

Интеллектуализация сложной системы как средство обеспечения её информационно-системной безопасности

А. В. ЧЕЧКИН

*Московский государственный университет
им. М. В. Ломоносова
e-mail: a.checkkin@mail.ru*

М. В. ПИРОГОВ

*Научно-производственное объединение
им. С. А. Лавочкина
e-mail: polvad@laspace.ru*

УДК 519.95

Ключевые слова: сложная система, безопасность, радикал, конфликт, ультраконтэйнер.

Аннотация

Исследуется проблема информационно-системной безопасности сложной системы. Основным вопросом обеспечения безопасности такой системы является оснащение её элементами интеллекта.

Abstract

A. V. Chechkin, M. V. Pirogov, Intellectualization of a complex system as a means of maintaining its information system safety, Fundamentalnaya i prikladnaya matematika, vol. 15 (2009), no. 3, pp. 225–239.

In this article, the problem of information system safety of a complex system is analyzed. The main task of ensuring security of such a system is equipping it with intellect elements.

1. Проблема

Мы исследуем проблему обеспечения информационно-системной безопасности сложной системы. Это понятие характеризует выживание сложной системы и включает в себя две части: информационную и системную.

Информационная безопасность — это особого рода функциональная устойчивость сложной системы, при которой обеспечивается безусловное решение всех частных задач жизненного цикла системы вне зависимости от формы (от языка) представления входной информации и от полноты этой информации путём использования дополнительной информации. Информационная безопасность

Фундаментальная и прикладная математика, 2009, том 15, № 3, с. 225–239.

© 2009 Центр новых информационных технологий МГУ,

Издательский дом «Открытые системы»

сложной системы реализуется переходом от алгоритмов к соответствующим логическим рассуждениям (принятию решений). В терминах математической информатики это переход от операторов к ультраоператорам [1].

Системная безопасность сложной системы — это безусловное сохранение её главных функций, основных составляющих сложной системы и их связей при решении любой частной задачи жизненного цикла для обеспечения должного функционирования сложной системы, её системной целостности в рамках её метасистемы (театра действий сложной системы) [1]. Системная безопасность сложной системы — это постоянный учёт и устранение конфликтов между её составляющими и их связями, которые появляются в течение жизненного цикла системы, а также конфликтов между самой сложной системой и внешними к ней системами. Системная безопасность требует реализации системного подхода к сложной системе с учётом её метасистемы [1].

2. Интеллектуализация сложной системы на основе среды радикалов

Большие разрушительные последствия нарушения информационно-системной безопасности поведения сложной системы для существования самой системы, экологические катастрофы, ошибки, связанные с человеческим фактором в сложных системах, приводящие зачастую к непредсказуемым последствиям для системы, — это и многое другое говорит о том, что проблема обеспечения информационно-системной безопасности выдвигается на одно из первых мест, причём чем сложнее система, тем важнее для неё обеспечение информационно-системной безопасности в течение её жизненного цикла.

Основной подход к обеспечению информационно-системной безопасности сложной системы — это *интеллектуализация* такой системы, т. е. оснащение системы элементами интеллекта, создание у неё специального информационно-программного окружения, обеспечивающего безопасность поведения этой системы в рамках её метасистемы на протяжении всего её жизненного цикла. Такое оснащение должно обеспечивать постоянную адаптацию сложной системы к изменяющимся внутренним и внешним условиям, проводить диагностику, контроль, анализ и синтез отдельных составляющих системы и функционирования системы в целом с учётом последствий этого функционирования.

По-видимому, интеллектуализация сложных систем будет основным процессом в современных технологиях. Уже сейчас происходит интеллектуализация современных информационных, транспортных и коммуникационных, военных и гражданских, коммунальных, научно-производственных, технических (автомобилей, летательных аппаратов и т. д.) систем. Признаками такого процесса является, например, компьютеризация и информатизация во всех сферах человеческой деятельности, переход от традиционных систем управления к адаптивным системам и многое другое.

Термином «интеллектуализация» подчёркивается главное свойство такого информационно-программного окружения — возможность расширения круга решаемых этим окружением штатных задач информационно-системной безопасности поведения за счёт самообучения, т. е. поиска решений некоторых нештатных для неё задач. Чем больше разнообразных и более трудных нештатных задач может решать такое окружение, тем оно интеллектуальнее.

Основой интеллектуализации любой системы является эффективная реализация метода проб и ошибок, а значит, сложная система должна быть снабжена некоторым набором свободных элементов, предназначенных для конструирования таких проб. На наш взгляд, лучшим средством организации творческого процесса поиска нового является так называемая *среда радикалов*. На её основе организуется поиск информационно-системной безопасности поведения и обучение сложной системы.

