Символьная модель проблемной области сложной системы — основа интеллектуализации такой системы на примере АКПУ

А. В. Чечкин, А. Е. Евграфов, В. В. Рожков, М. В. Пирогов

1. Введение

Успехи современных сложных систем, программно-технических средств (ПТС) и масштабы их распространения общеизвестны. Однако из разных источников постоянно появляется информация, свидетельствующая о многочисленных и разнообразных проблемах используемых сложных систем. Существует ли проблема современных сложных систем, которую можно определить как главную и фундаментальную? Если да, то в чем она заключается? По нашему мнению, такая проблема существует. Это — Проблема Информационно-Системной Безопасности (ИСБ) сложных систем, сформулированная А. В. Чечкиным [1].

Типичным примером сложной системы является Автоматизированный Комплекс Программного Управления (АКПУ), который предназначен для планирования работы космического аппарата и целевой аппаратуры на заданном временном промежутке в соответствии с запросами потребителей и с учетом ограничений на работу бортовых и наземных средств. АКПУ — сложная система, включающая в себя математические, программные, информационные, технические, методические, лингвистические, организационные, правовые, эргономические и метрологические средства.

Нарушение ИСБ влечет за собой нарушение системности, с которым связаны многочисленные и разнообразные проблемы совре-

менных сложных систем. Обеспечение ИСБ требует развития общей теории ИСБ и обеспечение ее специальными средствами. Основу такой теории должны составить математические средства сложных систем, роль которых должна качественно измениться — стать главенствующей. Для обеспечения ИСБ необходим переход сложных систем на принципиально новый уровень — необходима интеллектуализация сложных систем.

В ходе работ над методикой обеспечения ИСБ сложной системы анализировались современные системы (подходы) в области информатизации. Эти средства рассматривались с точки зрения концепции CALS, объединяющей принципы и технологии информационной поддержки жизненного цикла продукции на всех его стадиях и использующей понятие интегрированной информационной среды. CALS охватывает: данные об изделии, WorkFlow-системы, различные аспекты деятельности предприятия. Основа CALS — современные общесистемные информационные технологии. Особое внимание при анализе было уделено: структурному программированию, логическому программированию, средствам ООП — объектно-ориентированного проектирования и программирования; CASE-технологиям; языку объектного моделирования UML; средствам разработки приложений Borland C++ Builder и Delphi (Borland Developer Studio 2006); СУБД (Microsoft SQL Server), методам численного моделирования. Возможности этих средств велики и разнообразны. Однако, как показывает практика, с точки зрения сложных систем — это средства, ориентированные, прежде всего, на фрагментарное применение, на достижение успеха при решении определенных узких классов задач. Остаются многочисленные и трудноразрешимые проблемы ИСБ сложных систем, возникающие при использовании имеющихся программно-технических средств (ПТС) решения различных задач этих систем, при развитии самих ПТС.

При используемом сегодня традиционном подходе к человеко-машинным системам велика роль текстовых документов на естественном языке и, соответственно, неизбежны многочисленные проблемы, связанные с этим. Информационная основа сложных систем — по-прежнему текстовая документация на естественном языке, что накладывает глубокий отпечаток на всю технологию работы с инфор-

мацией и самым существенным образом влияет на методические основы ПТС. Слишком часто работа идет не с математическими моделями, а с текстами на естественном языке. Основная причина такого положения дел, как представляется, заключается в отсутствии специализированных Математических Средств (МС) сложных систем, для которых центральным требованием является требование обеспечения ИСБ. В настоящей статье сделан очередной шаг в направлении создания таких средств. [7–8].

Что же должны обеспечить эти специализированные МС? Какими они должны быть? Какие аспекты являются ключевыми аспектами проблемы? МС сложных систем должны использовать общий подход к решению задач жизненного цикла, основывающийся на единой математической основе для представления и объединения всех описаний, для осуществления преобразований этих описаний, для ухода от возникающих информационно-системных конфликтов жизненного цикла систем. МС должны охватывать проблемную область, как в целом, так и в частностях, использоваться на всех иерархических уровнях, для всех «сечений» проблемной области.

Для сложных систем и, в том числе, для АКПУ, характерны как значительные достижения, так и многочисленные и разнообразные проблемы. Главная и фундаментальная проблема всех сложных систем — проблема их информационно-системной безопасности (ИСБ). Путь к ИСБ-системам — их интеллектуализация, основанная на специфическом символьном моделировании проблемной области этих систем.

Символьное моделирование основано на понятии РАДИКАЛА. Радикалы организованы в СРЕДУ РАДИКАЛОВ, сопровождающую сложную систему на всех этапах ее жизненного цикла и обеспечивающую решение как штатных, так и нештатных задач. Для описания и реализации такой среды языковыми конструкциями — схемами радикалов — разработан специальный язык RADICAL, с помощью которого в три этапа осуществляется нормализация среды радикалов. Нормализованная среда радикалов представляет собой символьную модель проблемной области сложной системы.

Первый этап нормализации состоит в разделении радикалов на два вида: уникумы и контейнеры. Уникумы соответствуют компо-

нентам проблемной области, а контейнеры отвечают за их свойства (связи). С помощью уникумов и контейнеров реализуется ЕДИНАЯ КООРДИНАТНАЯ СИСТЕМА КОНТЕЙНЕРОВ среды радикалов, создаются средства визуализации среды радикалов, основанные на геометрическом отображении схем и составляющие основу ее интерфейса. Уникумы и контейнеры образуют среду опорных радикалов.

Второй этап нормализации состоит в УЛЬТРАОСНАЩЕНИИ среды радикалов. К опорной среде добавляется ультра среда, состоящая из ультрарадикалов — продукций базы знаний, и терминальная среда — радикалы-исполнители и радикалы-датчики.

Третий этап нормализации состоит в ОРГАНИЗАЦИИ среды радикалов по принципам нормализации — принцип первого звена среды радикалов, принцип первого контейнера среды радикалов и т. д. Организуется библиотека стандартных радикалов.

После нормализации среды радикалов создаются средства ее АКТИВАЦИИ и РЕГУЛЯЦИИ — средства выделения схем, схемы-запросы и схемы-ответы, средства анализа и синтеза схем. Для поиска ответов на запросы предназначены специальные схемы — АКТИВАТОРЫ. Применяются схемы, представляющие задачи, методы их решения, а также схемы оценки и классификации других схем.

В течение жизненного цикла сложной системы контейнеры, характеризующие уникумы, могут преобразовываться, что может приводить к нежелательным КОНФЛИКТАМ в смысле знаний, представленных в ультра среде радикалов. Возникает проблема разрешения конфликтов (ухода от конфликтов) в целях обеспечения ИСБ. Это делается с помощью управляющих воздействий, применяемых к управляемым (изменяемым) контейнерам со стороны специализированных схем — РЕГУЛЯТОРОВ. Для решения проблемы конфликтов выделяются конфликтующие (в смысле ультраоснащения) пары контейнеров среды радикалов. Используются приближения сложных контейнеров более простыми контейнерами. Рассматриваются как штатные, так и нештатные преобразования контейнеров. Предлагается двунаправленный метод ухода от конфликтов в среде радикалов и др.

2. Информационно-системная безопасность сложных систем

Понятие Информационно-Системной Безопасности (ИСБ) сложной системы [1] является главной характеристикой такой системы. ИСБ включает в себя две стороны — информационную безопасность и системную безопасность сложной системы.

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ является особого рода функциональной устойчивостью сложной системы, когда обеспечивается безусловное решение задач жизненного цикла системы вне зависимости от формы (от языка) представления входной информации и от полноты этой информации. Обеспечение информационной безопасности сложной системы реализуется, главным образом, путем постоянного использования символьного моделирования проблемной области системы и методов логического вывода. В терминах математической информатики — это означает переход от ОПЕРАТОРОВ к УЛЬТРАОПЕРАТОРАМ [2, 3].

СИСТЕМНАЯ БЕЗОПАСНОСТЬ сложной системы — это безусловное сохранение ЯДРА СИСТЕМЫ при решении любой частной задачи жизненного цикла. Другими словами, сохранение системообразующих (жизненных) составляющих системы и тех связей, которые обеспечивают полноценное функционирование сложной системы, ее системную целостность. Системная безопасность сложной системы — это постоянный учет и устранение конфликтов между ее составляющими и их связями, которые появляются при решении задач жизненного цикла системы, а так же конфликтов между самой сложной системой и внешними к ней системами. Системная безопасность требует реализации СИСТЕМНОГО ПОДХОДА ко всей проблемной области сложной системы [4, 5].

Разрушительные последствия нарушения функционирования сложных систем, вплоть до экологических катастроф, человеческий фактор в сложных системах, приводящий зачастую к непредсказуемым последствиям для такой системы, — это и многое другое говорят, что чем сложнее система, тем важнее для нее проблема обеспечения ИСБ в течение ее жизненного цикла.

Основной подход к обеспечению ИСБ сложной системы — это ИНТЕЛЛЕКТУАЛИЗАЦИЯ такой системы. Интеллектуализация любой системы подразумевает символьное представление ее проблемной области и на основе этого представления оснащение отдельных составляющих системы и ее подсистем элементами искусственного интеллекта. Такое оснащение должно обеспечивать постоянную ситуационную адаптацию сложной системы к изменяющимся внутренним и внешним условиям, возможность проводить диагностику, контроль, анализ и синтез отдельных составляющих системы и функционирования системы в целом с учетом последствий этого функционирования с целью обеспечения ИСБ на протяжении всего жизненного цикла системы.

