необходимости охвата всех этапов жизненного цикла СС. Очевидны трудности такого анализа, связанные с разнообразием, многочисленностью и постоянным развитием таких средств. Анализировались средства дискретной математики, методы численного моделирования, методы искусственного интеллекта, в том числе, методы, основанные на графах и предназначенные для использования в СС. Была проанализирована также концепция среды радикалов из математической информатики. (Эта концепция была выбрана в качестве основы формализма СС, метода обеспечения ИСБ СС и рассмотрена в следующем разделе.) Рассмотрены средства структурного, объектно-ориентированного, логического и визуального программирования, средства CASE-технологии, а также унифицированного языка моделирования UML (Unified Modeling Language) и Internet-технологии. Анализировались современные средства разработки программных приложений, в том числе, приложений баз данных и современные СУБД (систем управления базами данных). Рассматривалась концепция CALS (Continuous Acquisition and Life Cycle Support – «непрерывные поставки и информационная поддержка жизненного цикла продукции»), использующая понятие интегрированной информационной среды, основанная на современных информационных технологиях и объединяющая принципы и технологии информационной поддержки жизненного цикла продукции на всех его стадиях.

В целях разработки формализма и метода обеспечения ИСБ СС были выделены следующие фундаментальные идеи, используемые разнообразными математическими и программными средствами, применяемыми в СС. Это – *идея объекта, идея связывания*

объектов, а также *идея представления объекта* (возможно, с заданной точностью) с помощью других объектов, связанных между собой – идеи задания, построения, получения, порождения, реализации, формирования, выражения, генерирования, разбиения, разложения.

Проведенный анализ позволил сделать следующие выводы. За короткий промежуток развития современных математических и программных средств создано множество средств решения задач жизненного цикла СС и накоплен большой опыт их применения, как отрицательный, так и положительный. В целом, средства современные СС обладают разнообразными и значительными возможностями. С другой стороны, современные математические и программные средства СС представляют собой множество в значительной степени разнородных инструментов. И это - несмотря на разработку специальных средств, которые должны служить обеспечению необходимого взаимодействия и интеграции. Эти инструменты постоянно изменяются, развиваются, не только предлагая пользователям новые функциональные возможности, но и ставя перед ними разнообразные проблемы. Нередко изменения носят маркетинговый характер. Информатизация проблемных областей, достигаемая прикладными системами, созданными с использованием таких средств, является, как правило, фрагментарной. Отсутствуют единые, общепринятые, обязательные к применению, полные и нормализованные описания проблемных областей, а также правила преобразования таких описаний, включающие обязательную обработку последствий всех изменений, что существенно затрудняет обмен информацией и принятие решений. Средства СС, как правило, не интеллектуальны. Основным документом СС, несмотря на широкое применение программных средств, по-прежнему остается текстовый документ на естественном языке, в котором преобладают неформальные описания. Современные средства не обеспечивают непосредственно эволюционный аспект СС в его единстве со статическим и динамическим аспектами. Не учитываются, в общем случае, цели, ресурсы и конфликты СС в их единстве. Документация СС, справочные и обучающие подсистемы создаются, во многом, отдельно от основной системы, что приводит к неточностям, недостаточной полноте описаний и трудностям в оперативном получении необходимой информации. Не обеспечивается, как правило, полная система контрольных примеров (тестов). Таким образом, в целом рассмотренные математические и программные средства не являются единым в своей основе комплексом, непосредственно нацеленным на обеспечение ИСБ-эволюции СС. Практика решения задач жизненного цикла СС показывает, что проблема ИСБ в необходимом объеме не решается, что оказывает значительное отрицательное влияние (порой, вначале скрытое) при решении задач жизненного цикла СС. К тому же, приобретение, внедрение, освоение и последующая эксплуатация таких средств требует больших затрат временных, финансовых и людских ресурсов, а качество получаемых решений далеко не всегда соответствует желаемому. Основная причина такого положения дел, как представляется, заключается в отсутствии единого формализма СС и метода обеспечения ИСБ СС, на который могли бы опираться различные инструментальные и прикладные системы во всех своих аспектах. СС и ее составляющие, а также все составляющие проблемной области должны рассматриваться как математические объекты. Частные математические модели, представленные с помощью единого формализма, использующие их прикладные системы должны обеспечивать специалистам полное владение ситуациями жизненного цикла СС, решение всех целевых задач в заданные сроки с обеспечением ИСБ.

В третьем параграфе первой главы рассмотрен общий подход к моделированию СС на основе концепции среды радикалов из математической информатики, выбранной, по результатам анализа математических и программных средств, в качестве основы формализма СС. Понятие *радикала* является главным понятием математической информатики. Под радикалом понимается любая функциональная система, имеющая два доступных извне состояния: *активное и пассивное*. Активный радикал функционирует, согласно своему предназначению, а пассивный радикал - нет. Он как бы выключен. Множество радикалов со

связями между собой образуют *среду радикалов*. Вопросами активирования среды радикалов занимаются системы, называемые *активаторами*. Система всех активных радикалов среды радикалов образует, так называемый, *системоквант*, который определяет квант поведения среды на данный момент времени. Понятие радикала позволяет взглянуть на СС и ее окружение как на среду радикалов. Радикалами такой среды являются составляющие СС и ее окружения, их связи, задачи жизненного цикла СС, средства и методы решения таких задач, специалисты, нормативные документы и многое другое.

ИСБ СС обеспечивается с помощью интеллектуальной системы, являющейся информационно-программной надстройкой СС. Интеллектуальная система СС должна принимать такие решения (определять такие системокванты поведения СС), которые обеспечивают ИСБ СС на протяжении всего жизненного цикла. Интеллектуальная система СС строится из рабочей и активирующей подсистем. Рабочая подсистема будет символьным представлением в интеллектуальной системе радикалов проблемной области СС, то есть будет информационной моделью проблемной области (картиной мира) СС. Активирующая подсистема должна содержать средства анализа, синтеза и активации рабочей области. Символы в рабочей области называются *нейрорадикалами*. Таким образом подчеркивается принципиальная связь строения рабочей области интеллектуальной системы с элементарным сенсориумом естественного мозга, состоящего из нейронов. Тем самым, рабочая подсистема интеллектуальной системы является информационной моделью проблемной области сложной системы в форме *среды нейрорадикалов*. Активирующая подсистема отвечает за ИСБ-решение задач жизненного цикла СС, она постоянно разрешает конфликты в среде нейрорадикалов. Используя модель проблемной области, активирующая подсистема должна создавать в рабочей подсистеме системоквант, который определяет ИСБ-поведение СС. В активирующей подсистеме интеллектуальной системы реализуются средства решения разнообразных задач по обеспечению ИСБ СС с использованием среды нейрорадикалов. Главные роли в такой подсистеме играют *активаторы и регуляторы*. *Активаторы* – это специализированные радикалы, нацеленные каждый на решение задач своего класса. Работа активаторов опирается на использование распределенной базы знаний о проблемной области в форме среды ультрарадикалов (блоков продукций и правил). *Регуляторы* – это специализированные радикалы, отвечающие за обеспечение целостности и бесконфликтности среды нейрорадикалов. Кроме того, в активирующей подсистеме используются командные (окрашивающие) радикалы, которые вычленяют в среде нейрорадикалов отдельные схемы и являются средством навигации в среде нейрорадикалов, а также средством сохранения опыта решения задач. В разделе рассмотрены основные этапы функционирования интеллектуальной системы СС.

По нашему мнению, концепция среды радикалов соответствует специфике СС. В основе концепции – понятие радикала – составляющей проблемной области СС, которая представляется множеством связанных между собой составляющих – средой радикалов. ИСБ-эволюция СС обеспечивается с помощью интеллектуальной системы, решающей как штатные, так и нештатные задачи и основывающейся на концепции среды радикалов.

В четвертом параграфе первой главы рассматривается цель и задачи исследования. Приводится постановка задачи разработки метода обеспечения ИСБ СС и методики решения задач в среде радикалов, которая имеет следующий вид.

Дано: Двойка вида All=<Objects, Connections>, где Objects={obj_1, obj_2, …,obj_k} - множество всех значимых объектов (составляющих) проблемной области СС, Connections={c_1, c_2, …, c_m} - множество всех значимых свойств и связей (отношений) проблемной области СС. Двойки, множества объектов которых есть подмножества множества Objects, а множества свойств и связей – подмножества множества Connections: S – СС; Tasks - штатные целевые задачи СС; Res – ресурсы СС; Over_S - надсистема СС; Actions - штатные воздействия надсистемы на СС; Reactions - штатные реакции на штатные воздействия надсистемы на СС; ISSafety - требования ИСБ СС.

Двойки вида ZAll={ZObjects, ZConnections}, представляющие проблемную область СС в нештатных ситуациях. Здесь ZObjects={zobj_1, zobj_2, …, zobj_n} - множество всех значимых объектов (составляющих) проблемной области СС в нештатной ситуации; ZConnections={c_1, c_2, …, c_p} - множество всех значимых свойств и связей (отношений) проблемной области СС в нештатной ситуации. Двойки, множества объектов которых есть подмножества множества ZObjects, а множества свойств и связей – подмножества множества ZConnections. Это – следующие двойки: ZS – СС в нештатной ситуации; ZTasks - нештатные целевые задачи СС; ZRes – ресурсы СС в нештатной ситуации; ZOver_S – надсистема СС в нештатной ситуации; ZActions - нештатные воздействия надсистемы на СС; ZReactions - нештатные реакции СС на штатные и нештатные воздействия надсистемы на СС; Conflicts - формальные конфликты проблемной области СС, являющиеся следствием наступления нештатной ситуации.

Пусть ISS – отображение, аргументы которого – двойки All и ZAll, а значения – истинностные значения трехзначной логики – true, false и null, соответствующие выполнению, либо невыполнению требований ИСБ СС, а также тем случаям, в которых неизвестно, выполняются или нет эти требования.

Требуется разработать метод обеспечения ИСБ СС и методики решения задач жизненного цикла СС, основанные на концепции среды радикалов и включающие следующие этапы: (1) разработка радикал-активатора, моделей запросов и ответов в среде радикалов; (2) разработка модели и методики оценки конфликтов и методика ухода от конфликтов, обеспечивающих ISS(All, ZAll)=true; (3) разработка интерфейса и практических рекомендаций по применению метода обеспечения ИСБ СС при решении реальных задач жизненного цикла СС.

В первой главе мы рассмотрели проблему информационно-системной безопасности (ИСБ) сложной системы (СС), системы показателей и критериев СС с точки зрения ИСБ и сделали вывод о необходимости специального формализма для моделирования проблемной области СС. Были сформулированы требования к методу обеспечения ИСБ СС и проанализированы с этой точки зрения современные математические и программные средства СС. Обоснован выбор концепции среды радикалов в качестве основы формализма моделирования СС. Сформулированы цель и научная задача исследования. Приведена формальная постановка задачи, согласно которой требуется разработать метод обеспечения ИСБ СС и методику решения задач жизненного цикла СС, основанные на концепции среды радикалов. Для решения поставленной задачи необходимо разработать модель проблемной области СС в форме среды радикалов.

Во второй главе диссертации рассматривается разработанный формализм схем радикалов, с помощью которого осуществляется построение модели проблемной области СС в форме среды радикалов.

В первом параграфе второй главы описывается построение модели среды радикалов на основе схем радикалов. Вначале рассматривается синтаксис разработанного формализма. Схемы радикалов вводятся следующим образом. Фиксируется *алфавит*. Из символов алфавита строятся *имена радикалов*. Вводится *звено* – базовая конструкция (базис) разработанного формализма – слово, состоящее из двух имен радикалов, соединенных с помощью символа алфавита`→` (прямое звено) или `←` (обратное звено). Например, N0→N1 – прямое звено (здесь N0 и N1 – имена радикалов). Из звеньев строятся *цепочки радикалов*. Например, четырехзвенная цепочка радикалов N0→N1→N2→N3→N4; (допускается запись N0→…→N4; - здесь многоточие … - метасимвол, используемый для сокращения записи). *Схемой радикалов* называется любое множество цепочек радикалов. *Среда радикалов* представляется с помощью *схемы радикалов*. Радикалы среды радикалов могут образовывать *ветвления, схождения и вложения*, которые также представляются схемами радикалов. Например, следующая схема радикалов является ветвлением: Name30→{Name31→Name32; Name41→Name42;}. Для удобства допускается

идентификация цепочек ветвления с использованием *служебных радикалов вида d[*] (d[*]SmthName)*. Таким образом, для ветвления Name30, можем иметь, например, следующую запись: Name30→ {d[1]→ Name31→ Name32; d[2]→ Name41→ Name42;}.