Понятие радикала является главным понятием теории интеллектуальных систем и, по-видимому, всей дискретной математики [1—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. Все активные радикалы образуют *системоквант*, который определяет *квант* поведения системы на данный момент. Тем самым среда радикалов является своего рода набором для создания системокванта своего поведения.

Например, элементы детского конструктора «Лего» являются радикалами, из них можно собирать различные конструкции. Тот элемент, который используется в данной конструкции, является активным, а не используемый — пассивным. Создаваемая при этом конструкция будет системоквантом.

Любой объект управления, который может иметь различные состояния (фазовые координаты) в зависимости от параметров управления, является системоквантом. При этом наборы возможных управляющих параметров образуют среду радикалов. Приведём другие примеры радикалов.

1. Точки непустого множества являются радикалами, если их выбирают для оценивания по некоторому критерию или если они участвуют в алгебраических операциях в качестве образующих и т. д.
2. Буквы алфавита — радикалы. Они предназначены для составления из них слов (системоквантов). Та буква, которая используется в данном слове, является активной, а та, которая не используется, — пассивной.
3. Слова лексики любого языка — радикалы. Они предназначены для составления из них текстов (системоквантов).
4. Тексты (записи) любой базы данных — радикалы. Они предназначены для составления ответов (системоквантов) на запросы к базе данных.
5. Компьютерные программы, алгоритмы, вычислительные системы, автоматы, предназначенные для обработки текстов (информации), являются радикалами, если имеется возможность их использовать или нет (включать или выключать).

6. Военные подразделения вооружённых сил являются радикалами, так как их можно активировать приказом на боевые действия или вывести из боевых действий.
7. Функциональные системы живого организма или навыки являются радикалами, если имеется возможность их активировать или не активировать в зависимости от внешней ситуации.

Радикалы, предназначенные обеспечивать информационно-системную безопасность поведение сложной системы в рамках её метасистемы, образуют множество с разнообразными отношениями. Поэтому будем говорить, что радикалы образуют *среду*. Среда радикалов является рабочей подсистемой информационно-программного окружения и отражает опыт предыдущих этапов жизненного цикла системы. Процесс управления активацией среды радикалов обеспечивается активирующей подсистемой, которая, в свою очередь, использует для этого среду специфических активаторов (рис. 1).



Рис. 1. Интеллектуализация сложной системы

Предполагается, что в зависимости от ситуации, в которой функционирует сложная система с учётом её метасистемы, часть радикалов может быть активирована для решения задачи безопасности её поведения. Схему из активных радикалов, которая обеспечивает поведенческую реакцию на ситуацию, будем называть системоквантом.

В результате такой организации информационно-программного окружения сложной системы формируется интеллектуальная система, в предметной области которой находится сложная система и её метасистема. Целью такой интеллектуальной системы является информационно-системная безопасность поведения сложной системы. Структурная схема интеллектуальной системы представлена на рис. 2.

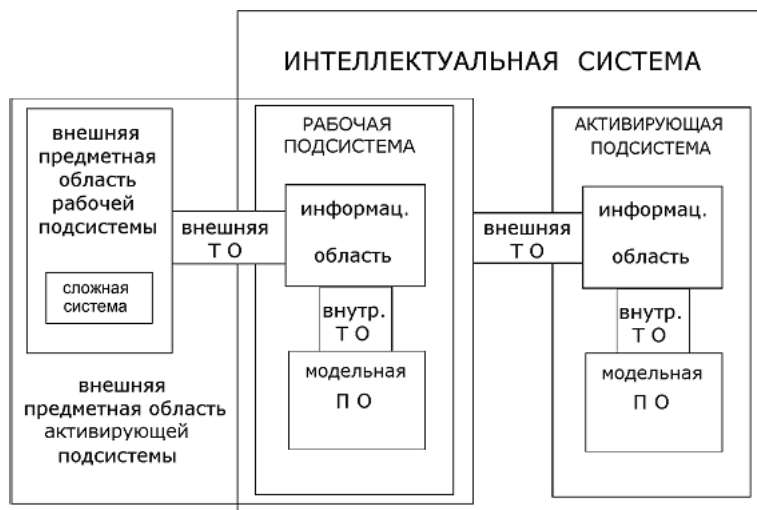


Рис. 2. Структурная схема интеллектуальной системы

3. Функционирование интеллектуальной системы

Работа интеллектуальной системы, являющейся информационно-программным оснащением сложной системы для обеспечения её информационно-системной безопасности в рамках метасистемы, заключается в следующем (рис. 3).

Этап 1. Системный анализ. При системном анализе очередного этапа жизненного цикла сложной системы в рамках её метасистемы интеллектуальная система проводит оценивание каждого действия сложной системы с целью обеспечения её информационно-системной безопасности. Всякая задача сначала исследуется интеллектуальной системой на предмет, является ли она штатной или нештатной. Этот этап можно назвать мотивацией интеллектуальной системы.