Таким образом, интеллектуализация сложной системы означает создание интеллектуальной НАДСТРОЙКИ, своего рода МОЗГА, сложной системы, включающей:

1) создание символьной МОДЕЛИ всей проблемной области сложной системы, то есть, КАРТИНЫ МИРА сложной системы, включающей саму систему, ее окружение и отражающей все проблемы жизненного цикла системы;

2) создание специального информационно-программного ОСНАЩЕНИЯ этой модели (картины мира) средствами обеспечения ИСБ сложной системы. Целью такого оснащения является обеспечение ИСБ сложной системы при решении задач ее жизненного цикла путем моделирования, анализа и синтеза проблемной области системы, включая окружение системы, ее отдельных составляющих, ее характеристик, адаптацию и прогнозирование поведения системы, и многое другое.

Термином интеллектуализация подчеркивается свойство такой надстройки развиваться и расширять круг решаемых ею ШТАТНЫХ задач ИСБ за счет обучения решению некоторых НЕШТАТНЫХ для нее задач. Чем больше разнообразных и более трудных нештатных задач сможет решить такая интеллектуальная система, тем более она интеллектуальная.

3. Радикалы и символьное моделирование проблемной области сложной системы с помощью нейрорадикалов

Понятие РАДИКАЛА является главным понятием математической информатики [2] и, по-видимому, всей дискретной математики [3]. 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 является СРЕДОЙ РАДИКАЛОВ. Вопросами активирования среды радикалов занимаются системы, которые называются АКТИВАТОРЫ. Система всех активных радикалов среды радикалов в данный момент времени образует, так называемый, СИСТЕМОКВАНТ, который определяет квант поведения среды в этот момент. За разрешение конфликтов в среде радикалов, то есть за обеспечение ИСБ среды радикалов, отвечают специализированные системы, РЕГУЛЯТОРЫ.

Типичным примером среды радикалов является набор лего-конструктора. Из элементов такого набора можно собирать разнообразные конструкции. Активатором в этом случае является робот или человек, собирающий конкретную конструкцию. Тот лего-элемент, который используется в данной конструкции, является активным, а тот, который нет, является пассивным. Созданная в результате сборки конструкция будет системоквантом. Регулятором в этом случае является система обеспечения постоянного наличия в наборе всех положенных элементов конструктора.

Другим примером среды радикалов является система управляющих параметров любого объекта управления. Такой объект имеет различные состояния (фазовые координаты) в зависимости от выбранных параметров управления. Наборы всех возможных значений параметров управления образуют среду радикалов, система управления объектом является активатором, а реализованные (выбранные) в данный момент значения управляющих параметров образуют системоквант в этой среде радикалов. Регулятором здесь является систе-

ма обеспечения работоспособности (реализуемости) всех возможных управляющих параметров.

Третьим примером среды радикалов является любой музыкальный инструмент (система источников звуков). Музыкант является активатором такой среды радикалов. Звучащая музыка является результатом функционирования соответствующего системокванта в данный момент. Регулятором здесь является настройщик инструмента.

Другими примерами радикалов являются:

1) Точки непустого множества являются радикалами, если их выбирают для оценивания по некоторому критерию или, если они участвуют в алгебраических операциях в качестве операндов и т. д. Активатором здесь является система выбора точек, системоквантом — набор выбранных точек. Регулятор обеспечивает эффективное задание исходного множества.
2) Буквы алфавита — радикалы. Они предназначены для составления из них слов (системоквантов). Та буква, которая используется в данном слове, является активной, а та, которая нет, является пассивной. Активатором является система составления слова. Регулятором является система поддержания целостности алфавита.
3) Слова лексики любого языка — радикалы. Они предназначены для составления из них текстов (системоквантов). Здесь активатор — это система генерирования текста. Регулятором является система, отвечающая за грамматическую правильность слов в тексте и полноту словаря.
4) Тексты (записи) любой базы данных — радикалы. Они предназначены для составления ответов (системоквантов) на запросы к базе данных. Активатором в этом случае выступает компьютерная программа (система) формирования ответа (решения задачи). Регулятором является система поддержки базы данных в рабочем состоянии.
5) Компьютерные программы (алгоритмы, вычислительные системы, автоматы), предназначенные для обработки текстов (информации) являются радикалами, если имеется возможность

использовать их или нет (запускать или останавливать ту или иную программу). Операционная (диспетчерская) система является активатором такой среды радикалов, то есть средств обработки текстов (решения задач). Задействованные средства обработки текста образуют системоквант. Регулятором здесь является система поддержки программных средств в рабочем состоянии, защиты их от несанкционированного доступа и т. п.

6) Функциональные системы живого существа или его навыки, в смысле П. К. Анохина, образуют среду радикалов, так как имеется возможность их активировать или не активировать в зависимости от внешней ситуации. Активатором живого существа является его мозг. Системоквант в этом случае определяет поведение существа. Регулятором является система поддержания всех функциональных систем в норме.

7) Вооруженные силы являются средой радикалами, так как военные подразделения можно активировать приказом на боевые действия или вывести из боевых действий. Активатором выступает здесь командующий военной операцией. При этом боевое соединение, выполняющее приказ, является системоквантом. Регулятором в этом случае выступает орган обеспечения штатного укомплектования военных подразделений.

Понятие РАДИКАЛА позволяет взглянуть на всю проблемную область сложной системы, то есть на сложную систему и её окружение с учетом жизненного цикла системы, как на среду радикалов. Радикалами такой среды являются составляющие сложной системы и ее окружения, их связи, задачи жизненного цикла, средства и методы решения таких задач, специалисты, нормативные документы и многое другое. При таком подходе создаваемая Интеллектуальная Система (ИС) сложной системы должна базироваться на символьной модели всей проблемной области сложной системы, что бы принимать такие решения (определять такие системокванты поведения сложной системы), которые обеспечивают ИСБ системы на протяжении всего ее жизненного цикла.

Интеллектуальную систему сложной системы будем строить из двух частей, рабочей и активирующей, рис. 1. Рабочая подсистема

будет символьным представлением в ИС радикалов проблемной области сложной системы, то есть будет МОДЕЛЬЮ ПРОБЛЕМНОЙ ОБЛАСТИ (картиной мира) сложной системы. Активирующая подсистема должна содержать средства активации, регуляции, анализа и синтеза рабочей области. Радикалы-символы в рабочей области будем называть НЕЙРОРАДИКАЛАМИ. Таким названием мы хотим подчеркнуть принципиальную ассоциацию строения рабочей области ИС со строением элементарного сенсориума естественного мозга из нейронов [6].

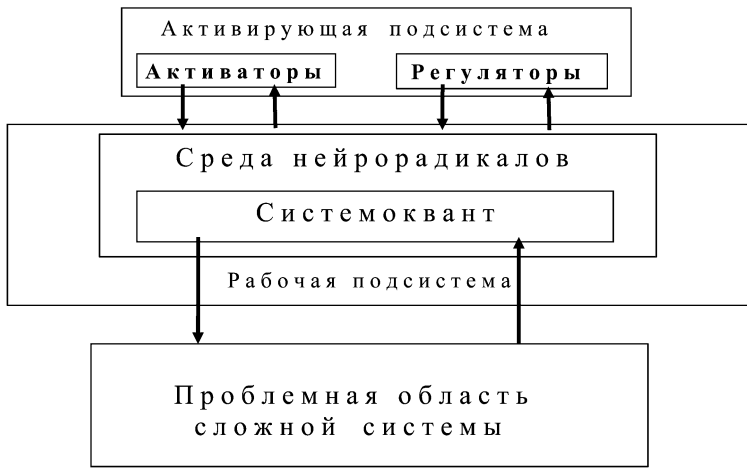

Рис. 1. Интеллектуальная система проблемной области сложной системы.

Тем самым рабочая подсистема ИС является символьной МОДЕЛЬЮ проблемной области сложной системы в форме СРЕДЫ нейрорадикалов. Активирующая подсистема отвечает за ИСБ-решение задач жизненного цикла системы, постоянно разрешает конфликты в среде нейрорадикалов. Используя модель проблемной области, активирующая подсистема должна создавать в рабочей подсистеме системоквант, который определяет ИСБ-поведение сложной системы.

4. Структура среды нейрорадикалов

Так как нейрорадикалы являются символьным представлением в ИС радикалов проблемной области сложной системы, то их должно быть не меньше, чем всех таких радикалов. На самом деле, нейрорадикалов в ИС будет намного больше, чем радикалов проблемной области. Это связано с тем, что в ИС мы должны отразить не только сами радикалы, как объекты (сущности), но и разнообразные их свойства, типы, классы, связи и многое другое.

Воспользуемся формализмом математической информатики [2, 3]. Сопоставим радикалам проблемной области сложной системы ПРЕДМЕТНЫЕ СИМВОЛЫ (ТЕРМЫ) логики предикатов, а свойствам, типам, классам этих радикалов — ПРЕДИКАТНЫЕ СИМВОЛЫ. Для примера, рассмотрим некоторые объекты моделирования (радикалы) проблемной области, которые обозначим термами $x_1, x_2, x_3, \ldots, x_n$. Сопоставим им в рабочей подсистеме ИС нейрорадикалы (точки) и обозначим их теми же символами, рис. 2. Пусть эти объекты обладают свойствами, которые описываются одноместными предикатами $A_1(x_1), A_1(x_2), \ldots, C_3(x_n)$. Каждому предикатному символу $A_1, \ldots, C_3$ в рабочей области ИС сопоставим нейрорадикалы с теми же обозначениями. Будем их интерпретировать как понятия (подмножества), описывающие некоторые свойства объектов предметной области сложной системы.