Удобно рассматривать геометрические отображения схем радикалов. На рисунке 1 представлен пример *типового геометрического отображения* схем радикалов, при котором схемы представляются с помощью векторов на плоскости. Отображается ветвление N30. На плоскости введена *прямоугольная система координат*, в начало которой помещен радикал N30. Оси координат H и V направлены вправо и вертикально вниз. Служебные радикалы d[1] и d[2], используемые для идентификации цепочек ветвления N30, обозначены как 1 и 2.

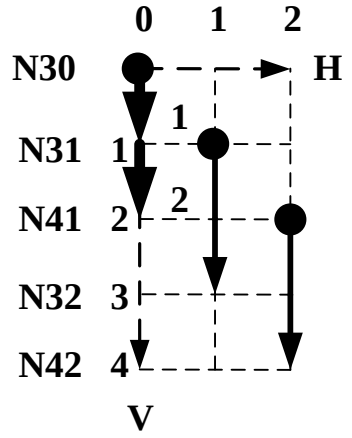

Рисунок 1 – Типовое геометрическое отображение ветвления N30

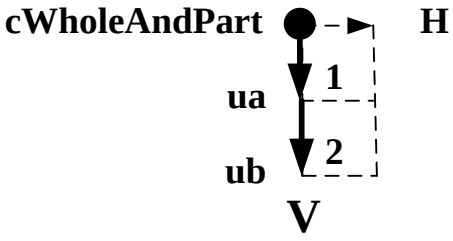

Рисунок 2 – Двуместный контейнер вхождения cWholeAndPart

Типовое геометрическое отображение схем радикалов широко используется в разработанном формализме. Оно положено нами в основу интерфейса при компьютерной реализации среды радикалов.

Далее рассматривается семантика разработанного формализма. Для представления проблемной области СС введены *стандартные схемы радикалов (стандартные радикалы)* – уникумы и контейнеры. Уникумы соответствуют компонентам проблемной области СС, а контейнеры – свойствам (связям) таких компонентов. *Уникумом (уникальным радикалом)* называется радикал, имя которого начинается с символа ‘u’ (от слова «unicum») и содержат *три индекса: u[1:*][2:*][3:*]SmthUnicum;*. С помощью *первого индекса* идентифицируется *тип уникального радикала*, например: целое число; конечная десятичная дробь; истинностное значение; единица измерения длины; составляющая проблемной области СС, принадлежащая некоторому классу составляющих. С помощью *второго индекса* идентифицируется *экземпляр уникума* определенного типа. С помощью *третьего индекса* идентифицируется *модификация уникума*. Допускается сокращенная запись вида *uSmthUnicum;, uSU;*.

С помощью радикал-контейнеров реализуется идея топологической фильтрации уникумов в среде радикалов. В математической информатике контейнеру соответствует многоместное отношение, в лингвистике, например, одноместному контейнеру соответствует термин «понятие», в логике предикатов контейнеру соответствует многоместный предикатный символ. *Контейнером* называется радикал, имя которого начинается с символа ‘c’ (от слова «container») и содержит *три индекса: c[1:*][2:*][3:*]SmthContainer* (допускается сокращенная запись: *cSmthContainer;, cSC;*). Индексы контейнеров определяются аналогично индексам уникумов. Всякий контейнер, соответствующий многоместному отношению (предикатному символу) обязательно связан со схемой радикалов, раскрывающей это отношение.

С помощью уникумов и контейнеров реализуется основная системообразующая идея координатизации среды радикалов – идея *единой координатной системы контейнеров (КСК)*: каждый уникум «вкладывается», в общем случае, во множество различных и разнотипных контейнеров (допускается также погружение контейнеров в контейнеры), причем, каждый уникум и каждый контейнер оказывается привязанными к общему зафиксированному *«началу» среды радикалов – радикалу FirstLink (FL)*. КСК позволяет осуществить *структуризацию* семейства радикалов, эффективно решать проблему *выбора* радикала, используя *запросы* и *радикал-активатор* (см. ниже).

Контейнеры являются радикалами ветвления среды радикалов. Соответствующие цепочки ветвления (направления, доступные в контейнерах) идентифицируются с помощью *служебных радикалов*, начинающихся с символа ‘d’ (от слова «direction»), имеющих вид *d[1:*][2:*]SmthDirect* (допускается сокращенная запись: *d[*]SD, dSD*) и предназначенных для ориентации (навигации) в среде радикалов. *Первый индекс* используется для идентификации самого направления в контейнере, *второй* – определяет порядковый номер рассмотрения этого направления при обработке контейнера в конкретной ситуации. Таким образом, контейнеры, в общем случае, имеют следующий вид: *cSC➔{dSD➔...; ...; dSD➔...;}*. Для контейнеров допускается использование записи вида cSC; и cSC➔…;.

С точки зрения решеток понятий и топологической координатизации множеств, одноместным контейнерам соответствуют *базисные понятия шкал решеток понятий*. Например, контейнер целых чисел, контейнер констант двузначной логики, контейнер конечных десятичных дробей и т.д. Тем самым *КСК* – это *система базисных понятий* различных шкал решеток понятий, которая позволяет реализовать идею *топологической фильтрации* отдельных уникумов в среде радикалов.

Приведем пример схемы радикалов - контейнера с вложенными в него уникумами. Двуместный контейнер вхождения cWholeAndPart, используется для описания непосредственного вхождения одних составляющих СС в другие. Пусть в составляющую ua входит составляющая ub, тогда имеем: cWholeAndPart➔ {d[1]Whole➔ ua; d[2]Part➔ ub;}. Геометрическое отображение этой схемы представлено на рисунке 2.

Если контейнеры соответствуют предикатным символам логики предикатов, то уникумы соответствуют термам логики предикатов, в том числе переменным (имена соответствующих радикалов имеют вид uvar*) и значениям функций - здесь используются контейнеры вида cf*. Для истинностных значений используются уникумы uFALSE и uTRUE (уникум uTRUE можно, по умолчанию, опускать). Для неопределенных значений используется специальный уникум uNULL. Для логических операций используются контейнеры cOR, cAND и cNOT. Для кванторов используются специальные контейнеры, например, cqEXIST, cqANY.

Уникумы и контейнеры называются *опорными радикалами*. Такие радикалы вместе образуют *среду (схему) опорных радикалов (опорную среду радикалов)*.

При активации среды опорных радикалов возможно использование дополнительных знаний в форме *ультрарадикалов*, образующих *ультрасреду*. Такое оснащение среды радикалов называется *ультраоснащением*. Для построения ульрасред введены схемы специального вида – *ультраконтейнеры типов 1 и 2*. *Ультраконтейнер типа 1* - это радикал с именем вида *cUltra1[1:*][2:*]SmthName* (допускается сокращенная запись *cU1SmthName*), определяющий схему-продукцию «если, то», которая является ветвлением. Одна ветвь – это схема-заключение вида d[1]Conclusion➔ ...; другая ветвь – это пустая схема-посылка вида d[2]Premise➔;. Схема-продукция имеет следующий вид: cUltra1[1:*][2:*]SmthName➔ {d[1]Conclusion➔ cSmthContainer➔…; d[2]Premis➔;}. Посылка обязательно является пустой схемой. Первый индекс в cUltra1[1:*][2:*]SmthName – это тип ультраконтейнера, второй индекс – экземпляр ультраконтейнера. Схемы, доступные по направлению d[1]Conclusion, могут содержать как уникумы, так и звенья-переменные вида uVarSmthName=;. *Ультрасреда* определяется множеством ультраконтейнеров. Пусть контейнер *cAllUltraContainers* объединяет ультраконтейнеры, используемые для описания некоторой ультрасреды. Пусть

опорная среда начинается с контейнера cAllContainers. Опорная среда и ультрасреда называются *связанными средами*, если существует схема вида FirstLink→ { d[*]→ cAllContainers→…; d[*]→ cAllUltraContainers→…; }, где FirstLink – *начало среды радикалов*. Полученную среду (схему) FirstLink→ … будем называть *ультраоснащенной средой (схемой)*. Ультраконтейнеры типа 2 имеют, в отличие от ультраконтейнеров типа 1, непустую посылку. *Ультраконтейнер типа 2* - это схема-продукция с непустой посылкой, имеющая следующий вид: cUltra2[1:*][2:*]SmthName→ {1 d[1]Conclusion→…; d[2]Premise→ cAND→ {2 d[1]And→…; … d[*]And→…; }2 }1. Схемы, связываемые ультраконтейнером типа 2, могут содержать как уникумы, так и звенья-переменные вида uVarSmthName=;. Ультраоснащенные среды радикалов реализуют *фразовую форму логики предикатов*. С точки зрения математической информатики, ультраконтейнеры типа 2, собранные в схемы ветвления (наборы), определяют *ультраоператоры*. Такие ультраоператоры обеспечивают отображения одних уникумов в другие в форме отображения информации об уникуме-аргументе в информацию об уникуме-значении.

Во втором параграфе второй главы начинается рассмотрение прагматики разработанного формализма. Вначале описывается модель формирования запросов и ответов на них в среде радикалов. Для решения задач анализа и синтеза сложной системы необходимо обеспечить *навигацию* в среде радикалов, а также *выделение* в среде радикалов тех или иных схем для последующего их использования. Проблема решается с помощью понятия помеченной схемы радикалов. *Помеченной (выделенной) схемой радикалов* называется множество схем радикалов, объединенных контейнером вида *cColor*, имеющим следующий вид: cColor→{d[1]SelectedColor→u(*); d[2]Chain→…; …; d[*]Chain→…;}. Здесь u(*) – ‘это уникумы u(1), u(2), …, соответствующие натуральным числам, с помощью которых определяется *«цвет» выделения*. Цепочки помеченной схемы доступны в контейнере cColor по направлениям d[2]Chain→…; …; d[*]Chain→…;. Контейнеры cColor – системообразующие контейнеры, с их помощью осуществляется выделение «слоев», построение иерархических структур в среде радикалов и др. При решении задач анализа и синтеза схем, эти иерархические структуры активируются с помощью специальных средств активации, которые мы рассмотрим ниже. Реализуется процесс передачи «возбуждения» от одних радикалов к другим в целях ИСБ-решения задач среды радикалов. Благодаря механизму пометок (выделения), в среде радикалов реализуется структура радикалов форме нейронных сетей.

Введены *средства активации* помеченных схем с целью использования их при решении задач анализа и синтеза. Среда радикалов может быть активирована с использованием схем-запросов. Будем различать запросы типов 1 и 2. *Запросы типа 1* используются для активации среды радикалов с целью подтверждения (опровержения) факта существования в ней некоторой схемы. Запрос типа 1 приводит к выделению соответствующей схемы в среде радикалов (если таковая схема имеется). *Запросы типа 2* содержат имена неопределенных схем и используются для активации опорной среды радикалов с целью связывания этих имен с конкретными схемами среды.