Этап 2. Адаптация. Если задача штатная, то далее происходит запрограммированное активирование соответствующих радикалов (системокванта) в рабочей подсистеме и решается задача. Если результат решения задачи будет эффективным, то интеллектуальная система выполнила поставленную задачу адаптивного поведения и переходит к следующей задаче. Этот этап можно назвать эксплуатацией или штатным поведением.

Если результат решения штатной задачи не приводит к адаптации поведения сложной системы, то задача объявляется нештатной и далее происходит творческий процесс (этапы 3—5) поиска (создания) нового системокванта.

Этап 3. Системный синтез. Если задача нештатная (см. рис. 3), то происходит анализ условий задачи с учётом имеющихся радикалов (опыта) в среде радикалов и ограничений к сложности (затратам) будущего системокванта. Тем

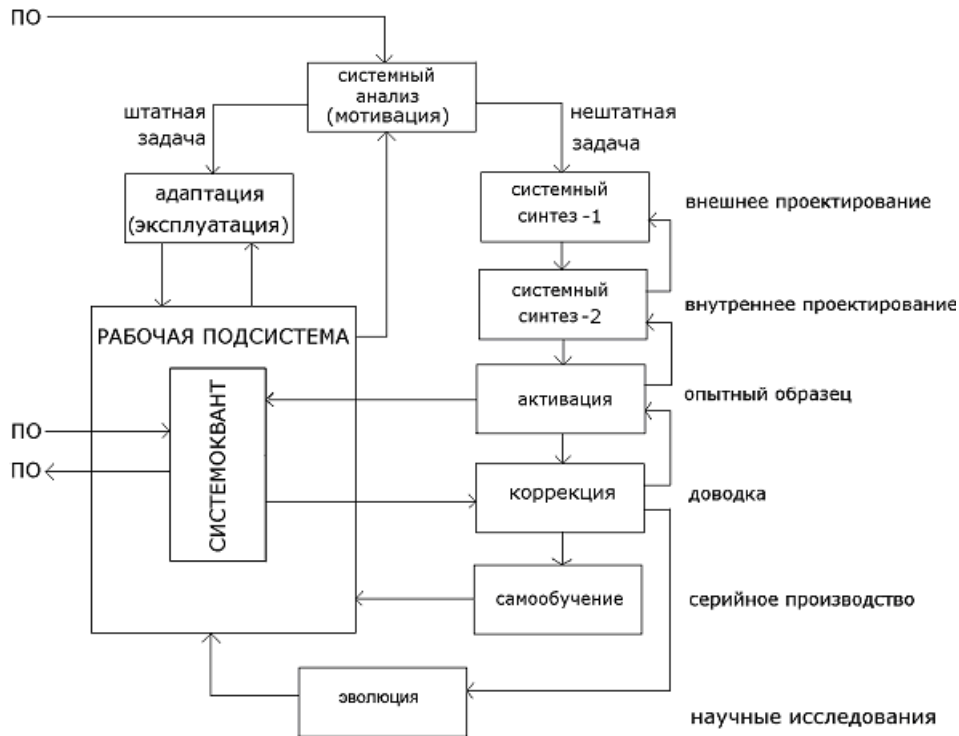


Рис. 3. Функционирование интеллектуальной системы

самым происходит обоснование и выработка как бы тактико-технического задания на системоквант. Этот этап можно назвать системным синтезом-1 или внешним проектированием системокванта.

Далее идёт системный синтез-2, т. е. создание проекта системокванта. Этот этап включает разработку метода и алгоритма решения задачи с требуемой эффективностью в форме сетевого плана активации имеющихся радикалов (заготовок), включая выработку признака (акцептора) окончательного результата, т. е. условия остановки решения. Этот этап можно назвать внутренним проектированием (см. рис. 3).

Этап 4. Активация (самоорганизация среды радикалов). Происходит реализация процесса решения задачи в форме создания системокванта в режиме самоорганизации активации среды радикалов. Этот этап можно назвать созданием опытного образца системокванта (рис. 3).

Этап 5. Коррекция. С учётом оценивания хода решения задачи и проверки условия остановки создания системокванта производится модификация системокванта. Если необходимо, производится несколько попыток модификации

системокванта. Этот этап можно назвать доводкой (отладкой, настройкой) системокванта (см. рис. 3).

Этап 6. Самообучение. В случае успешного решения нештатной задачи интеллектуальная система закрепляет полученный опыт в форме создания нового радикала целиком из системокванта или новых радикалов из отдельных частей системокванта. При каждом последующем решении той же задачи интеллектуальная система имеет дело уже со штатной задачей, для которой есть готовый план активации в среде радикалов. Этот этап можно назвать серийным производством системокванта в среде радикалов (см. рис. 3).