Факт принадлежности точки множеству $x \in A$, будем изображать графически стрелкой от вершины с символом множества к вершине с символом точки $A \to x$, рис. 2. Если одно множество является подмножеством другого $A \subseteq P$, то графически это будем изображать стрелкой от символа большего множества к символу меньшего $P \to A$, рис. 2.

В результате такого представления проблемной области сложной системы в рабочей подсистеме ее ИС появится символьная модель проблемной области, в которой стрелкам соответствуют ссылки. Среда нейрорадикалов является своего рода семантической сетью разнообразных понятий, например, в форме гипертекста. При этом нейрорадикалы-термы $x_1, \ldots, x_n$ образуют КЛАССИЧЕСКИЕ МАТЕМАТИЧЕСКИЕ МОДЕЛИ объектов проблемной области, а

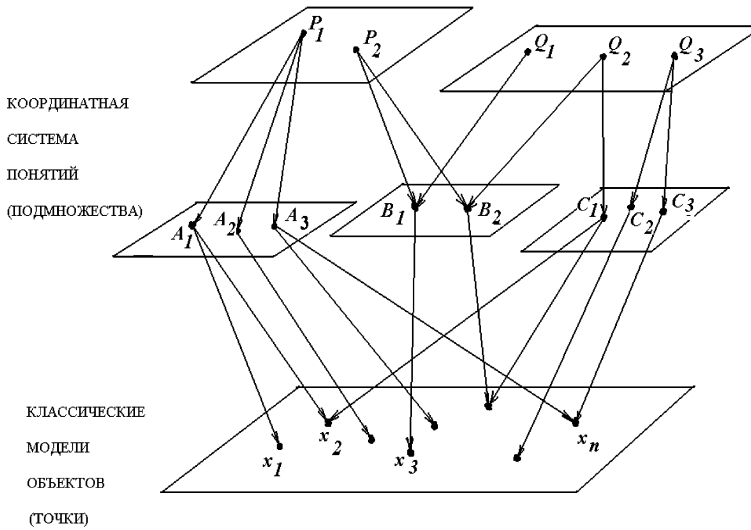

Рис. 2. Структура среды нейрорадикалов как семантическая сеть.

нейрорадикалы-понятия, соединенные в сеть, являются своего рода КООРДИНАТНОЙ ИЕРАРХИЧЕСКОЙ СИСТЕМОЙ ПОНЯТИЙ. На рис. 2 атомарные понятия $A_1, \ldots, C_3$ являются более конкретными, чем понятия неатомарные.

Построенная структура нейрорадикалов реализует идеи топологической координатизации множеств и топологической фильтрации множеств [2, 3]. Такая структура позволяет легко ориентироваться в ней, быстро находить конкретные нейрорадикалы по известной информации о них, получать требуемую информацию о любом нейрорадикале и многое другое. Каждое понятие является элементом атомарной шкалы соответствующей решетки понятий (подмножеств). Перекрестные связи таких понятий реализуют идею произведения решеток понятий [2, 3] и, тем самым, идею параллельных (дублирующих) информационных каналов подобных тем, которые есть в естественном интеллекте (мозге) [6] в виде зрительного канала, слухового, обонятельного, вкусового, осязательного каналов.

Так в построенной среде нейрорадикалов по информации (пути в графе) $P_1 \to A_1 \to x_1$ (рис. 2) легко найти точку (модель объекта) x_1.

Реализуя конъюнкцию двух сообщений $P_1 \to A_1 \to x$ и $Q_2 \to C_1 \to x$ о неизвестной точке x, можно найти эту точку $x = x_2$. Точку x_3 можно найти по разным информациям $P_2 \to B_1 \to x_3$ или $Q_1 \to B_1 \to x_3$. Наконец, про точку x_n можно узнать любую информацию $P_1 \to A_3 \to x_n$ или $Q_3 \to C_3 \to x_n$.

Отметим, что в предлагаемой символьной модели вместо традиционной записи предиката $P_1(x_1)$ будем иметь запись цепочки $P_1 \to A_1 \to x_1$.

Наряду с топологической координатизацией, в рабочей подсистеме ИС реализуются функциональные связи между нейрорадикалами. Например, алгебраические операции над точками или над понятиями. В результате среда нейрорадикалов реализует идею алгебраической системы в смысле А. И. Мальцева. При этом сами нейрорадикалы представляют собой только образующие соответствующих алгебр, а другие необходимые элементы алгебр получаются из образующих по ходу решения задач средствами активирующей подсистемы ИС.

Наряду с одноместными предикатами в рабочей области ИС представлены и различные многоместные предикаты, описывающие проблемную область сложной системы. В целом рабочая подсистема является распределенной базой данных проблемной области сложной системы.

Кроме контейнеров в рабочей области ИС вводятся УЛЬТРАКОНТЕЙНЕРЫ, отражающие накопленные знания о свойствах и связях в проблемной области сложной системы. Всякий ультраконтейнер вводится как оснащение для своего опорного контейнера рабочей подсистемы и представляет собой продукцию. Например, для многоместного контейнера, описывающего систему, имеющую свои подсистемы, оснащением будет ультраконтейнер, описывающий условия на температурные режимы подсистем, условия на их габариты, на их материал и т. п. Все такие ультраконтейнеры образуют ультрасреду рабочей подсистемы ИС. Ультрасреда является распределенной базой знаний о проблемной области сложной системы и является, так называемым УЛЬТРАОСНАЩЕНИЕМ опорной среды. Такое ультраоснащение реализует идею ультраоператоров в математической информатике [2, 3].

Перейдем к активирующей подсистеме ИС, где реализуются средства решения разнообразных задач по обеспечению ИСБ сложной системы с использованием среды нейрорадикалов. Главное в такой подсистеме принадлежит АКТИВАТОРАМ и РЕГУЛЯТОРАМ. Активаторы — это специализированные радикалы ИС, нацеленные каждый на решение своего класса задач. Работа активаторов опирается на использование распределенной базы данных и знаний о проблемной области в форме среды опорных радикалов и ультрарадикалов. Регуляторы — это радикалы, отвечающие за обеспечение целостности и безконфликтности среды нейрорадикалов. Кроме этого в активирующей подсистеме широко используются командные (окрашивающие) радикалы, которые вычленяют в среде нейрорадикалов отдельные схемы и являются средством навигации в среде нейрорадикалов, средством сохранения опыта решения задач в прошлом и многое другое. Далее в статье вместо термина нейрорадикал будем говорить радикал.

5. Язык RADICAL и схемы радикалов

Разработанный язык радикалов — язык RADICAL — является основой МС сложных систем и, соответственно, математического обеспечения. Язык радикалов нацелен на описание объектов проблемной области и решение задач жизненного цикла сложных систем, включая решение служебных задач. Описание объектов осуществляется с учетом их связей в форме схем радикалов. Рассмотрим языковые средства, которые позволят представлять радикалы и строить из них необходимые структуры.

АЛФАВИТ языка радикалов — множество, состоящее из следующих символов: пробел, строчные и заглавные буквы латинского и русского алфавитов, цифры от 0 до 9 и символы $+ - * / = ( ) \{ \} < > \_ ; ? \rightarrow \leftarrow [ ] \ldots$.

Из символов алфавита строятся неиндексированные и индексированные имена радикалов, например (соответственно) $Name9$ и $Name[9]$, $Name[*]$ (символ '$*$' используется для обозначения пропущенных символов).

Слово вида $Ni \rightarrow Nj$ называется звеном, здесь Ni, Nj — имена радикалов. Слово вида $Ni \leftarrow Nj$ называется обратным звеном, соот-

ветствующим звену $Ni \to Nj$. Звено является базовой конструкцией (базисом) языка радикалов.

Из звеньев строятся цепочки радикалов. Например, $N0 \to N1 \to \ldots \to Nm-1 \to Nm$; — m-звенная цепочка радикалов. Любое множество цепочек радикалов называется схемой радикалов. Над схемами радикалов определены операции объединения, пересечения, разности.

Радикалы могут образовывать структуры, которые мы будем называть ветвлениями и схождениями. (Ветвления и схождения также являются схемами радикалов.) Например, следующая схема является ветвлением: $\{Name30 \to Name31 \to Name32;\ Name30 \to Name41 \to Name42;\}$. Другая форма записи ветвления: $Name30 \to \{Name31 \to Name32;\ Name41 \to Name42;\}$. Для удобства допускается идентификация цепочек ветвления с использованием служебных радикалов вида $d[*]$ ($d[*]SmthName$). Таким образом, для ветвления можем иметь, например, следующую запись: $Name30 \to \{d[1] \to Name31 \to Name32;\ d[2] \to Name41 \to Name42;\}$.

Рассмотрим пример схождения к радикалу $Name53$: $\{Name51 \to Name52 \to Name53;\ Name61 \to Name62 \to Name53;\}$. Другая форма записи схождения: $\{Name51 \to Name52,\ Name61 \to Name62\} \to Name53$; или с использованием служебных радикалов вида $d[*]$ ($d[*]SmthName$) $\{d[1] \to Name51 \to Name52,\ d[2] \to Name61 \to Name62\} \to Name53;$.