Запросы вводятся следующим образом. Пусть FirstLink - начальный радикал среды радикалов, описывающей проблемную область. Пусть звено FirstLink связано с контейнером cAllContainers, объединяющим используемые для описания проблемной области контейнеры вида cSmthContainer: FirstLink→ cAllContainers→ {d[1]→ SmthContainer; …d[*]→ cSmthContainer;}. *Запрос типа 1* - это схема вида: *?[1]Question[i]*→ cSmthContainer→ …; (сокращенно ?Q→…;). Здесь: ?[1]Question[i] – служебный радикал; 1 - тип запроса; i - порядковый номер запроса в последовательности запросов. Смысл запроса типа 1 следующий: выяснить, существует ли в среде радикалов схема, завершающая цепочку ?[1]Question[i]→…;. Для представления *ответа на запрос типа 1* введен специализированный контейнер *cAnswer*, объединяющий краткий (dBrief→...) и полный (dFull→...) ответы. Краткий ответ фиксирует факт подтверждения (опровержения)

существования в среде радикалов FirstLink➔... схемы, соответствующей схеме запроса. Полный ответ представляет собой вызванный запросом результат навигации в среде радикалов. Полный ответ - это: либо схема-маршрут, содержащая искомую схему, либо (в случае отсутствия такового) пустая схема. *Ответ на запрос типа 1* - это схема вида: *cAnswer*➔ {d[1]Brief➔ ...; d[2]Full➔ ...;}. Для завершения цепочки d[1]Brief➔…; используется либо уникум uNO, либо уникум uYES. В случае неудачи в результате поиска ответа на запрос будем иметь cAnswer➔ {d[1]Brief➔ uNO; d[2]Full➔;} (допускается использовать также «пустой» радикал uNULL: d[2]Full➔ uNULL;). В случае успеха cAnswer➔ {d[1]Brief➔ uYES; d[2]Full➔ cSmthContainer➔ …;}

Продолжено использование визуализации среды радикалов в форме ее типового геометрического отображения. На плоскости введена прямоугольная систему координат, в начало которой помещен радикал FirstLink (FL). Оси координат HFL_cView* и VFL_cView* направлены, соответственно, вправо и вертикально вниз. Здесь *cView** - имена *контейнеров представления*. С их помощью фиксируются радикалы, доступные для наблюдения. Уникумы, контейнеры, переменные и запросы отображаются в точки, расположенные на вертикальной оси. Контейнеры (связи) представляются с помощью вертикально направленных векторов с указанием соответствующих направлений (1, 2, …) в представляемых контейнерах (напомним, что эти направления идентифицируются с помощью служебных радикалов вида dSmthDirect). В ряде случаев с целью упрощения записи, обозначения осей координат направлений в контейнерах, указание радикалов, недоступных для наблюдения, опускать. Таким образом, *срез среды радикалов* представляется *системой векторов*, активируемой запросами и изменяющейся во времени.

Запрос типа 2 обязательно содержит имена (имя) неопределенных схем и используется для активации опорной среды радикалов с целью связывания этих имен (имени) с конкретными схемами среды. *Запрос типа 2* - это схема вида: *?[2]Question[i]*➔ cSmthContainer➔ …; (сокращенно ?Q➔…;). Здесь: ?[2]Question[i] – служебный радикал; 2 – тип запроса; i – порядковый номер запроса в последовательности запросов. В контейнер cSmthContainer должно быть вложено конечное число радикалов-переменных, которые требуется определить в рамках данной среды радикалов. Допускается также вложение уникумов. Для *ответа на запрос типа 2* введен специализированный контейнер *cAnswers*, объединяющий все ответы *cAnswer*, полученные на запрос. Каждый из контейнеров cAnswer объединяет краткий (dBrief➔ ...) и полный (dFull➔ ...) ответы. Для краткого ответа используется контейнер переменных cvars, объединяющий связанные между собой имена неопределенных схем и конкретные схемы среды радикалов. Полный ответ – это, как и в случае ответа на запрос типа 1, маршрут, ведущий к искомой схеме. *Ответ на запрос типа 2* – это, в случае *успеха*, схема вида:

cAnswers➔
{1 d[1]Answer➔ cAnswer➔
{2 d[1]Brief➔ cvars➔ {3 d[1]var➔ uvarA= uA; ...; d[*]var➔ uvarZ= uZ; }3
d[2]Full➔ cSmthContainer➔ …;
}2 …
d[*]Answer➔ cAnswer➔
{2 d[1]Brief➔ cvars; d[2]Full➔ cSmthContainer➔ …; }2 }1.

В случае неудачи, в контейнерах cAnswer используются цепочки вида d[1]Brief➔cvars=uNULL; и d[2]Full➔uNULL;.

Опорная среда радикалов может быть связана, вообще говоря, с различными ультра средами. Используя запросы, мы можем активировать такие ультраоснащенные среды радикалов, получать ответы на запросы и осуществлять добавление новых схем. Результат активации зависит как от опорной среды радикалов, так и от ее ультраоснащения и запросов.

Введена схему, названную *радикал-активатором (активатором)*. С помощью этой схемы осуществляется поиск ответов на запросы. Контейнер cActivator объединяет пять

направлений: d[1]FL (d[1]FirstLink), d[2]Q (d[2]Question), d[3]AQs (d[3]ActiveQuestions), d[4]NowAQ (d[4]NowActiveQuestion) и d[5]Navigators. Направление 1 активатора ведет к начальному радикалу схемы, на которой ищется ответ на запрос: cActivator→d[1]FL→FL. Направление 2 активатора ведет к контейнеру исходного запроса cQ (cQuestion), объединяющему два направления: d[1]Q (d[1]Question), ведущее к исходному запросу ?Q[0], и d[2]Answer - по нему будет доступен найденный ответ (ответы) (вначале ответ – «пустой» радикал uNULL). По направлению 3 активатора приходим к контейнеру активированных запросов cAQs (cActiveQuestions). Активированные запросы – это, прежде всего, исходный запрос ?Q[0] и, возможно, промежуточные запросы, генерируемые активатором при его обработке. Каждый активированный запрос помещается в свой контейнер активированного запроса – контейнер вида cAQ (cActiveQuestion), доступный в контейнере cAQs по направлению d[*]AQ (d[*]ActiveQuestion).

Контейнер активированного запроса cAQ объединяет три направления. По направлению d[1]AQ контейнера cAQ доступен сам активированный запрос ?Q[*]. По направлению d[2]AC (d[2]ActiveContainer) контейнера cAQ доступен активированный контейнер среды радикалов, который используется при поиске ответа на запрос (вначале это радикал uNULL). По направлению d[3]Answer контейнера cAQ доступен ответ на активированный запрос (вначале это также радикал uNULL).

Вернемся к направлениям, непосредственно доступным в контейнере cActivator. Направление 4 активатора ведет к контейнеру cNowAQ (cNowActiveQuestion) – это контейнер запроса, обрабатываемого активатором в текущий момент времени. В контейнере cNowAQ имеются следующие направления: направление 1 ведет к обрабатываемому запросу; направление 2 – к активированному контейнеру среды радикалов, который используется при поиске ответа на обрабатываемый запрос; направление 3 – к ответу на запрос. Направление 5 активатора (направление d[5]Navigators) ведет к схемам нижнего уровня, используемым активатором при поиске ответов на запросы.

Активатор, пытаясь ответить на запрос, должен осуществлять поиск схем (перебор схем, сравнение их со схемами-образцами) и замену одних схем другими. Эти задачи решаются с помощью следующих схем, названных *базовыми схемами активации*. cGOTONext – контейнер, используемый для перебора радикалов в пределах одного контейнера. cGOTO – контейнер, аналогичный контейнеру cGOTONext, но допускающий переходы между различными контейнерами. cCompare – с помощью этого контейнера осуществляется сравнение схемы-образца с найденной схемой. cSubstitute – контейнер, используемый для замены «старой» схемы «новой» схемой. cTimeSubstitute – контейнер замены радикалов, отображающих время. cMessages – контейнер сообщений, принадлежащих некоторому множеству допустимых радикалов-сообщений. Это могут быть как внутренние сообщения активатора, образующие протокол его работы, так и внешние сообщения для диалога с другими схемами, в том числе со схемами – источниками управляющих сообщений (директив). Перечисленные контейнеры вкладываются в контейнеры-навигаторы cNavigator , помещаемые в контейнер cNavigators, доступный в контейнере cActivator по направлению d[5]Navigators. В контейнерах cNavigator и cNavigators доступны контейнеры состояний активатора (cState) и ультра контейнеры активатора, которые используются для анализа сообщений схем активатора, определения их состояния, принятия решений и перехода к дальнейшим действиям. К базовым схемам активации применяются промежуточные запросы, порождаемые системой ультраконтейнеров активатора. Действуя таким образом, активатор, с помощью контейнеров cNavigators, cNavigator и вложенных в них контейнеров, генерирует и решает «элементарные» промежуточные задачи поиска и замены схем другими схемами, в результате чего реализуется процесс поиска ответа на исходный запрос.

Процессы в среде радикалов удобно представлять преобразованиями векторов введенного типового геометрического отображения. При этом удобно рассматривать *части среды радикалов*, полученные путем *разрезания* плоскости геометрического отображения в

горизонтальном и вертикальном направлениях и последующего *склеивания* нужных фрагментов. Будем рассматривать следующие *«штатные» преобразования векторов на плоскости*: f1 – поворот; f2 – растяжение-сжатие; f3 – сдвиг и f4 – сложение. Эти преобразования активатор использует при поиске ответа на запрос (см. рисунок 3). В начале поиска ответа на запрос направления d[2]NowAC и d[2]AC соответственно контейнеров cNowAQ и cAQ оканчиваются «пустым» радикалом uNULL. Далее, активатор попытается найти в среде радикалов контейнер, однотипный контейнеру из исходного запроса (этот контейнер мы обозначим cG[2:1]), и сделать его доступным по рассматриваемым направлениям. Геометрически это выглядит как построение вертикально направленных векторов (работают схемы, вложенные в контейнер cNavigators). Пусть такой контейнер найден (обозначим его cG[2:0]). Вектора контейнера cNowAQ будут преобразованы: f3(f2(f1(cNowAQ_2_x2)))=cNowAQ_2_x3. На рисунке 3 это преобразование a. Соответственно преобразуются вектора контейнера cAQ: f3(f2(cAQ_2_x4))=cAQ_2_x5. Это преобразование b. Далее, активатор пытается согласовать между собой контейнеры

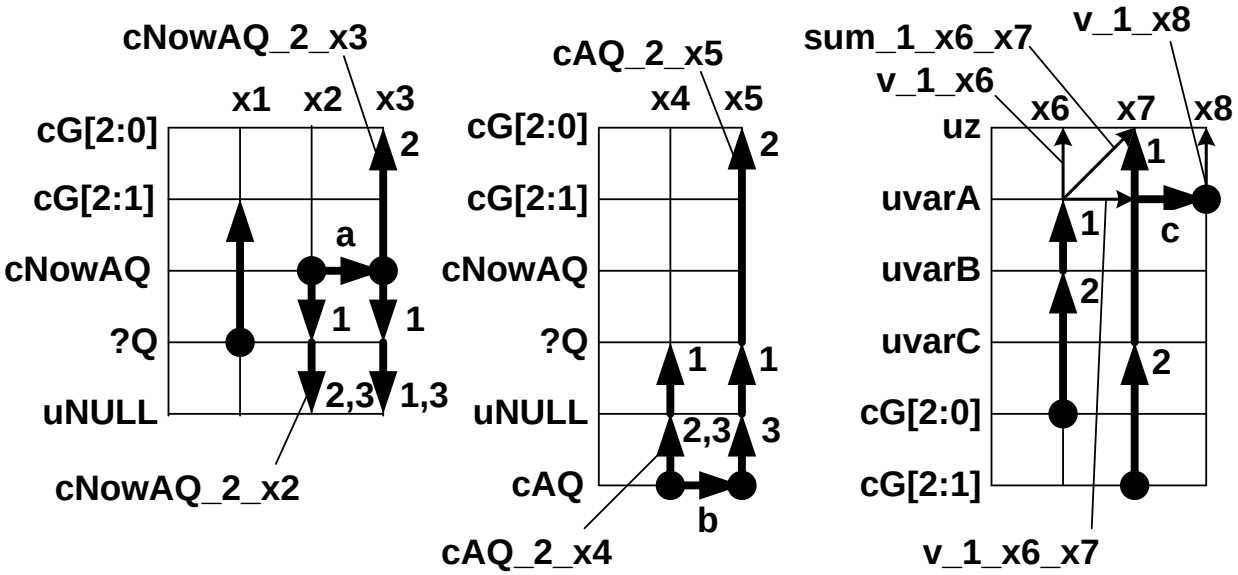

Рисунок 3 – Схемы преобразований a, b и c

cG[2:0] и cG[2:1]. Пусть эти контейнеры имеют вид: cG[2:0]→{d[1]→uvarA; d[2]→uvarB;} (такой контейнер может быть вложен в некоторый ультра контейнер) и cG[2:1]→{d[1]→uz; d[2]→uvarC;}. Значения переменных – уникум uNULL. Рассматриваются одноименные направления d[1] контейнеров cG. Строятся вспомогательные вектора v_1_x6 и v_1_x6_x7 так, как это показано на рисунке 3. Затем эти вектора складываются: f4(v_1_x6, v_1_x6_x7)=sum_1_x6_x7. Полученный вектор sum_1_x6_x7 поворачивается, сжимается и сдвигается вправо: f3(f2(f1(sum_1_x6_x7)))=v_1_x8. Полученный вертикально направленный вспомогательный вектор v_1_x8 соединяет горизонталь переменной uvarA и горизонталь ее нового значения – уникума uz. Таким образом, имеем преобразование векторов c. Рассмотренные преобразования векторов составляют *базовый цикл активатора в геометрической трактовке*. В базовом цикле активатора с помощью штатных преобразований векторов обеспечивается, в общем случае, следующее. *Поиск* контейнера, соответствующего контейнеру запроса с помощью построения вертикально направленных векторов и последующего сравнения векторов. *Замена* «старых» векторов «новыми» в случае нахождения нужного контейнера. Попытки *согласования контейнеров* между собой, построение (в случае успешного согласования) векторов, связывающих между собой