Этап 7. Эволюция. В случае безуспешных попыток решения нештатной задачи интеллектуальная система исследует причины неудачи и организует создание новых и уничтожение старых базовых элементов системы, из которых образуются радикалы. Этот этап можно назвать фундаментальными научными исследованиями или научно-исследовательскими и опытно-конструкторскими работами.

4. Язык радикалов и нормализация среды радикалов

Для описания среды радикалов введём специализированный язык радикалов [2].

Алфавит языка радикалов — это множество, состоящее из следующих символов: пробел, строчные и заглавные буквы латинского и русского алфавитов, цифры от 0 до 9 и символы

$$+, -, *, /, =, (,), \{, \}, <, >, _ , ;, ?, \rightarrow, \leftarrow, [,], \dots$$

Каждому радикалу поставим в соответствие его индивидуальное *имя*. Приведём примеры имён радикалов:

$$N9, Name9, Name[9], Name[*].$$

На множестве всех радикалов определим бинарное отношение, которое будем обозначать при помощи стрелок \rightarrow и \leftarrow .

Из имён радикалов (звеньев) строятся прямые и обратные цепочки радикалов, например:

$$Name9, N21 \rightarrow N22 \rightarrow N23, N23 \leftarrow N22 \leftarrow N21;$$

Схема радикалов определяется как конечное множество цепочек радикалов. Радикалы могут образовывать структуры, которые мы будем называть *ветвлениями* и *схождениями*. Приведём примеры (через $d[0]$, $d[1]$ обозначены вспомогательные радикалы направлений).

Ветвление: $N30 \rightarrow \{d[0] \rightarrow N31 \rightarrow N32; d[1] \rightarrow N41 \rightarrow N42\}$.

Схождение: $\{d[0] \rightarrow N51 \rightarrow N52; d[1] \rightarrow N61 \rightarrow N62\} \rightarrow N71;$

Вложение схем радикалов определяется с помощью фигурных скобок, возможно с числовыми индексами, например:

$$\begin{aligned} N0 \rightarrow \{ & 0 N1 \rightarrow N11 \rightarrow \\ & \{ 1 N111 \rightarrow N1110; \\ & \quad N112 \rightarrow N1120; \\ & N2 \rightarrow N21 \rightarrow \\ & \{ 2 N211 \rightarrow N2110; \\ & \quad N212 \rightarrow N2120; \} 2 \} 1 \} 0. \end{aligned}$$

Таким образом, вся среда радикалов на этом языке представляет одну большую схему радикалов. Для организации интерфейса введём отображения схем радикалов на плоскость. На рис. 4 представлен пример такого отображения для ветвления N30 из примера.

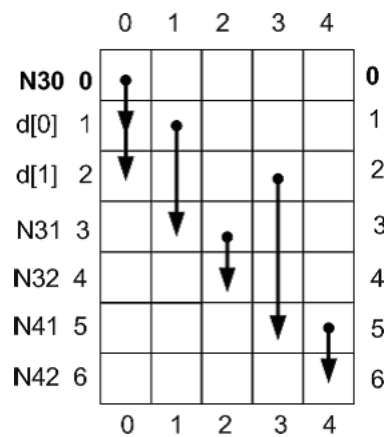


Рис. 4. Отображение схемы радикалов N30 на плоскость

Для обеспечения информационно-системной безопасности решения задач в среде радикалов проведём *нормализацию* среды радикалов в три этапа. На первом этапе проведём *координатизацию* среды с целью быстрого доступа к основным элементам учёта среды и организации интерфейса среды. На втором этапе проведём *ультраоснащение* среды радикалов с целью привлечения знаний о предметной области и их пополнения. Наконец, на третьем этапе реализуем общие организационные *принципы* нормализации среды радикалов.

Первый этап. Координатизация среды радикалов состоит в определении двух видов радикалов: предметных радикалов (*уникумов*) и предикатных радикалов (*контейнеров*). Уникум — это составляющая (элемент учёта) сложной системы или её метасистемы. Контейнер — это форма задания свойства радикала или данного типа связи между радикалами.