Допускается также запись ветвления и схождения с использованием обратных цепочек, например: $Name53 \leftarrow \{Name62 \leftarrow Name61;\ Name52 \leftarrow Name51;\}$.

Приведем пример вложения схем радикалов. Множество, состоящее из двух цепочек радикалов $\{Name81 \to Name82 \to Name83 \to Name84;\ Name81 \to Name91 \to Name83 \to Name84;\}$, является схемой радикалов. Эту схему можно записать в виде вложения схем: $\{Name81 \to \{Name82,\ Name91\} \to Name83 \to Name84;\}$. Вложение — результат замены радикала $Name81$ второй цепочки схемой $Name81 \to Name82 \to Name83;$.

Допускается нумерация фигурных скобок, то есть уровня вложенности пар скобок с использованием чисел из множества $\{0; 1; 2; \ldots\}$. Приведем соответствующий пример: $Name0 \to$

$\{0\ Name1 \to Name11 \to \{1\ Name111 \to Name1110;\ Name112 \to Name1120;\}1\ Name2 \to Name21 \to \{1\ Name211 \to Name2110;\ Name212 \to Name2120;\}1\ \}0$.

Удобно рассматривать геометрические отображения схем радикалов. На рис. 3 представлен пример отображения ветвления $N30 \to \{d[1] \to N31 \to N32;\ d[2] \to N41 \to N42;\}$ на плоскость. 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.

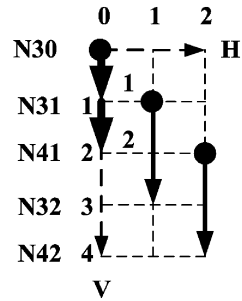

Рис. 3. Пример геометрического отображения ветвления $N30$.

Решение задач жизненного цикла сложной системы осуществляется с помощью активирования среды радикалов с использованием средств ухода от конфликтов. Соответствующие средства будут рассмотрены ниже.

6. Нормализация среды радикалов

Информационно-системная безопасность сложной системы обеспечивается в первую очередь нормализацией среды радикалов проблемной области. Нормализация проводится в три этапа.

Первый этап нормализации состоит в разделении радикалов на два вида: уникумы и контейнеры.

Имя уникума 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 сложной системы, составляющая некоторой проблемной области. С помощью второго индекса идентифицируется экземпляр уникума определенного типа. С помощью третьего индекса идентифицируется модификация уникума. Допускается сокращенная запись вида $uSmthUnicum$;, uSU;.

Приведем примеры некоторых уникумов: $u(0); u(1); u(-1); \ldots$ — целые числа; $u(0,0\ldots00); u(0,0\ldots01); u(-0,0\ldots01); \ldots$ — конечные десятичные дроби; $uFALSE$; $uTRUE$; — константы двузначной логики.

С помощью понятия контейнера реализуется идея топологической фильтрации уникумов в среде радикалов. Если уникумы соответствуют компонентам проблемной области сложной системы, то контейнеры отвечают за их свойства. Контейнер может содержать только те уникумы (компоненты системы), которые обладают выделенным свойством. В математической информатике контейнеру соответствует многоместное отношение или понятие, или многоместный предикатный символ.

Имя контейнера 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. Контейнер — способ задания связи определенного типа между радикалами.

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) (см. ниже). КСК позволяет осуществить структуризацию семейства радикалов, эффективно

решать проблему выбора радикала, используя запросы и радикал-активатор (см. ниже).

С точки зрения математической информатики [2, 3], всякий контейнер, соответствующий предикатному символу, интерпретируется как подмножество всех возможных значений переменной (уникумов), на которых истинен одноместный предикат. Между такими контейнерами определено бинарное отношение частичного порядка по включению, которое на языке радикалов обозначается стрелкой. Кроме этого, в среде радикалов для таких контейнеров определены логические операции: дизъюнкция (объединение), конъюнкция (пересечение) и отрицание (дополнение). Это обеспечивается путем ультраоснащения среды радикалов (см. ниже). Тем самым, все одноместные контейнеры, с учетом логических операций, образуют булеву решетку подмножеств сущностей (уникумов) проблемной области сложной системы. В этом состоит идея координатизации множеств и перехода от множеств к ультрамножествам [2, 3], которая положена в основу КСК.

Благодаря средствам ухода от конфликтов (они также будут рассмотрены ниже), КСК — самоконтролируемая, самокорректируемая и развивающаяся (изменяющаяся) система контейнеров. С помощью КСК и на основе механизма разрешения конфликтов (см. ниже) автоматически обеспечиваются многие аспекты ИСБ сложной системы, целесообразная реакция системы на внешние воздействия, автоматическое целесообразное управление внутренними процессами сложной системы и ее воздействием на внешнюю среду.

С помощью КСК, осуществляется координатизация всех используемых математических моделей. Все они представляются в нормализованном виде, уточняются, координируются относительно этой системы с фиксацией общих контейнеров, используемых при разрешении конфликтных ситуаций.

Контейнеры часто являются радикалами ветвления среды радикалов. Соответствующие цепочки ветвления (направления, доступные в контейнерах) будем идентифицировать с помощью служебных радикалов, начинающихся с символа ‘d’ (от слова «*direct*») и имеющих вид $d[1:*][2:*]SmthDirect$ (допускается сокращенная запись: $d[*]SD$, dSD), 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.

Например, одноместный контейнер трех целых чисел

$$\underline{cIntMany} \rightarrow \{d[1]UnitOfMany \rightarrow u(0);$$
$$d[2]UnitOfMany \rightarrow u(1);\ d[3]UnitOfMany \rightarrow u(-1);\}.$$

(Допускается сокращенная запись UofM вместо UnitOfMany.)

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 [2, 3].

Приведем пример еще одного контейнера. Двуместный контейнер вхождения $\underline{cWholeAndPart}$, используется для описания непосредственного вхождения одних составляющих системы в другие. Пусть в составляющую ua входит составляющая ub, тогда имеем:

$$cWholeAndPart \rightarrow \{d[1]Whole \rightarrow ua;\ \ d[2]Part \rightarrow ub;\}.$$

Геометрическое отображение этой схемы представлено на рис. 4.

Отметим, что контейнеры соответствуют предикатным символам логики предикатов, а уникумы соответствуют термам логики предикатов, в том числе переменным (имена соответствующих радикалов имеют вид $uvar*$) и значениям функций. Для функциональных символов используются контейнеры вида $cf*$. Для неопределенных значений используется специальный уникум $uNULL$. Для логических операций используются контейнеры cOR, $cAND$ и $cNOT$. Для кванторов используются специальные контейнеры — либо $cqvEXIST$, либо $cqvANY$. Соответствующие примеры приведены на рисунке 5.

На первом этапе нормализации на основе КСК создаются также средства визуализации среды радикалов, направленные на примене-

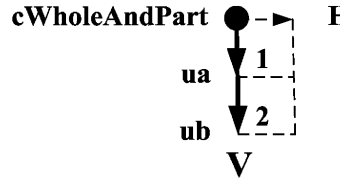

Рис. 4. Двухместный контейнер вхождения $cWholeAndPart$.

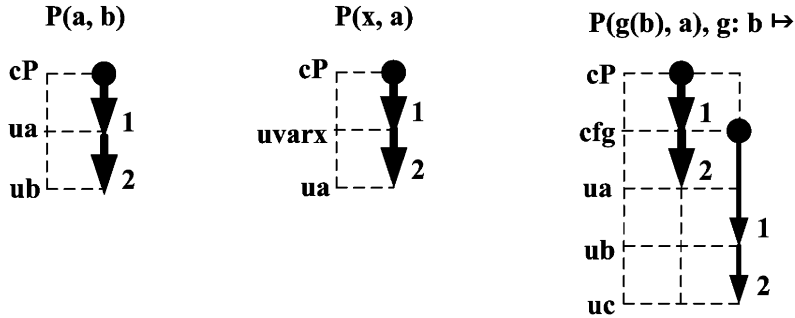

Рис. 5. Реализация в среде радикалов предикатов, переменных и употреблений функций.

ние средств компьютерной графики. Визуализация основана на геометрическом отображении схем радикалов. Визуализация составляет основу интерфейса среды радикалов, предназначенного для обеспечения визуального контроля и интерактивного взаимодействия человека со средой радикалов. Введение КСК позволило продвинуться вперед в геометризации языка радикалов и, соответственно, в геометризации среды радикалов, которая, с этой точки зрения, является системой с самоконтролируемой геометрией.

Итак, мы ввели класс основную часть радикалов — уникумов и контейнеров, которые назовем опорными радикалами, образующими среду опорных радикалов (опорную среду радикалов).

Второй этап нормализации состоит в ультраоснащении среды радикалов. В среде радикалов (рабочей подсистеме ИС) вводятся три взаимосвязанных части: опорная среда, ультрасреда и терминальная среда. Опорная среда образуется из опорных радикалов — это уникум-

мы и контейнеры возможных и присутствующих сущностей проблемной области, например, составляющих сложной системы. Ультрасреда образуется из ультрарадикалов (см. ниже). Это системы анализа и синтеза, предназначенные для ИСБ-решения задач жизненного цикла системы. Ультрарадикалы — это продукции базы знаний проблемной области. Терминальная среда образуется из радикалов-исполнителей и радикалов-датчиков, осуществляющих связь между опорными радикалами и ультра радикалами. Ультра среда вместе с терминальной средой определяют, так называемое, ультраоснащение среды радикалов, предназначенное для ИСБ-решения задач жизненного цикла сложной системы, выявления и снятия конфликтов и системных нарушений целостности системы.