переменные и уникумы, а также, в общем случае, переменные. Таким образом, приходим к следующему *алгоритму работы активатора*, имеющему следующие основные этапы.

1. Активатор находится в режиме ожидания исходного запроса.

2. Ввод исходного запроса и размещение его в цепочке cActivator→ d[2]Q→ cQ→ d[1]Q→…;.

3. Активация запроса. Размещение запроса в контейнере активированного запроса cAQ и контейнере cNowAQ активированного запроса, обрабатываемого в настоящий момент времени. Для размещения запроса используются цепочки cActivator→ d[3]AQs→ cAQs→ d[*]AQ→…; и cActivator→ d[4] NowAQ→ cNowAQ→ d[1]NowAQ→…;.

4. Поиск в среде радикалов контейнера, подходящего для согласования. Ищется контейнер, однотипный целевому контейнеру запроса. Поиск осуществляется как в опорной среде, так и в ультрасреде - среди контейнеров, доступных в ультраконтейнерах по направлению d[1]Conclusion.

5. Размещение подходящего для согласования контейнера в цепочках cActivator→ d[3]AQs→ cAQs→ d[2]AC→…; и cActivator→ d[4]NowAQ→ d[2]NowAC→…;.

6. Попытка согласования целевого контейнера запроса и найденного контейнера (ультраконтейнера). Для согласования ультраконтейнера необходимо и достаточно согласовать контейнер, доступный по направлению d[1]Conclusion, а затем успешно обработать все направления контейнера cAND. Для этого, с помощью контейнеров, доступных в ультраконтейнере по направлениям d[*]And, генерируются промежуточные запросы. Затем эти запросы обрабатываются так же, как исходный запрос (п. 3). Если согласуется не ультраконтейнер, а простой контейнер, то согласование заключается в непосредственном применении к нему и к целевому контейнеру запроса правила связывания с помощью связки ‘=’ радикалов согласуемых контейнеров. Согласно этому правилу, связать между собой связкой ‘=’ можно только переменные либо переменную и уникум однотипных контейнеров. В том случае, если активатор не находит в среде радикалов контейнер, согласуемый с контейнером запроса, то он возвращается назад. Возврат осуществляется к последнему успешному запросу. С целью продолжения работы по поиску ответов активатор принудительно разрывает все связи, осуществленные в результате успешной обработки этого запроса, и возвращается к поиску контейнера, подходящего для согласования (п. 4).

7. Если все переменные исходного запроса оказываются связанными с уникумами, это означает, что первый ответ на исходный запрос получен. Ответ размещается в контейнере cvars→…; и активатор переходит к поиску других ответов на исходный запрос (п. 8). Если все запросы обработаны, возвраты более невозможны, а ответ, тем не менее, не найден, то активатор заканчивает работу и переходит в режим ожидания следующего исходного запроса (п. 1).

8. Активатор, применяя возвраты, ищет другие ответы на исходный запрос до получения в качестве ответа «пустого» радикала uNULL и перехода в режим ожидания (п. 1).

Приведенный алгоритм работы активатора основывается на широко применяемом *методе резолюций*. Отметим две *особенности применения* рассмотренного алгоритма.

Существуют среды радикалов и запросы, для которых применение активатора ведет к зацикливанию. Следующая особенность связана с логической равноценностью направлений d[*]And контейнеров cAND, принадлежащих ультраконтейнерам. В нашем описании предполагалось, что активатор, пытаясь согласовать ультраконтейнер и дойдя в нем до контейнера cAND, всегда начинает построение промежуточных запросов с направления d[1]And. Среда радикалов, моделирующая проблемную область СС, вообще говоря, не стабильна. Множество уникумов, контейнеров и ультраконтейнеров, а также запросов по разным причинам изменяется. Изменяется и оптимальный порядок обработки направлений контейнеров cAND. Если имеется система ультраконтейнеров, с помощью которой можно определять оптимальный порядок обработки направлений d[*]And, то следует применять *direct-радикалы с двумя индексами*: cAND→ d[1:*][2:*]And→…;. Первый индекс – константа

– идентифицирует само направление в контейнере cAND. Второй индекс – переменная – порядковый номер направления в очереди направлений контейнера cAND на обработку их активатором.

Сформулирована и доказана теорему о радикал-активаторе. В теореме рассматриваются ультраоснащенные среды радикалов. Пусть активация соответствующих ультраконтейнеров с помощью исходных запросов и радикал-активатора приводит к появлению новых контейнеров, которые затем могут быть *добавлены* к опорной среде радикалов. В таких случаях будем говорить, что исходный запрос и радикал-активатор *раскрывают* среду радикалов.

Теорема 1 (теорема о радикал-активаторе). Для любой нормализованной среды радикалов, представленной нормализованной системой векторов типового геометрического отображения, существует конечная последовательность запросов, с помощью которой радикал-активатор, используя штатные преобразования векторов, полностью раскрывает исходную среду радикалов за конечное число шагов.

Теорема доказана с помощью штатных преобразований векторов типового геометрического отображения схем радикалов.

Теорема 1 (теорема о радикал-активаторе) может быть охарактеризована как теорема о приведении среды радикалов к виду полностью раскрытой схемы, если это потребуется.

В третьем параграфе второй главы рассмотрен подход к формализации конфликтов в среде радикалов. В среде радикалов, представленной схемами радикалов, конфликты формально рассматриваются с точки зрения заполнения контейнеров. *Эволюция СС* определяется как изменение СС в процессе решения целевых задач или подготовки к их решению с использованием предоставленных для этого ограниченных ресурсов (при требовании их минимизации) при наличии нештатных воздействий и необходимости ухода от внутренних и внешних конфликтов (разрешения конфликтов). Формально эволюция СС представляется добавлением к схеме радикалов уникумов и контейнеров. В целях разрешения конфликтов выделены следующие *классы преобразований систем контейнеров*: *нуль-конфликт* преобразования - не приближающие, но и не удаляющие систему от конфликта; *плюс-конфликт и минус-конфликт* преобразования – приближающие систему к конфликту и, соответственно, удаляющие («уводящие») систему от конфликта. Таким образом, минус-конфликт преобразования контейнеров – средство разрешения конфликтов.

В целях *предсказания конфликтов* схемы классифицируются с помощью *контейнеров классификации*, в которых доступны ультраконтейнеры классификации, а также схемы-оценки. Для формализованного определения *класса схемы* введен контейнер cClsOfSh (cClassOfScheme). Контейнеры частной классификации cClsOfSh входят в контейнеры cClsesOfSh (cClassesOfScheme) более общей классификации. Введен также контейнер всех классов некоторой схемы cAClsesOfSh (cAllClassesOfScheme), содержащий, в общем случае, и контейнеры вида cClsesOfSh, и контейнеры вида cClsOfSh.

Для доступа ко всем задачам среды радикалов введен контейнер cATsks (cAllTasks), вложенный в контейнер cACsOfFL, непосредственно связанный с радикалом FL:

FL➔ {dACsOfFL➔ cACsOfFL; dAUCsOfFL➔ …;}.

Здесь использованы следующие сокращения: dACsOfFL – dAllContainersOfFirstLink, dAUCsOfFL – dAllUltraContainersOfFirstLink). Раскроем контейнер cACsOfFL:

FL➔ …➔ cACsOfFL➔ {dASs➔ cASs; dATsks➔ cATsks; dAAs➔ cAAs;…}.

Здесь *по первому направлению* доступен контейнер всех систем cASs (cAllSystems). *По последнему направлению* доступен контейнер cAAs (cAllActivators), объединяющий все активаторы cA (cActivator) среды: FL➔ …➔ cAAs➔{dA➔ cA; …; dA➔ cA;}. Контейнер задач cATsks (cAllTasks) имеет следующий вид:

FL➔ …➔ cATsks➔

{d[1]Tsk➔ uSTsk ←cACsOfTsk; …; d[*]Tsk➔ uSTsk ←cACsOfTsk;}.

Здесь uSTsk (uSmthTask) – уникумы-задачи. Контейнеры cACsOfTsk (cAllContainersOfTask) используются для выделения всех контейнеров уникумов-задач и имеют следующий вид:

FL→…→ uSTsk ←cACsOfTsk→ {dQs→ …; dBgnSh→ …; dRsrs→ …; dAnss→ …; dEndShs→ …; dSbTsks→ …; dMtds→ …; …}.

Здесь контейнер вида cACsOfTsk объединяет следующие схемы (доступные по соответствующим направлениям): dQs (dQuestions) – запросы задачи; dBgnSh (dBeginScheme) – начальная схема задачи; dRsrs (dResources) – ресурсы задачи; dAnss (dAnswers) – ответы; dEndShs (dEndShemes) – конечные схемы задачи; dSbTsks (dSubTasks) – подзадачи задачи; dMtds (dMethods) – методы, используемые при решении задачи.

Рассмотрим эти направления более подробно. С использованием запросов dQs активируются методы, с помощью которых осуществляется попытка решения задачи. dBgnSh – начальная схема, с которой будут работать активированные методы. Направление dRsrs приводит к схемам-ресурсам, имеющим отношение к задаче uSTsk. Это могут быть временные и другие ресурсы. ы будем говорить также о ресурсах (задач) жизненного цикла некоторого уникума.

Схема dAnss объединяет ответы (промежуточные и окончательные) на запросы задачи. Схема dEndShs – конечные схемы задачи. Эти схемы строятся с использованием схемы-ответа dAnss и начальной схемы dBgnSh, которую одна из конечных схем заменяет, продолжая таким образом жизненный цикл среды радикалов. Перейдем к направлению dSbTsks (dSubTasks). Задачи uSTsk среды радикалов могут быть разбиты на подзадачи с помощью контейнера cASbTsks (cAllSubTasks), вложенного в контейнер cCsOfTsk и доступного по направлению dSbTsks:

FL→…→ uSTsk ←cACsOfTsk→ {…; dSbTsks→ cASbTsks→
{dSbTsks→ cSbTsks; …; dSbTsks→ cSbTsks;} …;}.

Контейнеры cSbTsks (cSubTasks) имеют следующий вид:

FL→…→ cSbTsks→
{dSbTsk→ uSTsk ←cACsOfTsk; …; dSbTsk→ uSTsk ←cACsOfTsk;}.