Уникумом называется радикал, имя которого обязательно начинается с символа «u» (от слова «unicum») и содержит шесть индексов:

$$u [1:*] \dots [6:*] \text{SmthUnicum};$$

С помощью первого индекса идентифицируется тип уникального радикала, например целое число, конечная десятичная дробь, истинностное значение, единица измерения длины, составляющая сложной системы, составляющая метасистемы и т. д. С помощью второго индекса идентифицируется массив уникамов. С помощью третьего — индивидуальный индекс уникама. С помощью четвертого — версия уникама. С помощью пятого — модификация уникама. С помощью шестого — копия уникама. Допускается сокращённая запись вида $u\text{SmthUnicum}$; или $u\text{SU}$;

Контейнером называется схема радикалов вида

$$c\text{SC} \rightarrow \{d\text{SD1} \rightarrow \dots; \dots; d\text{SDK} \rightarrow \dots\}.$$

Имя первого радикала обязательно имеет первым символом «с» (от слова «container») и содержит шесть индексов:

$$c [1:*] \dots [6:*] \text{SmthContainer}$$

(допускается сокращённая запись: $c\text{SmthContainer}$; или $c\text{SC}$);. Индексы контейнеров определяются аналогично индексам уникамов. Контейнеры содержат специализированные радикалы направления вида

$$d [1:*] [2:*] \text{SmthDirect}$$

(допускается сокращённая запись: $d[*] \text{SD}$, $d\text{SD}$), предназначенные 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. Число направлений в контейнере называется *местностью* контейнера. В одноместном контейнере направление можно опускать, например, вместо $c\text{SC} \rightarrow \{d\text{SD} \rightarrow \dots\}$ писать $c\text{SC} \rightarrow \dots$;

В качестве примера рассмотрим контейнер вхождения, используемый для описания непосредственного вхождения некоторых составляющих в сложную систему, а также для описания непосредственного вхождения одних составляющих в другие:

$$c\text{WholeAndParts} \rightarrow \{d[0] \text{Whole} \rightarrow \dots; d[1] \text{Parts} \rightarrow \dots\}.$$

Здесь по направлениям $d[0] \text{Whole}$ и $d[1] \text{Parts}$ доступны следующие схемы:

$$\begin{aligned} c\text{WholeAndParts} \rightarrow d[0] \text{Whole} \rightarrow c\text{UnitVar} \rightarrow \\ \{0 \ d[0] \text{UnitVarValue} \rightarrow c\text{UnitVarValue} \rightarrow \\ \{1 \ d[0] \text{Many} \rightarrow \dots; \\ d[1] \ \text{ChoiceUnitOfMany} \rightarrow \dots \rightarrow u\text{SmthUnitN};\}1\}0, \end{aligned}$$

$$\begin{aligned}
 & c\text{WholeAndParts} \rightarrow d[1] \text{Parts} \rightarrow c\text{Parts} \rightarrow \\
 & \{ d[0] \text{Part} \rightarrow \dots \rightarrow u\text{SmthUnitA}; \dots; \\
 & d[*] \text{Part} \rightarrow \dots \rightarrow u\text{SmthUnitZ}; \}.
 \end{aligned}$$

В результате процесса координатизации среда радикалов приобретает семантическую *координатную систему контейнеров*, в которую «вложены» уникамы. Координатная система контейнеров является единой для всей среды радикалов. Каждый уникам обязательно «вкладывается» в некоторый контейнер, но обычно в несколько различных и разнотипных контейнеров. Координатная система контейнеров позволяет осуществить структуризацию семейства радикалов, эффективно решать проблему выбора радикала. Это самоконтролируемая, самокорректирующаяся и развивающаяся (изменяющаяся) система контейнеров. С помощью координатной системы контейнеров обеспечивается существенная часть информационно-системной безопасности сложной системы. На основе координатной системы контейнеров создаётся дружественный интерфейс среды радикалов, в результате чего человеку, работающему в этой среде, предоставляется возможность визуального контроля и интерактивного взаимодействия со средой радикалов. Основу интерфейса среды радикалов составляет её визуализация (геометризация), направленная на применение компьютерной графики (см. рис. 3).

Второй этап. Ультраоснащение среды радикалов состоит в создании дополнительной специализированной части такой среды, названной *ультрасредой*. Ультрасреда образуется из ультрарадикалов (знаний о предметной области), предназначенных для выявления и снятия конфликтов, системных нарушений целостности сложной системы. Тем самым ультрасреда (как и координатная система контейнеров) предназначена для информационно-системной безопасности решения задач жизненного цикла сложной системы. Среду уникамов и их контейнеров будем называть *опорной средой*. Для осуществления связи между опорными радикалами и ультрарадикалами организуется *терминальная среда* из радикалов-исполнителей и радикалов-датчиков. Ультрасреда вместе с терминальной средой определяют так называемое ультраоснащение среды радикалов. Для построения ультрасред будем использовать схемы радикалов специального вида: ультраконтейнеры двух типов, реализующие *импликацию*. *Ультраконтейнеры типа 1* имеют пустую посылку и используются при поиске схем в опорной среде радикалов, а также для добавления новых схем к этой среде. *Ультраконтейнеры типа 2* имеют непустую посылку и учитывают условия их применения.