Для построения ультрасред будем использовать схемы специального вида, определяемые ультра контейнерами двух типов 1 и 2.

Ультра контейнеры типа 1 используются при поиске схем в среде опорных радикалов, а также для добавления новых схем к этой среде. Пусть имеем схему опорных радикалов $FirstLink \rightarrow cAllContainers \rightarrow \{\dots\}$. Здесь звено $FirstLink$ — «начало» среды радикалов. Ультра контейнер типа 1 — это радикал с именем вида $cUltra1[1:*][2:*]SmthName$ (допускается сокращенная запись $cU1SmthName$), определяющий схему-продукцию. При этом схема-продукция является ветвлением. Одна ветвь — это схема-заключение вида $d[1]Conclusion \rightarrow \dots$; другая ветвь — это пустая схема-посылка вида $d[2]Premise \rightarrow$;. Схема-продукция имеет следующий вид:

$$cUltra1[1:*][2:*]SmthName \rightarrow$$
$$\{d[1]Conclusion \rightarrow cSmthContainer \rightarrow \dots;\ \ d[2]Premis \rightarrow;\}.$$

(Посылка обязательно оканчивается пустой схемой.) Первый индекс в $cUltra1[1:*][2:*]SmthName$ — это тип ультра контейнера, второй индекс — экземпляр ультра контейнера. Схемы, доступные по направлению $d[1]Conclusion$, могут содержать как уникумы, так и звенья-переменные вида $uVarSmthName =$;.

Ультра среда определяется множеством ультра контейнеров. Пусть контейнер $cAllUltraContainers$ объединяет ультра контейнеры, используемые для описания некоторой ультра среды. Опорная

среда, начинающаяся с радикала $cAllContainers$, и ультра среда, начинающаяся с радикала $cAllUltraContainers$, называются связанными средами, если существует схема вида: $FirstLink \to \{d[*] \to cAllContainers;\ d[*] \to cAllUltraContainers;\}$. Полученную среду (схему) $FirstLink \to \ldots$ будем называть ультраоснащенной средой (схемой).

Ультра контейнеры типа 2 имеют, в отличие от ультра контейнеров типа 1, непустую посылку. Ультра контейнер типа 2 — это схема-продукция с непустой посылкой, имеющая следующий вид:

$$\underline{cUltra2[1:*][2:*]SmthName} \to \{1\underline{d[1]Conclusion} \to \ldots;$$
$$\underline{d[2]Premise} \to \underline{cAND} \to \{2d[1]And \to \ldots;\ \ldots d[*]And \to \ldots;\}2\}1.$$

Схемы, связываемые ультра контейнером типа 2, могут содержать как уникумы, так и звенья-переменные вида $uVarSmthName =;$.

Итак, мы рассмотрели ультра контейнеры и примеры ультраоснащенных сред радикалов. Отметим, что ультраоснащенные среды радикалов реализуют фразовую форму логики предикатов.

С точки зрения математической информатики [1, 2], ультра контейнеры типа 2, собранные в блоки (наборы), определяют ультраоператоры. Такие ультраоператоры обеспечивают отображения одних уникумов в другие в форме отображения информации об уникуме-аргументе в информацию об уникуме-значении. Именно ультраоператоры обеспечивают информационную устойчивость (безопасность) сложной системы.

Третий этап нормализации состоит в эффективной организации среды радикалов по специальным принципам нормализации. Это, например, принцип первого звена среды радикалов, принцип первого контейнера среды радикалов и т. д. Соблюдение этих принципов обеспечивает корректное рассмотрение в СОКР иерархических структур в проблемной области сложной системы, параметров составляющих системы, изменений проблемной области во времени, вариантность при построении отдельных составляющих и всей системы в целом, взаимодействие между составляющими системы и др. На третьем этапе нормализации организуется библиотека стандартных радикалов.

7. Активация среды радикалов

После нормализации требуется создать средства активации среды радикалов в форме активаторов и механизм разрешения конфликтов в среде радикалов.

Рассмотрим вопросы активации в среде опорных радикалов, когда некоторые радикалы используемых схем могут быть активированы, и множество таких активированных радикалов может изменяться во времени.

Помеченные схемы и навигация 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 цепочек радикалов, объединенных контейнером вида $cColor$, имеющим следующий вид:

$$cColor \to \{d[1]SelectedColor \to u(*);\\ d[2]Chain \to \ldots; \ldots; d[*]Chain \to \ldots;\}.$$

Здесь $u(*)$ — это уникумы $u(1), u(2), \ldots$, соответствующие натуральным числам, с помощью которых определяется «цвет» выделения. Цепочки помеченной схемы доступны в контейнере $cColor$ по направлениям $d[2]Chain \to \ldots; \ldots; d[*]Chain \to \ldots;$. Контейнеры $cColor$ — системообразующие контейнеры, с их помощью осуществляется выделение «слоев», построение иерархических структур в среде радикалов. 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.

Рассмотрим средства активации помеченных схем с целью использования их при решении различных задач анализа и синтеза.

Запросы. 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 аналогичными им конкретными схемами среды.

Определим запросы типа 1. Пусть $FirstLink$ — начальный радикал среды радикалов, описывающей проблемную область. Пусть звено $FirstLink$ связано с контейнером $cAllContainers$, объединяющим используемые для описания проблемной области контейнеры вида $cSmthContainer$: $FirstLink \to cAllContainers \to \{d[1] \to c[2:1]SmthContainer; \dots d[*] \to c[2:*2]SmthContainer;\}$. Запрос типа 1 — это схема вида: $?[1]Question[i] \to cSmthContainer$; (сокращенно $?Q \to \dots;$). Здесь: 1 — тип запроса; i — порядковый номер запроса в последовательности запросов. Смысл запроса типа 1 следующий: выяснить, существует ли в среде радикалов схема, завершающая цепочку $?[1]Question[i] \to \dots;$.

Для представления ответа на запрос типа 1 будем использовать специализированный контейнер $cAnswer$, объединяющий краткий ($dBrief \to \dots$) и полный ($dFull \to \dots$) ответы. Краткий ответ (да, нет) фиксирует факт подтверждения (опровержения) существования в среде радикалов $FirstLink \to \dots$ 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:

$$cAnswer \to \{d[1]Brief \to \dots; d[2]Full \to \dots;\}.$$

Для завершения цепочки $d[1]Brief \to \dots;$ используется либо уникум uNO, либо звено $uYES$. В случае неудачи в результате поиска ответа на запрос будем иметь $cAnswer \to \{d[1]Brief \to uNO; d[2]Full \to;\}$. В случае успеха $cAnswer \to \{d[1]Brief \to uYES; d[2]Full \to cSmthContainer \to \dots;\}$

Визуализация среды радикалов. В дальнейшем мы будем использовать, как уже делали ранее, визуализацию в форме геометрического отображения среды радикалов, которое назовем типовым. Приведем неформальное описание этого отображения.

Введем на плоскости прямоугольную систему координат, в начало которой поместим радикал $FirstLink$ (FL). Оси координат HFL_cView* и VFL_cView* направим, соответственно, вправо и вертикально вниз. Здесь $cView*$ — имена контейнеров представления. С их помощью будем фиксировать радикалы схем, доступные для наблюдения. Уникумы, контейнеры, переменные и запросы будем изображать точками, расположенными на вертикальной оси. Контейнеры (связи) будем определять с помощью вертикально направленных векторов с указанием соответствующих направлений $(1, 2, \ldots)$ в схемах (направления идентифицируются с помощью специальных звеньев вида $d[1]SmthDirect, d[2]SmthDirect, \ldots$). В ряде случаев с целью упрощения записи, обозначения осей координат, направлений, указание радикалов, недоступных для наблюдения, будем опускать.

Таким образом, часть (срез) среды радикалов изображается 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. В случае успешного связывания запрос типа 2, как и запрос типа 1, приводит к нахождению схем (схемы) радикалов, удовлетворяющих этому запросу. Запрос типа 2 — это схема вида: $?[2]Question[i] \rightarrow cSmthContainer$; (сокращенно $?Q \rightarrow \ldots;$). Здесь: 2 — тип запроса; i — порядковый номер запроса в последовательности запросов. В контейнер $cSmthContainer$ обязательно должен быть вложен радикал вида $uvarSmthName =;$, который требуется определить в рамках данной среды радикалов. В контейнер $cSmthContainer$ могут быть вложены также радикалы-уникумы.

Для ответа на запрос типа 2 будем использовать специализированный контейнер $cAnswers$, объединяющий все ответы $cAnswer$, полученные на запрос. Каждый из контейнеров $cAnswer$ объединяет краткий $(dBrief \rightarrow \ldots)$ и полный $(dFull \rightarrow \ldots)$ 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:

$$\underline{cAnswers} \rightarrow$$
$$\{1d[1]Answer \rightarrow cAnswer \rightarrow \{2d[1]Brief \rightarrow cvars \rightarrow$$
$$\{3d[1]var \rightarrow uvarA = uA; \ldots; d[*]var \rightarrow uvarZ = uZ;\}3$$
$$d[2]Full \rightarrow cSmthContainer;\}2d[*]Answer \rightarrow cAnswer \rightarrow$$
$$\{2d[1]Brief \rightarrow cvars; d[2]Full \rightarrow cSmthContainer;\}2\}1.$$

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

Отметим, что опорная среда радикалов может быть связана, вообще говоря, с различными ультрасредами, что приводит к различным ультраоснащениям этой среды. Используя запросы, мы можем активировать такие ультраоснащенные среды радикалов и получать в результате ответы на запросы, а также осуществлять добавление новых схем. Результат активации зависит как от опорной среды радикалов, так и от ее ультраоснащения и запросов.