Таким образом, в общем случае рассматривается конечное (бесконечное) число вариантов разбиения задачи на подзадачи и используется множество направлений dSbTsks и, соответственно, множество контейнеров cSbTsks. Для представления подзадач используются те же схемы, что и для представления задач, - уникумы-задачи uSTsk и контейнеры вида cACsOfTsk, выделяющие все контейнеры уникумов-задач. Подзадачи, в свою очередь, могут быть разбиты на другие подзадачи и т.д. до получения *штатных задач* uSBTsk (uSmthBasicTask), т.е. таких задач, для которых методы решения известны.

Методы рашения задач рассмотрены с точки зрения нормализации. Введено понятие *схемы-метода*. Эти схемы доступны по направлению dMtds (dMethods) в контейнере cACsOfTsk: FL→…→ uSTsk ←cACsOfTsk→ dMtds→ …;.

Введены схемы, предназначенные для получения и представления оценок других схем радикалов. В среде радикалов некоторой схеме uSShN (uSmthSchemeName) может соответствовать множество схем-оценок:

FL→…→ uSShN ←cAEsOfSh→ {dEOfSh→ cEOfSh; …; dEOfSh→ cEOfSh;}.

Здесь в контейнер cAEsOfSh (cAllEstimatesOfScheme) вложены контейнеры cEOfSh (cEstimateOfScheme) – контейнеры оценок схемы uSShN, доступные по направлениям dEOfSh. Контейнеры cEOfSh имеют следующий вид:

FL→…→ cEOfSh→ {dSh→…; dEmr→…; dEOfSh→…;}.

Здесь: dSh (dScheme) – оцениваемая схема; dEmr (dEstimator) –метод решения задачи оценки схемы; dEOfSh – схема-оценка. Направление dEOfSh ведет к схеме, полученной с использованием метода dEmr при решении задачи оценки. В частном случае, схема-оценка является, по существу, одномерным параметром оцениваемой схемы. Изложенный подход позволяет оценивать сложность схем. Причем, в общем случае, мы будем говорить о векторной сложности схем. Для оценки эффективности метода оцениваются соответствующие временные ресурсы. Применительно к жизненному циклу (задачам

жизненного цикла) некоторого уникума возможны следующие типы оценок: перспективные оценки; текущие (этапные) оценки, связанные с решением подзадач исходной задачи; итоговые оценки, получаемые с использованием специализированных схем-методов по окончании жизненного цикла уникума.

Рассмотрена *классификацию задач* по виду используемых в них *запросов*. (Рассмотрены *запросыв* об одномерных и многомерных *параметрах* некоторой СС uSS (uSmthSystem).

В четвертом параграфе второй главы рассмотрены более подробно схемы-методы, используемые для решения задач жизненного цикла среды радикалов. Схемы, представляющие задачи среды радикалов, вложены в контейнер cATsks (см. выше). Методы, используемые для решения некоторой конкретной задачи uSTsk, вкладываются в контейнер cACsOfTsk:

FL→…→cACsOfFL→…→cATsks→…→uSTsk ←cACsOfTsk→dMtds→…;.

Методы решения задачи uSTsk представляются уникумами вида uSMtd (uSmthMethod) и объединяются контейнерами вида cMtds (cMethods). Для выделения всех контейнеров метода используются контейнеры вида cACsOfMtd (cAllContainersOfMethod):

FL→…→ uSTsk ←cACsOfTsk→…→ cMtds→

{dMtd→ uSMtd→ cACsOfMtd; …; dMtd→ uSMtd→ cACsOfMtd;}.

Зафиксируем некоторую задачу uSTsk и некоторый метод ее решения uSMtd. В контейнер cACsOfMtd, связанный с методом uSMtd, вложим контейнер cNxtTsks (cNextTasks). В рассматриваемом случае этот контейнер объединяет все независимые друг от друга подзадачи, которые должны быть (могут быть) активированы непосредственно после активации метода uSMtd. (Для другого метода решения той же задачи uSTsk подзадачи из контейнера cNxtTsks будут, вообще говоря, другими.) Рассмотрение начато с того случая, прикотором в контейнер cNxtTsks вложена только одна задача, и эта задача является штатной. Для представления штатных задач будем используются уникумы вида uSBTsk (uSmthBasicTask):

FL→…→ uSTsk ←…→uSMtd ←cACsOfMtd→

{…; dNxtTsks→ cNxtTsks→ {dSbTsk→ uSBTsk→…;}…;}.

Для выделения всех контейнеров рассматриваемой задачи uSBTsk введен контейнер вида cACsOfBTsk (cAllContainersOfBasicTask). В этот контейнер вложен контейнер cBMtds (cBasicMethods), объединяющий уникумы вида uSmthBasicMethod (uSBMtd) – методы, каждый из которых решает задачу uSBTsk:

FL→…→ uSTsk ←…→uSMtd ←…→ uSBTsk ←cACsOfBTsk→{…

dBMtds→cBMtds→{dBMtd→ uSBMtd→…; …; dBMtd→ uSBMtd→…;}… }.

Активация любого из методов uSBMtd приводит к решению базовой задачи uSBTsk и, в рассматриваемом случае, к решению задачи uSTsk.

Вернемся к контейнеру cNxtTsks, вложенному в контейнер cACsOfMtd, соответствующий методу uSMtd решения задачи uSTsk. Пусть контейнер cNxtTsks объединяет конечное число независимых друг от друга задач:

FL→…→ uSTsk ←…→ uSMtd ←cACsOfMtd→ {…; dNxtTsks→ cNxtTsks→

{dSbTsk→ uSbTsk→…; …; dSbTsk→ uSbTsk→…;}…;}.

Пусть каждую из подзадач uSbTsk можно решить конечным числом базовых методов uSBMtd. Применим к каждой из подзадач любой из соответствующих методов. Пусть все активированные методы успешно завершат свою работу. В рассматриваемом случае это приведет к решению исходной задачи uSTsk.

Рассмотрен случай, при котором решение задачи uSTsk некоторым методом uSMtd сводится к последовательному решению конечного числа штатных задач, каждая из которых использует результаты решения предыдущей задачи (предыдущих задач).

В этом случае контейнер cACsOfMtd имеет следующий вид:

FL→…→ uSTsk ←…→ uSMtd ←cACsOfMtd→

{…; dSbTsksOfMtd→ cSbTsksOfMtd; dNxtTsks→ cNxtTsks; …;}.

Здесь контейнер cSbTsksOfMtd (cSubTusksOfMethod) содержит все задачи рассматриваемой последовательности задач:

FL→…→ uSTsk ←…→ uSMtd ←…→ cSbTsksOfMtd→
{dSbTsk→ uSBTsk→…; …; dSbTsk→ uSBTsk→…;}.

Контейнер cNxtTsks содержит первую задачу рассматриваемой последовательности задач:

FL→…→ uSTsk ←…→ uSMtd ←…→ cNxtTsks→
{dSbTsk→ uSBTsk→ cACsOfBTsk;}.

Добавим в контейнер cACsOfBTsk следующие схемы: dPrevTsk (dPreviousTask) – задача, предшествующая рассматриваемой в последовательности задач; dNxtTsk (dNextTask) – последующая задача. Теперь контейнер cACsOfBTsk имеет следующий вид:

FL→…→ uSTsk ←…→ uSMtd ←…→ uSBTsk ←cACsOfBTsk→
{ …; dBMtds→ cBMtds; dPrevTsk→…; dNxtTsk→ uSBTsk→…; …;}.

Решив штатную задачу одним из методов контейнера cBMtds, переходим к следующей задаче, доступной по направлению dNxtTsk. Решение последней задачи рассматриваемой последовательности штатных задач приводит, согласно методу uSMtd, к решению исходной задачи uSTsk.

Возможны следующие случаи заполнения контейнера cNxtTsks для фиксированной исходной задачи uSTsk и метода uSMtd ее решения: одна штатная задача; конечное число независимых друг от друга штатных задач (их можно решать параллельно); конечная последовательность штатных задач.

Пусть теперь подзадачи фиксированной задачи uSTsk по фиксированному методу uSMtd не являются, вообще говоря, штатными. Для заполнения контейнера cNxtTsks будем рассматривать три случая, аналогичные только что рассмотренным случаям. Если подзадача из контейнера cNxtTsks является штатной, то процесс разбиения исходной задачи в этом радикале завершается. Если подзадача из контейнера cNxtTsks не является штатной, то для нее возможно продолжение процесса разбиения с использованием трех рассмотренных случаев. Процесс разбиения исходной задачи uSTsk успешно завершается, если все терминальные контейнеры cNxtTsk будут содержать только штатные задачи uSBTsk, решаемые соответствующими штатными методами uSBMtd. Изложенный подход мы назовем *принципом базовых задач (методов)*.

Рассмотрим более подробно схемы базовых задач и базовых методов. Контейнер cACsOfBTsk, используемый для выделения всех контейнеров базовой задачи uSBTsk, аналогичен рассмотренному ранее контейнеру cACsOfTsk и имеет следующий вид:

FL→…→ uSTsk ←…→ uSMtd ←…→ uSBTsk ←cACsOfBTsk→
{dQs→…; dBgnSh→…; dRsrs→…; dAnss→…; dEndShs→…; dSbTsks→ uNULL; dBMtds→ cBMtds; dPrevTsk→…; dNxtTsk→ …;}.

Направление dSbTsks ведет к «пустому» радикалу, поскольку штатная задача не имеет подзадач. Контейнер cBMtds объединяет методы решения рассматриваемой задачи и имеет следующий вид:

FL→…→ uSTsk ←…→ uSMtd ←…→ uSBTsk ←…→ cBMtds→
{dBMtd→ uSBMtd→ cACsOfBMtd; …; dBMtd→ uSBMtd→ cACsOfBMtd;}.

Контейнер cACsOfBMtd (cAllContainersOfBasicMethod), используемый для выделения всех контейнеров некоторого штатного метода uSBMtd, имеет следующий вид:

FL→…→ uSTsk ←…→ uSMtd ←…→ uSBTsk ←…→ uSBMtd→ cACsOfBMtd→ {dAsOfBMtd→ cAsOfBMtd; dNxtAs→ cNxtAs;}.

Штатный метод uSBMtd использует активаторы, объединенные контейнером cAsOfBMtd (cActivatorsOfBasicMethod):

FL→…→uSTsk ←…→ uSMtd ←…→ uSBTsk ←…→ uSBMtd ←…→
cAsOfBMtd→{dAOfBMtd→ cA; …; dAOfBMtd→ cA;}.

Контейнер cNxtAs, вложенный в контейнер cACsOfBMtd, объединяет все независимые друг от друга активаторы, которые должны быть (могут быть) активированы непосредственно после активации метода uSBMtd:

FL→…→uSTsk ←…→ uSMtd ←…→ uSBTsk ←…→ uSBMtd ←…→ cNxtAs→{dNxtAOfBMtd→ cA; …; dNxtAOfBMtd→ cA;}.

Контейнеры cA (cActivator) мы рассматривали выше.

Добавим в контейнер cA (cActivator) следующие схемы: d[6]PrevAs→cPrevAs;, где контейнер cPrevAs объединяет все активаторы, которые должны завершить свою работу непосредственно перед активацией рассматриваемого радикал-активатора; d[7]NxtAs→cNxtAs;, где cNxtAs - контейнер, в который вложены все активаторы, для начала работы которых необходимо (но, вообще говоря, не достаточно) завершение работы рассматриваемого активатора.

Все штатные задачи мы соберем в контейнере cABTsks (cAllBasicTasks), который вложим в контейнер cACsOfFL, непосредственно связанный с начальным радикалом FL среды радикалов:

FL→{dACsOfFL→ cACsOfFL; dAUCsOfFL→…;}.

FL→…→ cACsOfFL→ {dASs→…; dATsks→…; dARAs→…; dABTsks→ cABTsks;…;}.

FL→…→ cABTsks→ {dBTsk→ uSBTsk→…; …; dBTsk→ uSBTsk→…;}.