Ультраконтейнер типа 1 — это схема, начинающаяся звеном вида

$$\text{Ultra1C}[*][*] \text{SmthName}$$

(допускается сокращённая запись $\text{U1C} \text{SmthName}$), объединяющая схему-заключение

$$d[0] \text{Conclusion} \rightarrow \dots;$$

(сокращённо $d[0] \text{ Cln} \rightarrow \dots$;) со схемой-посылкой

$$d[1] \text{ Premise} \rightarrow;$$

(сокращённо $d[0] \text{ Prm} \rightarrow$;) и имеющая вид

$$\begin{aligned} & \text{Ultra1C}[*][*] \text{ SmthName} \rightarrow \\ & \{1 \ d[0] \text{ Conclusion} \rightarrow \text{cSmthContainer} \rightarrow \\ & \quad \{2 \ \dots \ d[*] \rightarrow \text{cSmthUnitsMany} \rightarrow \{3 \ \dots\}3 \\ & \quad \quad d[*] \text{ ChoiceUnitOfMany} \rightarrow \text{Name}[i]; \dots\}2 \\ & \quad d[1] \text{ Premis} \rightarrow; \}1, \end{aligned}$$

где посылка обязательно оканчивается пустой схемой. Здесь первый индекс в $\text{Ultra1C}[*][*] \text{ SmthName}$ — это тип ультраконтейнера, второй индекс — экземпляр ультраконтейнера. Каждый радикал $\text{Name}[i]$, $i = 0, \dots, N$, — это либо уникам вида uSmthUnicum , либо переменный радикал вида $\text{VarSmthName} =$. Переменные (неопределённые) радикалы используются при поиске схем в среде опорных радикалов, а также для добавления новых схем к этой среде.

Ультраконтейнер типа 2 — это схема вида

$$\begin{aligned} & \text{Ultra2C}[*][*] \text{ SmthName} \rightarrow \\ & \{1 \ d[0] \text{ Conclusion} \rightarrow \text{cSmthContainer} \rightarrow \\ & \quad \{2 \ \dots \ d[*] \rightarrow \text{cSmthUnitsMany} \rightarrow \{3 \ \dots\}3 \\ & \quad \quad d[*] \text{ ChoiceUnitOfMany} \rightarrow \text{Name}[*]; \dots\}2 \\ & \quad d[1] \text{ Premise} \rightarrow \text{cAnd} \rightarrow \\ & \quad \quad \{2 \ d[0] \text{ Premise} \rightarrow \text{cSmthContainer} \rightarrow \\ & \quad \quad \quad \{3 \ \dots \ d[*] \rightarrow \text{cSmthUnitsMany} \rightarrow \{4 \ \dots\}4 \\ & \quad \quad \quad \quad d[*] \text{ ChoiceUnitOfMany} \rightarrow \text{Name}[*]; \dots\}3 \dots \\ & \quad \quad d[*] \text{ Premise} \rightarrow \text{cSmthContainer} \rightarrow \\ & \quad \quad \quad \{3 \ \dots \ d[*] \rightarrow \text{cSmthUnitSMany} \rightarrow \{4 \ \dots\}4 \\ & \quad \quad \quad \quad d[*] \text{ ChoiceUnitOfMany} \rightarrow \text{Name}[*]; \dots\}3\}2\}1. \end{aligned}$$

Здесь $\text{Name}[*]$ — это либо уникам, либо неопределённый радикал.

Ультрасреда определяется множеством всех ультраконтейнеров, объединённых начальным контейнером вида $\text{cAllUltraContainers}$. Опорная среда, начинающаяся с радикала cAllContainers , и ультрасреда, начинающаяся с радикала $\text{cAllUltraContainers}$, называются *связанными* средами, если существует схема вида

$$\text{FirstLink} \rightarrow \{d[*] \rightarrow \text{cAllContainers}; \ d[*] \rightarrow \text{cAllUltraContainers}; \}.$$

Полученную среду (схему) $\text{FirstLink} \rightarrow \dots$ будем называть ультраоснащённой средой (схемой).

Тем самым в среде радикалов будем различать три области (см. рис. 2): модельную, информационную и внутреннюю терминальную. Последняя осуществляет связь между модельной и информационной областями. Модельная область — это опорная среда радикалов-моделей отдельных составляющих сложной системы и её внешних систем в рамках метасистемы. Информационная область — это ультрасреда радикалов—знаний о связях между составляющими сложной системы и между ней и внешними к ней системами в виде продукций, теорем, методов решения штатных задач и т. п. Связь между этими областями осуществляет внутренняя терминальная область из радикалов-датчиков и радикалов-исполнителей, которые мы объединяем в терминальную среду.