Среда радикалов изменяется и развивается с помощью запросов с использованием средств, предназначенных для поиска ответов на эти запросы и называемых активаторами.

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 $d[5]Navigators$ (последнее направление в рассматриваемых ниже представлениях не показано). Рассмотрим эти направления.

Направление 1 активатора ведет к начальному радикалу схемы, на которой ищется ответ на запрос: $cActivator \rightarrow d[1]FL \rightarrow FL$.

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. Вообще говоря, активатор может содержать конечное число контейнеров вида $cNowAQ$ и, соответственно, обрабатывать одновременно конечное число запросов. Здесь мы рассматриваем случай одного контейнера $cNowAQ$. 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. Мы рассмотрим это направление позже.

При визуализации работы радикал-активатора — специального средства активации (а также в некоторых других случаях) оказывается удобным использовать различные срезы среды радикалов, своего рода точки зрения на среду радикалов. Процессы в среде радикалов

удобно представлять преобразованиями векторов типового геометрического отображения (поворот, растяжение, сжатие, сдвиг векторов), которые приводят к формированию ответов на запросы. Часть схем среды радикалов может быть «скрыта» с помощью ультра контейнеров. В результате активации запросов и работы радикал-активатора такие схемы могут быть выявлены и добавлены в явном виде к среде радикалов, которая таким образом «раскрывается». Доказана теорема об активаторе, согласно которой, для любой нормализованной среды радикалов, представленной нормализованной системой векторов геометрического отображения, существует конечная последовательность запросов, с помощью которой радикал-активатор, используя штатные преобразования векторов, полностью раскрывает исходную среду радикалов за конечное число шагов.

8. Технология решения задач АКПУ

Рассмотрим работу активатора на следующем примере (см. рис. 5–9). Пусть в состав некоторой сложной системы входят составляющие — уникумы $u1, u2, u3, u4$. Составляющие $u2$ и $u3$ могут пребывать, по крайней мере, в двух состояниях — они могут быть либо включены, либо выключены. Для этих уникумов имеется координатная система контейнеров. Контейнер $c[2:0]R$ ($c[2:0]$ $CanPutOnRegulary$) говорит о том, что $u1$ может включить $u2$ штатно. Контейнер $c[2:1]R$ — о том, что $u1$ может включить штатно $u3$. Некоторые составляющие являются дублерами других составляющих: $u2$ дублирует $u3$, а $u4$ дублирует $u1$. Такие связи будем представлять контейнерами вида cS ($cSubstituteOfUnit$): $c[2:0]S$ и $c[2:1]S$. Для доступа к контейнерам $c[2:*]R$ и $c[2:*]S$ используется контейнер всех контейнеров $cACs$ ($cAllContainers$).

Имеет место ультра оснащение среды радикалов — в среду включен ультра контейнер cUE ($cUltra2CanPutOnExtremely$), объединяющий переменные $uvar1$, $uvar2$ и $uvar3$, которые могут быть связаны с уникумами-составляющими сложной системы. Согласно этому ультра контейнеру, составляющая $uvar1$ может включить аварийно составляющую $uvar2$ (об этом говорит контейнер заключения $c[2:0]E$

($cCanPutOnExtremely$)), если $uvar1$ — дублер $uvar3$ (что описывается контейнером cS ($cSubstituteOfUnit$)) и $uvar3$ может включить штатно $uvar2$ (контейнер cR ($cCanPutOnRegulary$)). Контейнеры посылки $c[2:2]S$ и $c[2:2]R$ объединены контейнером $cAND$, доступным в ультра контейнере cUE, который, в свою очередь, вложен в контейнер всех ультра контейнеров $cAUCs$ ($cAllUltraContainers$).

Будем поэтапно рассматривать работу активатора для некоторой среды радикалов, используя для нее типовое геометрическое отображение.

Этап 0 (предварительный этап). В среде радикалов размещаются схемы, вложенные в контейнеры $cACs$ и $cAUCs$.

Этап 1. В среду радикалов поступает исходный запрос $?Q[0]$ ($?Question[0]$), имеющий следующий вид: $?Q[0] \rightarrow c[2:1]E \rightarrow \{d[1]Sub \rightarrow u4;\ d[2]Obj \rightarrow uvar4 =;\}$ (в целевом контейнере $c[2:1]E$ доступны направления $d[1]Sub$ ($d[1]Sublect$) и $d[2]Obj$ ($d[2]Object$)). То есть требуется найти составляющие $uvar4$, которые могут быть включены аварийно составляющей $u4$. Ссылка на схему FL и начальный запрос $?Q[0]$ помещаются в головные контейнеры схемы $cActivator$. Далее, запрос продвигается в контейнеры активации cAQ и $cNowAQ$ и активируется там с помощью окрашивания радикалов.

Этап 2. Активатор приступает к поиску в ультраоснащенной среде FL контейнера того же вида, что и целевой контейнер $c[2:1]E$ запроса $?Q[0]$. Поиск осуществляется как в опорной, так и в ультра среде (анализируются контейнеры, доступные в ультра контейнерах по направлению $d[1]Conclusion$). Если контейнер вида $c[2:*]E$ не будет найден, то запрос будет считаться неудачным, а ответом на него будет «пустой» радикал $uNULL$. В нашем случае поиск успешен — найден контейнер cUE, в котором по направлению $d[1]Conclusion$ доступен контейнер заключения $c[2:0]E$.

Запрос связывается с ультра контейнером cUE, контейнер $cNowAQ$ преобразуется (преобразование $a1$) — теперь имеем $cNowAQ \rightarrow d[2] \rightarrow cUE$.

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, которые мы назовем базовыми схемами активации (в этой

статье мы не будем рассматривать их подробно). $cGOTONext$ — 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 уникумов-сообщений, доступный в перечисленных контейнерах. Это могут быть как внутренние сообщения контейнера, образующие протокол его работы, так и внешние сообщения диалога контейнера 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 $cActivator$. (Если имеется более одного контейнера $cNavigator$, возникает возможность параллельной работы активатора над запросами.) В контейнерах $cNavigator$ и $cNavigators$ имеются направления $d[*]Messages$, $d[*]State$ и $d[*]UltraContainers$. Ультра контейнеры навигаторов используются для анализа сообщений, состояния вложенных в них контейнеров и для принятия решения о дальнейших действиях — о выборе контейнеров, об их настройке и активации с помощью окрашивания.

Например, на рассматриваемом этапе 2 нашего примера работа контейнеров $cGOTONext$ и $cCompare$, доступных в контейнере $cActivator$ по направлению $d[5]Navigators$, приведет к сообщению о нахождении контейнера $c[2:0]E$ (по направлению $d[1]Conclusion$ ультра контейнера cUE). Далее, работа системы ультра контейнеров приведет к настройке и активации контейнера замены $cSubstitute$. Контейнер сработает, и цепочка $cActivator \to d[3]AQs \to cAQs \to d[1]AQ \to cAQ \to d[2]AC \to uNULL$; будет заменена цепочкой $cActivator \to \ldots \to d[2]AC \to cUE$;, а цепочка $cActivator \to d[4]NowAQ \to cNowAQ \to d[2]NowAC \to uNULL$; будет заменена цепочкой $cActivator \to \ldots \to d[2]NowAC \to cUE$;.

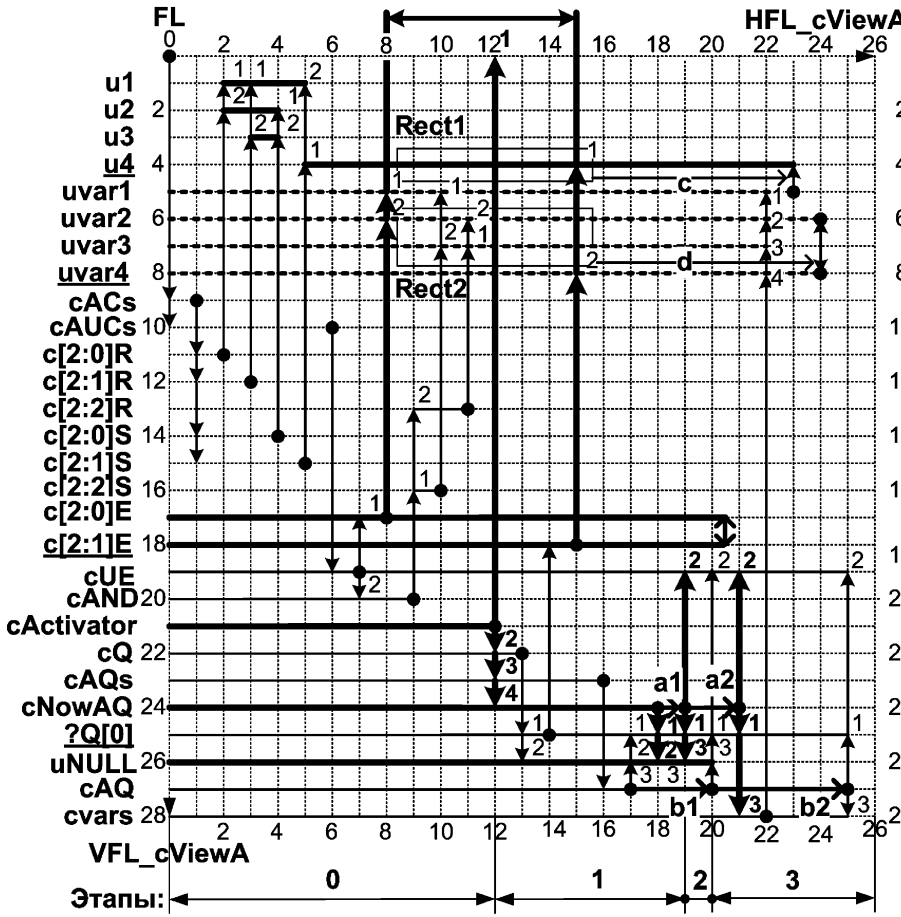

Рис. 6. Активация среды радикалов с целью поиска ответа на запрос.