В результате применения методов для решения задач жизненного цикла среды радикалов схемы преобразуются. Рассмотрены некоторые классы преобразований схем. Рассматриваются множества задач uSTsk для ультраоснащенных схем. Как было рассмотрено выше, уникумы-задачи uSTsk связаны с контейнерами cACsOfTsk, используемыми для выделения всех контейнеров уникумов-задач и имеющими следующий вид:

FL→…→ uSTsk ←cACsOfTsk→ {dQs→…; dBgnSh→…; dRsrs→…; dAnss→…; dEndShs→…; dSbTsks→…; dMtds→…;}.

Пусть преобразования некоторой схемы uSS, осуществляемые в результате решения задачи uSTsk некоторого класса таковы, что конечные схемы dEndShs отличаются от начальной схемы dBgnSh только схемами представления времени. Такие преобразования и схемы называются *тождественными* относительно рассматриваемого класса задач и методов их решения. Преобразования и схемы, не удовлетворяющие этому условию, называются *нетождественными*.

Если для некоторой схемы в результате решения задач некоторого класса мы получаем только тождественные преобразования, то схема называется *неразвертываемой* относительно рассматриваемого класса задач и методов их решения. Если же преобразования не являются тождественными, а схемы dEndShs можно представить как результат объединения тождественно преобразованных схем dBgnSh и схем, полученных в результате активации имеющихся ультраконтейнеров, то схема называется *развертываемой* относительно рассматриваемого класса задач и методов их решения.

Важное практическое значение имеет выделение и исследование классов схем. Рассматривается классификация задач (uSTsk): по виду преобразований схем (тождественные и нетождественные преобразования); по классам исходных запросов (dQs), начальных схем (dBgnSh) и ответов (dAnss).

Сформулированы и доказаны теоремы о рассмотренных преобразованиях схем с использованием такой классификации.

Теорема о преобразованиях неразвертываемых схем. Пусть исходные запросы к схеме принадлежат следующим классам запросов:

?[2]Query[*]→FirstLink→…→uSmthSystem←…→ cAllBigger1DParametersOfSystem→d[*]→uvarBigger1DParametersOfSystem=;

(запрос типа 2 о всех многомерных параметрах системы uSmthSystem, в результате активации такого запроса переменная uvarBigger1DParametersOfSystem может быть связана с одним или несколькими контейнерами вида cBigger1DParameterOfSystem);

?[2]Query[*]→FirstLink→…→uSmthSystem←…→
cAll1DParametersOfSystem→d[*]→uvar1DParametersOfSystem=;

(запрос типа 2 о всех одномерных параметрах системы uSmthSystem, результат активации запроса – связывание переменной uvar1DParametersOfSystem с контейнером (контейнерами) вида c1DParameterOfSystem).

Пусть схема, к которой обращены эти запросы, принадлежит следующему классу схем:

FirstLink→…→ uSmthSystem ←…→ cAllBigger1DParametersOfSystem→
uNULL;

FirstLink→…→uSmthSystem←…→cAll1DParametersOfSystem→
{d[1]→c[1:i]1DParameterOfSystem→…→uSmthFloat_i;}

FirstLink→d[*]→cAllUitraContainersOfFirstLink→uNULL;

(схема, описывающая систему единственным одномерным параметром, значение которого – конечная десятичная дробь; ультраконтейнеры отсутствуют).

Тогда ответы на запросы принадлежат, соответственно следующим классам схем:

cAnswers→d[*]Answer→…→uvarBigger1DParametersOfSystem=uNULL;

cAnswers→d[*]Answer→…→
uvar1DParametersOfSystem=c[1:i]1DParameterOfSystem;.

Схема FirstLink→…; - неразвертываемая, преобразования – тождественные.

Теорема о преобразованиях развертываемых схем. **Пусть** исходный запрос к схеме принадлежит следующему классу запросов:

?[2]Query[*]→FirstLink→…→uSmthSystem←…→
cAll1DParametersOfSystem→d[*]→c[1: ijk]c1DParameterOfSystem→…→
uvarSmthFloat=;

(запрос типа 2 о всех одномерных параметрах типа jik системы uSmthSystem; в результате активации такого запроса переменная uvarSmthFloat может быть связана с уникумами, соответствующими конечным десятичным дробям и вложенными в контейнеры типа jik).

Пусть схема, к которой обращен этот запрос, принадлежит следующему классу схем:

FirstLink→…→uSmthSystem←…→cAllBigger1DParametersOfSystem→…;

FirstLink→…→uSmthSystem←…→cAll1DParametersOfSystem→
{ d[*]→c[1:i0]1DParameterOfSystem→…→uSmthFloat_i0; …
d[*]→c[1:ij]1DParameterOfSystem→…→uSmthFloat_ij; …
d[*]→c[1:im]1DParameterOfSystem→…→uSmthFloat_im;
}.

(Контейнер типа ij здесь – единственный.)

FirstLink→d[*]→cAllUitraContainersOfFirstLink→d[*]→cUltra2→
{1 d[1]Conclusion→c[1:ijk]1DParameterOfSystem→…→uSmthFloat_ijk;
d[2]Premise→cAND→
{2 d[1]And→c[1:ij]1DParameterOfSystem→…→uSmthFloat_ij;
}2}1

(схема, описывающая систему с конечным числом одномерных параметров, значения которых – конечные десятичные дроби; ультраоснащение состоит из одного ультраконтейнера типа 2, определяющего условие возможности добавления к схеме контейнера c[1:ijk]1DParameterOfSystem→…→ uSmthFloat_ijk).

Тогда ответ на запрос принадлежит следующему классу схем: cAnswers→d[*]Answer→…→uvarSmthFloat=uSmthFloat_ijk;. Схема FirstLink→…; - развертываемая, преобразования – нетождественные. Исходный запрос приводит к полной развертке схемы.

Вторая глава была посвящена разработке модели проблемной области сложной системы (СС) в форме среды радикалов. Введены схемы радикалов – основа моделирования проблемной области. Разработано геометрическое отображение схем радикалов, которое положено в основу интерфейса среды радикалов. Выделены стандартные схемы радикалов – уникумы и контейнеры, с помощью которых реализуется идея координатизации среды радикалов. Уникумы и контейнеры – это опорные радикалы, образующие среду (схему) опорных радикалов. Введены средства активации среды радикалов – ультраконтейнеры (правила), контейнеры окраски и схемы-запросы. Ультраконтейнеры образуют ультрасреду, которая связывается с опорной средой радикалов (имеет место ультраоснащение среды радикалов) и активируется при помощи запросов. Поиск ответов на запросы осуществляется с помощью специальной схемы – радикал-активатора. Сформулировна и доказана теорема о радикал-активаторе. Введены схемы, предназначенные для разрешения конфликтов в среде радикалов. Представлены схемы-методы решения задач. Определены понятия неразвертываемой и развертываемой схем. Сформулированы теоремы о преобразованиях таких схем. Введенные средства-схемы радикалов позволяют перейти к разработке метода ИСБ-решения задач в среде радикалов.

В третьей главе диссертации рассматривается разработанный метод решения задач в среде радикалов.

В первом параграфе третьей главы введены этапы нормализации среды радикалов – целенаправленного процесса применения схем радикалов для обеспечения ИСБ СЭС. *Первый этап нормализации* состоит в разделении радикалов на два вида: уникумы и контейнеры. Осуществляется построение координатной системы контейнеров (КСК). Создаются средства визуализации среды радикалов, основанной на типовом геометрическом отображении схем радикалов и направленные на применение средств компьютерной графики. *Второй этап нормализации***.** В среде радикалов вводятся три взаимосвязанные части: опорная среда, ультрасреда и терминальная среда. Первая и вторая часть были рассмотрены выше. Третья часть - терминальная среда - образуется из радикалов-исполнителей и радикалов-датчиков, осуществляющих связь между опорными радикалами и ультрарадикалами. Ультрасреда вместе с терминальной средой определяют ультраоснащение среды радикалов, предназначенное для обеспечения ИСБ-решения задач жизненного цикла СС на основе выявления и разрешения конфликтов. *Третий этап нормализации* – организация - состоит в организации среды радикалов по специально введенным *принципам***.** Это, например, принцип первого радикала среды радикалов, принцип первого контейнера среды радикалов и т.д. На третьем этапе нормализации организуется библиотека стандартных радикалов. Рассматриваются принципы организации среды радикалов, сформулированные в терминах схем радикалов.

Во втором параграфе третьей главы рассматриваются основы разработанного метода решения штатных и нештатных задач жизненного цикла СС в среде радикалов. Решение штатных задач основывается на использовании контейнеров классификации и вложенных в них ультраконтейнеров с последующей активацией схемы-метода решения классифицированной задачи. При решении нештатных задач вначале также используются контейнеры классификации, задача классифицируется как нештатная, после чего активируются схемы, реализующие двунаправленный метод синтеза уникума (ухода от конфликтов), который будет рассмотрен ниже. Решение как штатных, так и нештатных задач приводит к использованию радикал-активаторов на нижних уровнях иерархии подзадач. Работа активатора рассмотрена на примере, демонстрирующем, в том числе, интерфейс разработанного метода.

В третьем параграфе третьей главы рассматриваются разработанные средства, направленные на решение проблемы конфликтности СС: метод оценки конфликтов в среде радикалов и двунаправленный метод ухода от конфликтов. Конфликты формально рассматриваются с точки зрения заполнения контейнеров. Для ИСБ-решения задач СС

необходимо постоянное прогнозирование и устранение конфликтов в среде радикалов в целях обеспечения ее ИСБ-эволюции. В течение жизненного цикла СС контейнеры среды радикалов могут преобразовываться, что может приводить к конфликтам между ними. Конфликты могут возникать при решении практически любых задач жизненного цикла СС. Например, пусть требуется построить уникум uGoal с некоторой характеризующей его целевой системой контейнеров и ультраконтейнеров. Это может быть сложная система, ее составляющая, некоторое управляющее действие и др. При решении задачи должны использоваться библиотечные схемы, решаться подзадачи и штатные задачи с целью связывания целевых схем с экземплярами библиотечных схем, заполнения соответствующих контейнеров. В итоге, целевой уникум может быть реализован иерархией составляющих-уникумов, характеризуемых своими контейнерам и ультраконтейнерами. В процессе построения такой иерархии в нее добавляются все новые и новые уникумы, контейнеры которых могут, вообще говоря, конфликтовать с контейнерами уже имеющихся уникумов и эти конфликты необходимо разрешать. В течение жизненного цикла СС представляющие ее а также окружающую среду схемы преобразуются. Это также может привести к конфликтам. Например, на этапе эксплуатации может быть зарегистрирован уникум, соответствующий некоторому нештатному внешнему воздействию, способному вызвать конфликт контейнеров, которого необходимо избежать, преобразуя доступные для этого схемы среды радикалов. Одна и та же схема одновременно может участвовать во множестве конфликтов различных классов.

Для определения класса конфликта введены контейнеры вида cSmthConflictClass, входящие в контейнеры вида cClassOfScheme и имеющие следующий вид:

cSmthConflictClass→
{dScheme→…; dUltraContainersOfClass→…; dEstimate→…;}.

Здесь dScheme – оцениваемая (классифицируемая) схема; dUltraContainersOfClass – ультраконтейнеры, используемые для классификации схемы; dEstimate – схема-оценка (соответствующая цепочка может оканчиваться уникумами: uFALSE, либо uTRUE, либо «пустым» радикалом uNULL). Схема среды радикалов называется *конфликтной* для заданной системы ультраконтейнеров, если в результате решения задачи классификации этой схемы получена оценка вида cSmthConflictClass→dEstimate→…→uTRUE;. Класс конфликта определяется контейнером cSmthConflictClass, а соответствующая система ультраконтейнеров должна быть доступна в контейнере cSmthConflictClass по направлению dUltraContainersOfClass.