Третий этап. На этом этапе происходит организация среды радикалов по специальным принципам. Соблюдение этих принципов вместе с наличием координатной системы контейнеров и ультраоснащения обеспечивает информационно-системную безопасность решения задач жизненного цикла сложной системы, корректное рассмотрение разнообразных структур в проблемной области сложной системы, её параметров и составляющих, изменений проблемной области во времени, вариантности при построении отдельных составляющих и всей сложной системы в целом, взаимодействия между составляющими и др.

Приведём примеры принципов организации среды радикалов.

Принцип первого звена. Среда радикалов начинается со специального первого звена, имеющего имя FirstLink, или, короче, FL.

Принцип первого контейнера. Следующим за звеном FirstLink идёт контейнер cAllContainersOfFirstLink, или, короче, cACsOfFL, объединяющий все контейнеры верхнего уровня.

Принцип выделения всех систем верхнего уровня. В объединяющий контейнер cACsOfFL должен входить контейнер cAllSystems (cASs), предназначенный для выделения всех зарегистрированных систем верхнего уровня — уникамов вида uSmthSystem (uSS).

Таким образом, нормализованная схема всей среды радикалов имеет вид

$$FL \rightarrow cACsOfFL \rightarrow dASs \rightarrow cASs \rightarrow dSS \rightarrow uSS;$$

Принцип неизвестной системы. В контейнере cASs всегда должно присутствовать звено вида uNullSystem, или, короче, uNS, — уникам, соответствующий системе, о которой ничего не известно — уникам пустой системы. Это специфическое звено предназначено для появления новых схем в среде радикалов. В частности, непосредственная работа над созданием новой системы начинается именно с этого звена. Если это звено будет конкретизировано (с использованием контейнеров), то оно заменится на звено вида uSS, а в контейнере cASs должно быть включено новое звено вида uNS.

Есть много других принципов нормализации среды радикалов.

5. Активация среды радикалов

Для информационно-системной безопасности решения задач жизненного цикла сложной системы при помощи среды радикалов необходима активирующая подсистема (см. рис. 2). Такую систему можно разработать многими современными средствами, например CASE, WORKS FLOW, CALS и др. Нами разработаны в формате SQL специализированные радикалы активирующей подсистемы, называемые активаторами. Они обеспечивают информационно-системную безопасность решения задач в форме обработки запросов. В частности, активаторы обеспечивают навигацию в среде опорных радикалов, а также выделение в среде радикалов тех или иных схем для последующей их активации. Проблема решается с помощью понятия помеченной схемы радикалов.

Помеченным звеном схемы называется любое звено, которому в цепочке радикалов предшествует служебное звено вида $m[*]$ («m» от слова «маркер»). Индекс — число из множества $\{0, 1, \dots\}$ — определяет цвет выделения. Для одновременного окрашивания звена в разные цвета используются маркеры вида $m[1:*][2:*] \dots [n:*]$.

Служебные звенья вида $m[*]$ используются для динамической ориентации в среде радикалов. Отмена пометки звена осуществляется удалением служебного звена $m[*]$, а также стоящего справа от него символа \rightarrow .

Помеченной схемой радикалов называется любое множество цепочек радикалов, каждое звено которых помечено одним и тем же цветом, т. е. служебным звеном вида $m[*]$ с одинаковым индексом.

Пример. Пусть имеется схема радикалов

$$N30 \rightarrow \{d[0] \rightarrow N31 \rightarrow N32; d[1] \rightarrow N41 \rightarrow N42; \}.$$

Пометим некоторые звенья цепочек этой схемы одним цветом $m[0]$:

$$m[0] \rightarrow N30 \rightarrow \{m[0] \rightarrow d[0] \rightarrow m[0] \rightarrow N31 \rightarrow m[0] \rightarrow N32; \\ d[1] \rightarrow N41 \rightarrow N42; \}.$$

Выделим помеченную схему и дадим ей собственное имя N50:

$$N50 = m[0] \rightarrow N30 \rightarrow \{m[0] \rightarrow d[0] \rightarrow m[0] \rightarrow N31 \rightarrow m[0] \rightarrow N32; \}.$$

Теперь удалим пометки:

$$N50 = N30 \rightarrow \{d[0] \rightarrow N31 \rightarrow N32; \}.$$

Тем самым мы создали новый радикал N50.

Среда радикалов подвержена изменению и развитию, что вызывается обработкой запросов активирующей подсистемой (см. рис. 2). Имеются запросы двух типов. *Запросы типа 1* используются для активации среды радикалов с целью подтверждения (опровержения) факта существования в ней схемы, аналогичной схеме в запросе. *Запросы типа 2* содержат имена переменных схем и используются для активации опорной среды радикалов с целью связывания этих имён с конкретными схемами среды.