Активация базовых схем активации осуществляется с помощью «элементарных» промежуточных запросов, порождаемых системой ультра контейнеров. 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.

Для упрощения записи мы не будем рассматривать контейнеры $cNavigators$, $cNavigator$ и вложенные в них контейнеры.

Этап 3. На предыдущем этапе контейнер $cNowAQ$ был преобразован (преобразование $a1$), и мы имеем $cNowAQ \to d[2] \to cUE;$. Теперь контейнер активированного запроса cAQ преобразуется соответственно (преобразование $b1$): $cAQ \to d[2] \to cUE;$. Делается попытка связать радикалы контейнера $c[2:1]E$, принадлежащего запросу, с соответствующими радикалами однотипного контейнера, доступного в ультра контейнере cUE по направлению $d[1]Conclusion$. При этом активатор пользуется правилом связывания радикалов согласуемых однотипных контейнеров, согласно которому связать между собой связкой '=' можно только переменные либо переменную и уникум. В нашем случае связывается переменная $uvar1$ из ультра контейнера с уникумом $u4$ — $uvar1 = u4$ (направление 1 в контейнерах $c[2:*]E$), а также переменная $uvar4$ из запроса с переменной $uvar2$ из ультра контейнера — $uvar4 = uvar2 =$ (направление 2 в контейнерах $c[2:*]E$).

Контейнер $cNowAQ$ преобразуется (преобразование $a2$): теперь $cNowAQ \to d[3]Answer \to cvars;$. Контейнер cvars объединяет имеющиеся переменные в их состоянии на текущий момент времени. (В представлении $cViewA$ цепочки, связывающие радикалы-переменные с радикалом $uNULL$, не показаны.)

Контейнер активированного запроса вновь преобразуется соответственно (преобразование $b2$): $cAQ \to d[3]Answer \to cvars;$.

Этап 3 работы радикал-активатора завершен, но поиск ответов будет продолжен. Мы не будем подробно рассматривать дальнейшие этапы работы радикал-активатора, поскольку они во многом аналогичны этапам 1, 2 и 3. Действия активатора будут аналогичны описанным — реализующим базовый цикл активатора (он будет рассмотрен ниже). Приведем лишь краткие замечания.

Этапы 4–8 (см. рис. 7). От представления $cViewA$ среды радикалов перейдем к представлению $cViewB$, которое содержит: уникумы $u1, u2, u3, u4$ — составляющие сложной системы; переменные $uvar1$, $uvar2$, $uvar3$, $uvar4$; контейнеры $cAUsM$ ($cAllUnitsMany$) и $cAVsM$ ($cAllVariablesMany$), объединяющие, соответственно, используемые уникумы и переменные; исходный запрос $?Q[0]$, а также промежу-

точные запросы $?Q[1], ?Q[2]$, сгенерированные активатором при обработке исходного запроса. Кроме того, представление $cViewB$ содержит схемы $?QContinue, cContinue, uvarContine, uFALSE$ и $uTRUE$, с помощью которых либо прерывается поиск ответов на запрос, либо инициируется его продолжение. По сравнению с $cViewA$, представление $cViewB$ позволяет следить за изменениями среды радикалов с «более высокой» точки зрения.

На этапе 8 мы будем иметь $uvar2 = uvar4 = u2$, то есть получен первый ответ на исходный запрос: составляющей $u4$ может быть включена аварийно составляющая $u2$. Далее активатор генерирует специальный запрос $?QContinue$ о необходимости продолжения поиска других ответов на исходный запрос. Запрос $?QContinue$ обращен к человеку, работающему в среде радикалов (либо к схеме более высокого уровня, управляющей работой данного активатора). В запрос $?QContinue$ вложен контейнер $cContinue$, в котором доступна переменная $uvarContinue$. Эту переменную человек (если запрос обращен к нему) может связать либо с уникумом $uFALSE$, либо с уникумом $uTRUE$. В первом случае поиск ответов будет прерван, во втором — продолжен. Пусть действия человека привели к связыванию $uvarContinue = uFALSE$. В этом случае $uvar4 = u2$ останется единственным ответом на исходный запрос. Дальнейшего поиска других ответов не будет.

Этапы 9–13 (см. рис. 8). Рассмотрим теперь случай, когда действия человека привели к связыванию $uvarContinue = uTRUE$. В этом случае поиск других ответов на исходный запрос будет продолжен.

Рассмотрим соответствующие этапы работы радикал-активатора. Активатор продолжит поиск, и на этапе 10 мы получим второй ответ на исходный запрос: $uvar2 = uvar4 = u3$, то есть составляющей $u4$ может быть включена аварийно составляющая $u3$.

И вновь активатор генерирует запрос $?QContinue$ о необходимости продолжения поиска других ответов на исходный запрос. (На рисунке соответствующие схемы не показаны.) Пусть действия человека привели к связыванию $uvarContinue = uTRUE$. И вновь будет продолжен поиск ответов. На этапах 11–13 будет выяснено, что ответов на исходный запрос больше нет, и радикал-активатор завершит свою работу.

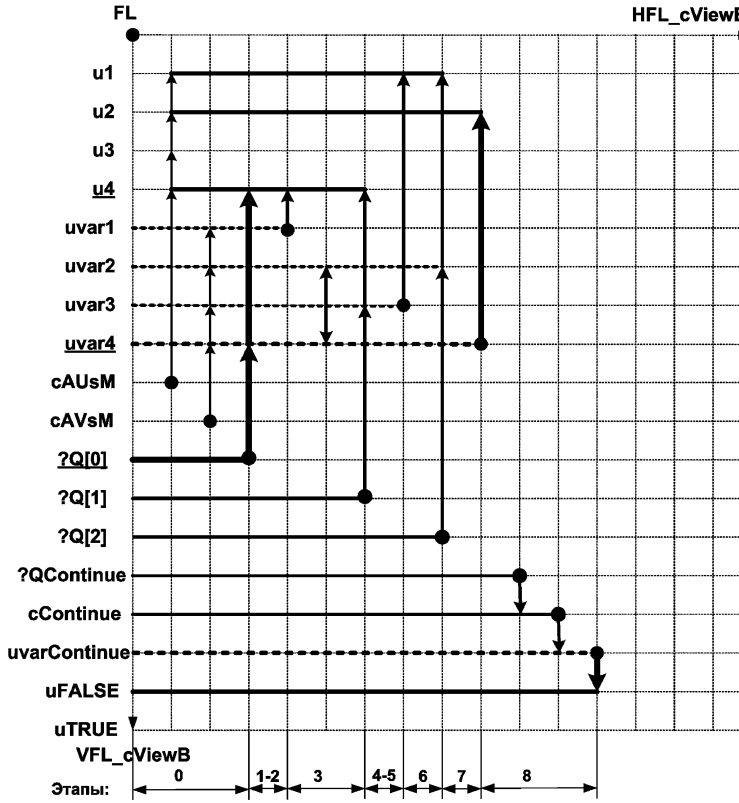

Рис. 7. Связывание переменных и уникумов при получении ответа на запрос. Получение первого ответа и отказ от поиска других ответов.

Пример завершен, и, делая обобщения, приходим к следующему алгоритму работы активатора, который, в главном, представляется следующим образом.