По сформулированному определению, контейнер c0 *охватывает* контейнер c1, если c0 и c1 однотипны и все уникумы контейнера c1 являются уникумами контейнера c0, но не наоборот. Рассмотрим пример конфликтной схемы. Пусть имеем схему, содержащую два контейнера - c0 и c1, причем c0 охватывает c1. Пусть преобразования контейнеров приводят к появлению уникумов, принадлежащих преобразованному контейнеру c1, но уже не принадлежащих преобразованному контейнеру c0. Пусть активирована система ультра контейнеров, с помощью которой получена оценка вида cSmthConflictClass→ dEstimate→…→ uTRUE;. Таким образом, схема объявляется конфликтной из-за конфликта контейнеров c0 и c1, вызванного их преобразованиями.

Для описания преобразований контейнеров (схем) среды радикалов введены схемы вида:

cChangeScheme→{dBeforeScheme→…; dAfterScheme→…;…}.

Здесь dBeforeScheme – схема до преобразования, dAfterScheme – после преобразования. Преобразования могут привести к конфликтам, принадлежащим некоторым классам конфликтов. Именно в смысле преобразований и конфликтов контейнеров мы говорим о преобразованиях и конфликтах уникумов среды радикалов. Целесообразно использовать, по возможности, простые контейнеры для приближения более сложных контейнеров, и, соответственно, заменять преобразования сложных контейнеров преобразованиями простых контейнеров.

Далее, рассматриваются пары контейнеров, для описания которых введены контейнеры вида cSmthCPair→{dFirstC→…; dSecondC→…;}, связывающие первый и второй контейнеры пары. Для пары контейнеров определены *плюс конфликт преобразования* пары контейнеров, приближающие пару к конфликту, а также *нуль конфликт и минус конфликт преобразования*. Определены *безопасные и опасные преобразования* пары контейнеров относительно некоторого класса конфликтов. Так мы приходим к понятию *бесконфликтности жизненного цикла рассматриваемой пары контейнеров*. Далее, переходя от пар контейнеров к множеству всех контейнеров среды радикалов, приходим к понятию *бесконфликтности жизненного цикла СС (некоторого уникума)* в смысле бесконфликтности жизненного цикла всей системы контейнеров.

Выделены еще два класса контейнеров (преобразований контейнеров) – *контролируемые (управляемые)* и, соответственно, *неконтролируемые (неуправляемые)*. Управляемые контейнеры используются для «увода» системы контейнеров от конфликтов при решении задач жизненного цикла СС. Среда радикалов, в целях обеспечения ИСБ СС, должна автоматически «разводить» контейнеры, приближающиеся к конфликту, используя для этого управляемые контейнеры. Определены *штатные преобразования* схем среды радикалов. Такие преобразования не нарушают планируемого жизненного цикла сложной системы. В противном случае, преобразования - *нештатные*.

Поставлена *проблема базовых задач* о конфликтах: представляется важным выделить базовые задачи о конфликтах, к которым можно будет свести большинство задач о конфликтах СС. Рассмотрен пример, иллюстрирующий подход к решению этой проблемы, основанный на автоматически осуществляемых преобразованиях контейнеров.

Рассмотрен разработанный метод, позволяющий решать задачу построения (реализации) уникума, характерную для всех этапов жизненного цикла СС. Метод назван *двунаправленным методом* ухода от конфликтов в среде радикалов. С его помощью решается задача синтеза целевого уникума - СС, ее составляющей, некоторого управляющего воздействия, уводящего СС от конфликта. Суть метода – осуществление попыток использования библиотечных радикалов разных уровней для решения целевой задачи. Производится перебор и связывание большого числа библиотечных уникумов между собой с последующим анализом полученных результатов. Процесс осуществляется по принципу самоорганизации. Библиотечные уникумы, для которых допустимы изменения, постоянно преобразуются, т.е. изменяются соответствующие системы контейнеров и ультраконтейнеров, характеризующих такие уникумы. Такие преобразования индивидуальны для каждого уникума и зависят также от исходной задачи синтеза целевого уникума. Контейнеры некоторых уникумов таковы, что с их помощью могут образовываться связи с другими уникумами, и это приводит, в некоторых случаях в процессе постоянных преобразований уникумов, к появлению новых уникумов – преобразователей доступных ресурсов. Некоторые из этих новых уникумов характеризуются отрицательными контурами обратных связей, в которых изменение какого-либо параметра подавляет его дальнейшее изменение, что обеспечивает устойчивость уникума. Лишние ресурсы при этом либо не допускаются в уникум, либо выбрасываются из него, если это возможно. Некоторые из сформированных обратных связей, возможно, будут таковыми, что в своем функционировании они будут направлять уникум к его цели. Сформулирован следующий тезис.

Тезис о двунаправленном методе. Для любого целевого уникума и характеризующей его схемы и любой библиотеки стандартных радикалов, представленных нормализованными системами векторов, существует конечная последовательность шагов применения двунаправленного метода построения уникума, с помощью которой: либо осуществимо построение конечного числа схем, реализующих целевой уникум; либо делается вывод о том, что при данной библиотеке стандартных радикалов целевой уникум нереализуем.

Таким образом, метод решения задач в среде радикалов, опирающийся на введенные средства, заключается в следующем. С помощью имеющихся ультраконтейнеров осуществляется обнаружение угрозы конфликта, его прогнозирование и классификация. Если конфликт классифицирован как штатный, то и задача ухода от него, и метод ее решения – штатные. Решение задачи – уникум, уводящий среду радикалов от конфликта. Если конфликт – нештатный, то задача ухода от него – нештатная. Метода решения такой задачи в среде радикалов не существует – его необходимо построить. Для этого активируется двунаправленный метод ухода от конфликтов. Результатом его работы также может быть уникум, уводящий среду радикалов от конфликта. Построенная схема запоминается - происходит самообучение среды радикалов. В своей работе двунаправленный метод использует библиотеку стандартных радикалов. Таким образом, нештатная задача решается с помощью конструирования новой схемы из штатных схем. Двунаправленный метод использует также новые схемы, полученные в результате исследований среды радикалов на предмет ухода от разнообразных гипотетических конфликтов. Отметим, что такие исследования должны носить систематический характер. Если работа двунаправлено метода привела к выводу о нереализуемости целевого уникума, то это означает требование пополнения библиотеки стандартных радикалов новыми схемами.

В среде радикалов могут быть выделены специализированные уникумы, в терминах многоагентных систем – «агенты», предназначенные для решения задач тех или иных классов. Состав таких специализированных уникумов, связи между ними, правила обучения и самообучения формируются и изменяются в соответствии с целевыми схемами радикалов, схемами-ресурсами и библиотекой стандартных радикалов. Среда радикалов отличается от многоагентных систем существенно большей гибкостью, для среды радикалов частная перестройка и перенастройка структуры специальных уникумов – обычное дело. Кроме этого, среда радикалов отличается наличием единого формализма схем радикалов, который «понимают» все специализированные уникумы («агенты») и используют все схемы, с которыми эти уникумы могут быть связаны.

Обсудим вопросы непротиворечивости, выразимости и полноты для среды радикалов. Пусть за конечное число шагов построен целевой уникум – допустимое ИСБ-решение некоторой задачи. При этом предполагается, что библиотека стандартных радикалов фиксирована и непротиворечива. Непротиворечивость среды радикалов означает, что, ни за какое сколь угодно большое число шагов, полученный целевой уникум не может быть классифицирован как недопустимое решение. Выразимость и полнота для среды радикалов понимается в смысле возможности построения (реализации) целевых уникумов фиксированных классов с помощью библиотеки стандартных радикалов. Выразимость и полнота для среды радикалов обеспечиваются разнообразием имеющихся стандартных схем радикалов.

Для любой составляющей СС может быть построено в среде радикалов столько моделей, сколько потребуется для решения целевой задачи. С помощью схем радикалов могут быть описаны любые дискретные и любые непрерывные объекты, а также средства манипулирования ими. Другими словами, в среде радикалов доступны все средства классического моделирования.

В четвертом параграфе третьей главы представлены практические рекомендации по применению разработанного метода обеспечения ИСБ для реальной СС - автоматизированного комплекса планирования и управления АКПУ-Э. Путь практической реализации среды радикалов – создание и развитие системы обеспечения комплексных разработок – СОКР (см. рисунок 4).

СОКР – это СС, основанная на специализированных математических средствах – схемах радикалов - и обеспечивающая среду радикалов создания СС с помощью математического, программного и других видов обеспечений при приоритете математического обеспечения.

Основной принцип построения ПО СОКР – использование широко распространенных систем промышленного класса. Используются СУБД и средств разработки приложений для представления схем радикалов, манипулирования ими, а также для обеспечения автоматического преобразования соответствующих данных в программный код выбранного языка программирования. В то же время среда радикалов (ПО СОКР) должна интегрировать готовые методы решения актуальных задач и реализующие их программные средства. Такое построение ПО направлено на эффективное использование существующих наработок, а также на четкое разделение (1) уровня специалиста СС и (2) уровня программиста, чьи решения используются в составе СС. На уровне (1) с помощью схем радикалов формируется и развивается библиотека штатных радикалов, решаются целевые задачи СС. На уровне (2) с помощью схем радикалов формируется и развивается библиотека штатных радикалов – программных решений, в автоматическом/автоматизированном режиме генерируются программные решения, реализующие схемы радикалов уровня (1). Формализм схем радикалов связывает между собой эти уровни в смысле корректности реализации схем уровня (1) с помощью схем и программного кода уровня (2). Входной информацией для

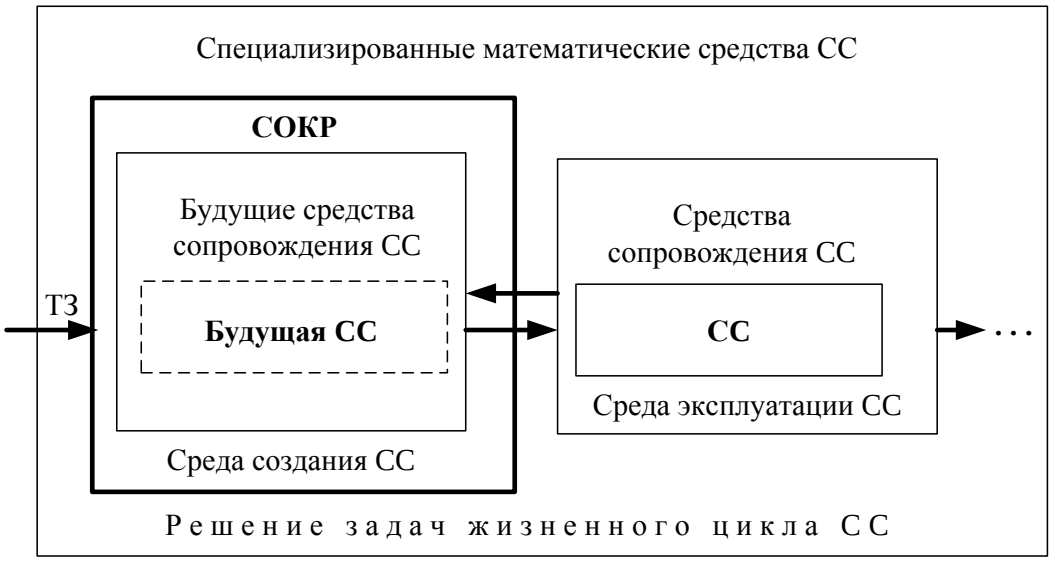

Рисунок 4 – Структурная схема СОКР

СОКР является техническое задание, преобразуемое в схему радикалов. Результат работы СОКР – схема радикалов, представляющая СС, снабженную средствами ее ИСБ-сопровождения на этапе эксплуатации. Среда радикалов создания и среда радикалов эксплуатации – части единой среды радикалов проблемной области СС. С помощью среды радикалов осуществляется решение как штатных, так и нештатных задач всех этапов жизненного цикла СС. Средства сопровождения СС анализируют состояние проблемной области, представленной схемой радикалов, и обеспечивают ИСБ-решение задач жизненного цикла СС. Результаты анализа, при необходимости, используются СОКР в целях изменения СС и средств ее сопровождения.