Определим запросы типа 1. Пусть `FirstLink` — начальное звено схемы радикалов, описывающей проблемную область. Пусть звено `FirstLink` связано с контейнером `cAllContainers`, объединяющим используемые для описания проблемной области контейнеры вида `cSmthContainer`:

$$\begin{aligned} & \text{FirstLink} \rightarrow \text{cAllContainers} \rightarrow \\ & \quad \{1 \text{ d[*]} \rightarrow \text{cSmthContainer} \rightarrow \\ & \quad \{2 \text{ d[0] Many} \rightarrow \text{cSmthUnitsMany} \rightarrow \\ & \quad \{3 \text{ d[0] UofM} \rightarrow \text{uUnitNull}; \dots \text{ d[*] UofM} \rightarrow \text{uSmthUnicum}; \dots\}3 \\ & \quad \text{d[1] ChoiceUnitOfMany} \rightarrow \text{uSmthUnicum}; \dots\}2 \dots\}1. \end{aligned}$$

Запрос типа 1 — это схема вида

$$?[1] \text{ Question}[i] \rightarrow \text{FirstLink} \rightarrow \text{cAllContainers} \rightarrow \text{d[*]} \rightarrow \text{cSmthContainer}; .$$

Здесь: $?[1]$ — тип запроса, i — порядковый номер запроса в последовательности запросов. Смысл запроса типа 1 — выяснить, существует ли конкретная схема в среде радикалов, завершающая цепочку $?[1] \text{ Question}[i] \rightarrow \dots$.

Для представления ответа на запрос типа 1 будем использовать специализированный контейнер `cAnswer`, объединяющий краткий (`dBrief` → ...) и полный (`dFull` → ...) ответы. Краткий ответ фиксирует факт подтверждения (опровержения) существования данной схемы в среде радикалов. Полный ответ представляет собой результат навигации в среде радикалов в форме полной цепочки (пути) для конкретной схемы или, в случае отсутствия таковой, в форме пустой схемы.

Запрос типа 2 обязательно содержит имена переменных схем и используется для активации опорной среды радикалов с целью связывания этих имён (имени) с конкретными схемами среды. В случае успешного связывания запрос типа 2, как и запрос типа 1, приводит к отысканию соответствующих схем радикалов.

Запрос типа 2 — это схема вида

$$\begin{aligned} & ?[2] \text{ Question}[i] \rightarrow \text{FirstLink} \rightarrow \text{cAllContainers} \rightarrow \text{d[*]} \rightarrow \text{cSmthContainer} \rightarrow \\ & \quad \{1 \dots \text{ d[*]} \rightarrow \text{cSmthUnitsMany} \rightarrow \{2 \dots\}2 \\ & \quad \text{d[*] ChoiceUnitOfMany} \rightarrow \text{Name}[i]; \dots\}1. \end{aligned}$$

Здесь каждое `Name[i]` ($i = 0, \dots, N$) — это либо звено вида `uSmthUnicum`, либо звено вида `VarSmthName =`. В запросах типа 2 используются звенья вида `VarSmthName =` — имена переменных (неопределённых) схем, которые требуется определить в рамках данной среды радикалов.

Для ответа на запрос типа 2 будем использовать специализированный контейнер `cAnswers`, объединяющий все ответы `cAnswer`, полученные на запрос. Каждый из контейнеров `cAnswer` объединяет краткий (`d[*] Brief` → ...) и полный (`d[*] Full` → ...) ответы. Для краткого ответа используется контейнер переменных `cVars`, объединяющий связанные между собой имена переменных схем и конкретные схемы среды радикалов. Полный ответ, как и в случае ответа на запрос типа 1, содержит полные цепочки (пути), ведущие к искомым схемам.

Активация среды радикалов с помощью активаторов приводит к появлению новых контейнеров, которые затем могут быть добавлены к опорной среде радикалов. Будем говорить в таких случаях, что активаторы раскрывают среду радикалов.

Теорема об активаторах. Для любой нормализованной среды радикалов существует конечная последовательность запросов, с помощью которой активаторы полностью раскрывают исходную среду радикалов за конечное число шагов.

Литература

- [1] Соболева Т. С., Чечкин А. В. Дискретная математика. — М.: Академия, 2006.
- [2] Чечкин А. В., Пирогов М. В., Шундеев А. С. Слабоформальные системы // Обратные и некорректно поставленные задачи. VIII конференция. Тезисы докладов. МГУ им. М. В. Ломоносова. Факультет вычислительной математики и кибернетики. — М.: МАКС Пресс, 2003.
- [3] Чечкин А. В. Ультрамедиа — новое направление в развитии искусственного интеллекта // Интеллектуальные системы. — 1996. — Т. 1, вып. 1-4.