1) Активатор находится в режиме ожидания исходного запроса.
2) Ввод исходного запроса и размещение его в цепочке $cActivator \to d[2]Q \to cQ \to d[1]Q \to \ldots;$.
3) Активация запроса. Размещение запроса в контейнере активированного запроса cAQ и контейнере $cNowAQ$ активированного запроса, обрабатываемого в настоящий момент времени. Для

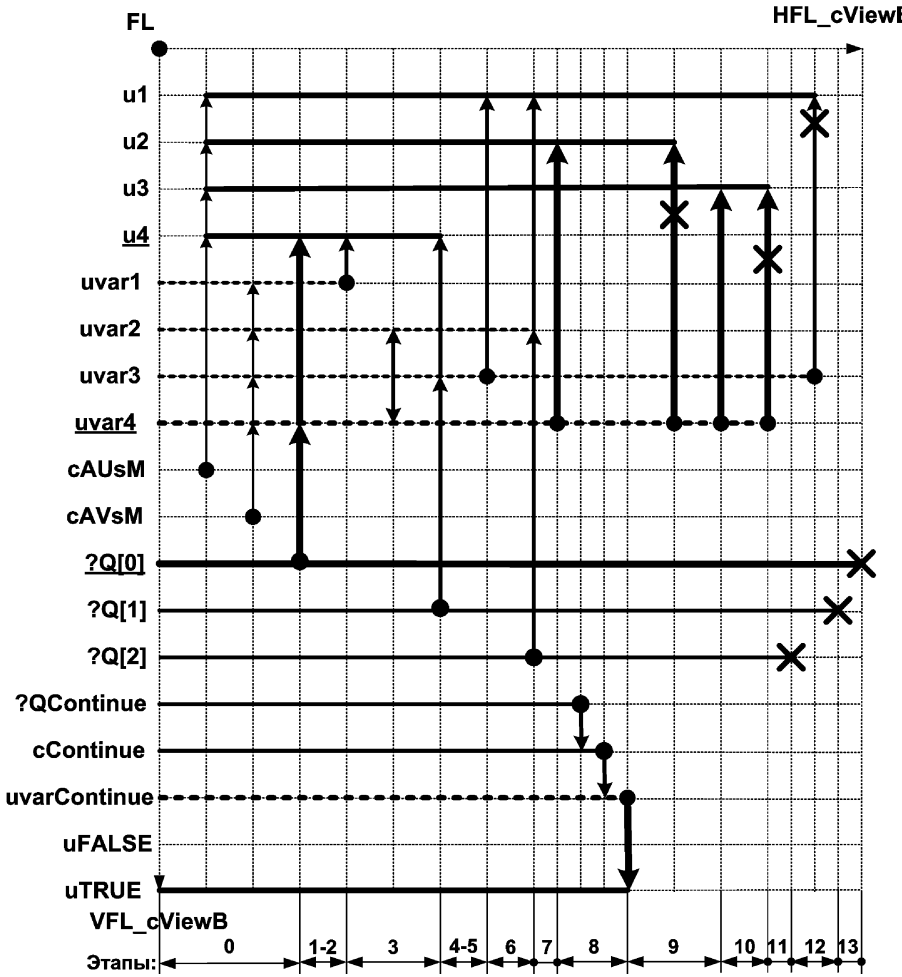

Рис. 8. Завершение поиска ответов на запрос.

размещения запроса используются цепочки
$cActivator \to d[3]AQs \to cAQs \to d[*]AQ \to \ldots$; и
$cActivator \to d[4]NowAQ \to cNowAQ \to d[1]NowAQ \to \ldots$;.

4) Поиск в среде радикалов контейнера, подходящего для согласования. Ищется контейнер вида целевого контейнера запроса.

Поиск осуществляется как в опорной среде, так и в ультра среде (среди контейнеров, доступных в ультра контейнерах по направлению $d[1]Conclusion$).

5) Размещение подходящего для согласования контейнера в цепочках $cActivator \rightarrow d[3]AQs \rightarrow cAQs \rightarrow d[2]AC \rightarrow \ldots;$ и $cActivator \rightarrow d[4]NowAQ \rightarrow d[2]NowAC \rightarrow \ldots;$.

6) Попытка согласования целевого контейнера запроса и найденного контейнера (ультра контейнера).
Для согласования ультра контейнера 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 ультра контейнере по направлениям $d[*]And$, генерируются промежуточные запросы. Затем эти запросы обрабатываются так же, как исходный запрос (п. 3).
Если согласуется не ультра контейнер, а простой контейнер, то согласование заключается в непосредственном применении к нему и к целевому контейнеру запроса правила связывания с помощью связки ‘=’ радикалов согласуемых 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 $cvars \rightarrow \ldots;$.
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. Мы не будем здесь рассматривать этот алгоритм более подробно. Отметим лишь две особенности его применения.

Существуют среды радикалов и запросы, применение к которым активатора ведет к зацикливанию.

Следующая особенность связана с логической равноценностью направлений $d[*]And$ контейнеров $cAND$, принадлежащих ультра контейнерам. Активатор, пытаясь согласовать ультра контейнер и дойдя в нем до контейнера $cAND$, всегда начинает построение промежуточных запросов с направления $d[1]And$. Состав и структура среды радикалов, моделирующей проблемную область сложной системы, вообще говоря, не стабильны. Множество используемых уникумов, переменных, контейнеров и ультра контейнеров, а также запросов, требующих ответа, по разным причинам изменяется. Изменяется и оптимальный порядок обработки направлений контейнеров $cAND$. Если имеется система ультра контейнеров, с помощью которой можно определять оптимальный порядок обработки направлений $d[*]And$, то следует применять direct-радикалы с двумя индексами: $cAND \rightarrow d[1:*][2:*]And \rightarrow \ldots;$. Первый индекс — константа — идентифицирует само направление в контейнере $cAND$. Второй индекс — переменная — порядковый номер направления в очереди направлений контейнера $cAND$ на обработку их активатором.

В технологии ИСБ-решения задач АКПУ, основанной на применении запросов и радикал-активаторов, применяются схемы, с помощью которых представляются задачи. Для каждой задачи обеспечивается представление используемых запросов, начальной схемы, ресурсов, ответов, отчетных схем и подзадач. С помощью схем представляются методы решения задач, а также базовые задачи и соответствующие базовые методы, непосредственно использующие радикал-активаторы. Используются также специальные схемы, применяемые в ходе решения задач для оценки и классификации других схем. Бла-

годаря активации запросов и работе активаторов среда радикалов постоянно находится в «движении». Схемы претерпевают тождественные и нетождественные преобразования; в результате использования ультра контейнеров они развертываются (то есть пополняются новыми схемами), если принадлежат классу развертываемых схем. Класс схемы-ответа на запрос зависит от класса схемы, к которой этот запрос обращен, а также от класса схемы самого запроса. Развертываемая схема может быть развернута полностью (частично) запросом некоторого класса за определенное число шагов. Этому аспекту среды радикалов посвящены доказанные теоремы о не развертываемых и развертываемых схемах.

9. Проблема конфликтов в среде радикалов

В течение жизненного цикла сложной системы контейнеры, характеризующие уникумы, могут преобразовываться, что может приводить к нежелательным конфликтам в смысле знаний, представленных в ультра среде радикалов. Один и тот же уникум одновременно может участвовать во множестве конфликтов различных типов. Возникает проблема разрешения конфликтов (ухода от конфликтов) в целях обеспечения ИСБ с помощью управляющих воздействий, применяемых к управляемым (изменяемым) контейнерам.

Для решения проблемы конфликтов рассматриваются конфликтующие (в смысле ультраоснащения) пары контейнеров среды радикалов. Часто один из контейнеров является «охватывающим», а другой — «охватываемым» в смысле вложенных в них уникумов. Используются приближения сложных контейнеров более простыми контейнерами. В общем случае, рассматриваются как штатные, так и нештатные преобразования контейнеров. С точки зрения приближения пары контейнеров к конфликту или удаления от него (в смысле используемых ультра контейнеров), выделяются нуль-конфликт, минус-конфликт и плюс-конфликт преобразования пары контейнеров. С использованием этих понятий рассматриваются последовательности преобразований контейнеров и выделяются безопасные и опасные последовательности. Вводится понятие критического состо-

яния среды радикалов — такого состояния среды, по достижении которого контроль над ней, при имеющихся ресурсах, теряется.

Разработан двунаправленный метод ухода от конфликтов в среде радикалов. Метод реализует ИСБ-синтез целевого уникума в среде радикалов. Сформулирован тезис о двунаправленном методе обеспечения ИСБ, согласно которому 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.

Практическое использование полученных результатов показало их эффективность на всех этапах жизненного цикла сложной системы. С помощью нормализованных схем радикалов однозначно и наглядно представляются (могут быть представлены) все объекты и связи проблемной области сложной системы, все процессы, протекающие в ней. Удобно представляются системы и подсистемы, компоненты всех видов обеспечений (программного, технического и других), решаемые задачи. С помощью разработанного интерфейса среды радикалов удобно наблюдать за изменяющейся проблемной областью, используя различные ее «сечения» («точки зрения»), «масштабы» для нормализованной среды радикалов. Удобно управлять решением задач, используя типизированные сообщения в целях обеспечения ИСБ. Схемы радикалов наглядны, удобны для отчетов, генерируемых при работе системы, в документации, выпускаемой на различных этапах жизненного цикла системы.

Список литературы

[1] Чечкин А. В. Информационно-системная безопасность сложной системы // Математические методы решения инженерных задач. М.: МО РФ, 2006.

[2] Чечкин А. В. Математическая информатика. М.: Наука, 1991.

[3] Соболева Т. С., Чечкин А. В. Дискретная математика. М.: Издательский центр «Академия», 2006.

[4] Ильичев А. В. Начала системной безопасности. М.: Научный мир, 2003.

[5] Надежность и эффективность в технике. Справочник в 10 т. Т. 3 / под ред. В. Ф. Уткина, Ю. В. Крючкова. М.: Машиностроение, 1988.

[6] Воронков Г. С. Механизмы решения задач в элементарном сенсориуме // Нейрокомпьютеры: разработка и применение. 2004. № 2–3. С. 93–100.

[7] Чечкин А. В. Обеспечение информационно-системной безопасности сложной системы на основе среды нейрорадикалов её проблемной области // Нейрокомпьютеры: разработка и применение. М.: Радиотехника, 2008. № 7. С. 6–11.

[8] Чечкин А. В., Пирогов М. В. Методика обеспечения информационно-системной безопасности сложных систем // Нейрокомпьютеры: разработка и применение. М.: Радиотехника, 2008. № 7. С. 11–17.