Разработанный метод применен для реального автоматизированного комплекса планирования и управления АКПУ-Э, входящего в состав космического комплекса «Электро». АКПУ-Э входит в состав наземного комплекса, который, в свою очередь, входит в состав космического комплекса наряду с космическим аппаратом. АКПУ-Э предназначен для планирования работы космического аппарата и целевой аппаратуры на требуемом промежутке времени. Структурирование проблемной области АКПУ-Э осуществлялось с помощью схем радикалов, что позволило проанализировать проблемную

область и выделить типовые схемы, характерные для АКПУ-Э (и видимо, для СС других классов). Рассмотрены примеры схем радикалов.

Разработаны *практические рекомендации* по применению средств разрешения конфликтов для решения средствами АКПУ-Э задачи формирования программы работы целевой аппаратуры (ЦА) космического аппарата. Исходными данными для этой задачи являются заявки пользователей на режимы работы ЦА - тройки вида <r, ti, tj>, где r – требуемый режим работы ЦА, [ti, tj] – временной промежуток, соответствующий этому режиму. Задача решается с помощью поэтапной обработки исходных и промежуточных данных с использованием ультраконтейнеров. При этом обеспечивается выявление конфликтных троек с установлением класса конфликта и последующим устранением конфликта.

Третья глава была посвящена разработке метода решения задач в среде радикалов. Были введены этапы нормализации среды радикалов и сформулированы принципы ее организации. Описан метод решения штатных и нештатных задач. На демонстрационном примере рассмотрена работа радикал-активатора, используемого при решении как штатных, так и нештатных задач. Разработан интерфейс среды радикалов. Описана методика оценки конфликтов в среде радикалов, а также двунаправленный метод ухода от них. Метод предназначен для решения центральной задачи построения (реализации) уникума, характерной для всех этапов жизненного цикла СС. Сформулирован тезис о двунаправленном методе. Обсуждаются вопросы непротиворечивости, выразимости и полноты для среды радикалов. Приведены практические рекомендации по применению метода обеспечения ИСБ и продемонстрировано его применение для реального автоматизированного комплекса планирования и управления беспилотного космического аппарата.

В **заключении** перечисляются основные результаты диссертационной работы.

- Описан метод обеспечения ИСБ СС, основанный на концепции среды радикалов и включающий радикал-активатор, модели запросов и ответов в среде радикалов.
- Разработана модель и методики оценки конфликтов и средств их разрешения в среде радикалов.
- Разработан интерфейс и практические рекомендации по применению метода обеспечения ИСБ СС при решении реальных задач жизненного цикла СС.

Разработанные метод, модель и методики можно охарактеризовать следующим образом:

Они универсальны – применимы к различным СС; охватывают весь жизненный цикл СС, статический, динамический и эволюционный аспекты СС. Основываются на едином специальном формализме СС. Учитывают цели, ресурсы и конфликты СС в их единстве. Обеспечивается упорядочение, нормализация и достигается прозрачность разработки и эксплуатации математического, программного, информационного, технического, методического, лингвистического, организационного, правового, эргономического, метрологического и других видов обеспечения СС при приоритете математического обеспечения. Учитываются все составляющие проблемной области СС, все их значимые свойства и связи, а также все задачи, методы, алгоритмы, программы, базы данных и так далее проблемной области. Разработаны средства для описания проблемной области, а также средства для поиска, выделения описаний и их активации. Обеспечивается оценка и, при необходимости, обработка всех последствий (возможно, отсроченная) изменений проблемной области СС. Обеспечивается методическая основа для создания и упорядоченного развития специализированных библиотек СС большого объема, ориентированных на решение проблемы ИСБ. Разработанные средства направлены на обеспечение процесса интеллектуализации СС, включающего: обеспечение решения как штатных, так и нештатных задач СС; обеспечение прогнозирования, классификации, оценки и разрешения конфликтов (ухода от конфликтов) СС; самообучение в процессе решения

нештатных задач СС. Разработанные средства ориентированы на применение современных программно-технических средств.

Обеспечивается визуализация проблемной области СС, ориентированная на применение современных средств компьютерной графики. При это важно, что среда радикалов представляется с помощью ее всевозможных сечений, что позволяет отобразить любые выделенные объекты и связи между ними. (Этого нельзя достичь с помощью обычных плоских схем.) Тем самым, разработаны основы для создания принципиально нового пользовательского интерфейса программного обеспечения СС.

Проводится четкое разграничение моделей, задач и методов прикладного уровня и реализующих их программных средств. Получена методическая основа для разработки математических и программных средств нового класса, ориентированных на обеспечение ИСБ СС.

Для целенаправленного применения схем радикалов по обеспечению ИСБ введены этапы формализованного описания проблемной области СС. Разработанные средства позволяют перейти к постепенному внедрению формальных описаний в разнообразную техническую документацию на естественном языке, в которой, как правило, преобладают неформальные описания. Затраты на внедрение разработанных средств минимальны. Предлагаемые средства не конфликтуют с другими методами, технологиями, средствами СС, взаимодействие с которыми, при необходимости, должно быть обеспечено.

Разработанные средства нацелены на создание и упорядоченное развитие полномасштабной интеллектуальной моделирующей среды, основанной на едином формализме и направленной на обеспечение ИСБ СС. Эти средства могут применяться для независимой и объективной экспертизы СС и их составляющих с целью получения соответствующих оценок и рекомендаций. Предлагаемые средства направлены на поддержку постепенного объединения таких моделирующих сред в единую глобальную моделирующую среду СС. Это соответствует объективному единству мира СС, погруженного в единую окружающую его естественную среду. Разработанные средства направлены как на теоретические исследования, так и на практическое применение.

Приведено краткое описание пути практической реализации среды радикалов с помощью системы обеспечения комплексных разработок (СОКР). Описаны результаты применения разработанных средств для реального автоматизированного комплекса планирования и управления АКПУ-Э, входящего в состав космического комплекса «Электро». Разработаны практические рекомендации по применению разработанных средств разрешения конфликтов для решения средствами АКПУ-Э задачи формирования программы работы целевой аппаратуры. Приведены экспериментальные данные, характеризующие эффективность применения разработанных средств при формировании программы работы целевой аппаратуры.

Автор глубоко благодарен научному руководителю – доктору физико-математических наук, профессору Александру Витальевичу Чечкину за постановку задач, обсуждение результатов и постоянное внимание к работе, а также доктору физико-математических наук, профессору Валерию Александровичу Васенину за ценные замечания.

Публикации по теме диссертации

1. Пирогов М.В. Обеспечение информационно-системной безопасности человеко-машинных систем при помощи среды радикалов / М.В. Пирогов // Математические методы решения инженерных задач: Сборник научных трудов – М.: МО РФ, 2006. – С. 85 – 126.
2. Пирогов М. В. Интеллектуальный стенд обеспечения информационно-системной безопасности сложных систем / М.В. Пирогов // Нейрокомпьютеры: разработка, применение – 2008. – №7. – С. 18 – 25.

3. Пирогов М.В. Метод ухода от конфликтов сложных систем / М.В. Пирогов // Информационно-измерительные управляющие системы – 2009. – Т. 7, № 3. – С. 34 – 48.
4. Пирогов М.В. Интеллектуализация сложных систем. Язык схем радикалов. Методы и алгоритмы / М.В. Пирогов, С.А. Радько, А.В. Рожнов, А.С. Савицкий, А.В. Чечкин // Монография под. ред. А.В. Чечкина и А.В.Рожнова – М.: «Радиотехника», 2008 – 96 с. [Пирогову М.В. принадлежит разработка формализма схем радикалов как основы для обеспечения информационно-системной безопасности сложной системы, а также разработка метода применения схем радикалов для создания интеллектуального стенда обеспечения информационно-системной безопасности.]
5. Пирогов М.В., Чечкин А.В. О радикалах проектирования сложных систем / М.В. Пирогов, А.В. Чечкин; под ред. В.В. Блаженкова, А.В. Чечкина // Математические методы решения инженерных задач: Научно-технические материалы. – М.: МО РФ, 1999. – С. 126 – 137. [Пирогову М.В. принадлежат анализ проблемной области и разработка основных требований к специализированному языку радикалов для сложных систем.]
6. Пирогов М.В., Чечкин А.В. Язык среды радикалов для решения задач жизненного цикла человеко-машинных систем / М.В. Пирогов, А.В. Чечкин; под ред. В.В. Блаженкова, А.В. Чечкина // Математические методы решения инженерных задач: Научно-методические материалы – М.: МО РФ, 2001. – С. 34 – 98. [Пирогову М.В. принадлежит разработка основных конструкций языка среды радикалов в форме схем радикалов, а также геометрического отображения таких схем.]
7. Пирогов М.В., Чечкин А.В. Технология решения задач в нормализованной среде радикалов / М.В. Пирогов, А.В. Чечкин // Фундаментальная и прикладная математика – 2009. Т. 15, вып. 3. – С. 205 – 223. [Пирогову М.В. принадлежит разработка метода применения схем радикалов для технологии решения задач среды радикалов.]
8. Чечкин А.В. Применение схем радикалов для описания проблемной области автоматизированного комплекса планирования и управления / А.В. Чечкин, А.Е. Евграфов, В.В. Рожков, В.И. Лощенков, М.В. Пирогов // Информационно-измерительные и управляющие системы – 2009. – Т. 7, № 3. – С. 5 – 11. [Пирогову М.В. принадлежат формальные описания проблемной области автоматизированного комплекса планирования и управления, разработанные с помощью схем радикалов]
9. Чечкин А.В. Метод синтеза алгоритма формирования программы работы целевой аппаратуры космического аппарата с помощью разрешения конфликтов в среде радикалов / А.В. Чечкин, В.И. Лощенков, А.Е. Евграфов, В.В. Рожков, М.В. Пирогов // Вестник ФГУП «НПО им. С.А. Лавочкина» – 2010. – № 3. – С. 42 – 47. [Пирогову М.В. принадлежат разработка метода.]
10. Чечкин А.В., Пирогов М.В. Методика обеспечения информационно - системной безопасности сложных систем / А.В. Чечкин, М.В. Пирогов // Нейрокомпьютеры: разработка, применение – 2008. – №7. – С. 11 – 17. [Пирогову М.В. принадлежит разработка метода использования формализма схем радикалов для методики обеспечения информационно-системной безопасности сложных систем.]
11. Чечкин А.В., Пирогов М.В. Интеллектуализация сложной системы как средство обеспечения ее информационно-системной безопасности / А.В. Чечкин, М.В. Пирогов // Фундаментальная и прикладная математика – 2009. – Т. 15, – вып. 3. – С. 225 – 239. [Пирогову М.В. принадлежит разработка метода применения формализма схем радикалов для интеллектуализации сложной системы.]
12. Чечкин А.В., Пирогов М.В. Обеспечение информационно-системной безопасности сложной системы на основе математического моделирования ее проблемной области нейрорадикалами / А.В. Чечкин, М.В. Пирогов // Международная конференция «Современные проблемы математики, механики и их приложений», посвященная 70-летию ректора МГУ академика В.А. Садовничего: Материалы конференции. 30 марта – 02 апреля 2009 года. Московский Государственный Университет им. М.В. Ломоносова – М.: МГУ им.

М.В. Ломоносова - 2009. – С. 404. [Пирогову М.В. принадлежит разработка метода реализация среды радикалов с помощью схем радикалов.]
13. Chechkin A.V., Pirogov M.V. Intellectualization of a complex system as a means of maintaining its information-system safety / A.V. Chechkin, M.V. Pirogov // Journal of Mathematical Sciences. – Springer New York, 2010. – Vol. 168, N 1. – P. 147 – 156. [Пирогову М.В. принадлежит разработка метода применения формализма схем радикалов для интеллектуализации сложной системы.]
14. Pirogov M.V., Chechkin A.V. Technology of task-solving in normalized radical media / Pirogov M.V., Chechkin A.V. // Journal of Mathematical Sciences. – Springer New York, 2010. – Vol. 168, N 1. – P. 133 – 146. [Пирогову М.В. принадлежит разработка метода применения схем радикалов для технологии решения задач среды радикалов.]