

МЕТОД СИТУАЦИОННОГО АНАЛИЗА И ПРОЕКТИРОВАНИЯ МОДЕЛИ ПРЕДМЕТНОЙ ОБЛАСТИ ПРОИЗВОЛЬНОЙ ПРИРОДЫ

**Болотова Л.С.¹, Мороз Ю.В.²,
Смирнов С.С.³, Смольянинова В.А.⁴**

*(Московский государственный институт радиотехники,
электроники и автоматики (технический университет))*

Рассматривается метод ситуационного анализа и проектирования модели предметной области произвольной природы для экспертных систем поддержки принятия решений. В основу метода положены идеи ситуационного управления большими системами и концептуального анализа и проектирования систем организационного управления. Приводятся основные положения метода, формально-логический аппарат анализа концептуальных структур единичных решений и формирования концептуальной структуры всей предметной области.

Ключевые слова: экспертные системы, инженерия знаний, ситуационное управление, ситуационный анализ, модель предметной области, концептуальный анализ, проектирование, концептуализация, формализация, решение, действие, концептуальная структура, база знаний, система управления знаниями.

¹ Болотова Людмила Сергеевна, профессор каф. Вычислительной техники, доктор технических наук, профессор (lubolotova@mail.ru).

² Мороз Юлия Владимировна, аспирант (moroz@mirea.ru).

³ Смирнов Сергей Сергеевич, ст. преп. каф. Вычислительной техники (silicon_logic@rambler.ru).

⁴ Смольянинова Валерия Аполлоновна, ст. преп. каф. Математического обеспечения вычислительных систем (valerysmol@mail.ru).

1. Введение

Как известно, проблемы инженерии знаний возникают преимущественно на первых трех этапах технологии разработки экспертных систем (ЭС): *идентификация* → *концептуализация* → *формализация*. На этапах *реализации* и *тестирования* могут проявиться недостаточность, неполнота, некачественность описания элементов и предметной области (ПрО) в целом. В этих случаях осуществляется возврат на предыдущие этапы проектирования и это может происходить не единожды. При неопытных инженерах по знаниям хорошей базы знаний (БЗ) и, соответственно, ЭС, можно не получить никогда. Соответственно этим этапам, методы инженерии знаний тоже распределяются по группам применимости. Анализ этой богатой палитры показывает, что, несмотря на разнообразие методов (около 200 – по подсчетам Гавриловой Т.А. [8,9]), они в итоге не дают целостного видения самой модели ПрО. В этой связи проектирование модели ПрО до сих пор остается искусством. Мы считаем, что проблема инженерии знаний является фундаментальной и тормозит дальнейшее развитие ЭС, систем управления знаниями (СУЗ) и т.п. В этой связи мы предлагаем метод ситуационного анализа и проектирования модели ПрО произвольной природы, который, на наш взгляд позволяет снять эту проблему или существенно её ослабить, а в технологии разработки ЭС этапы *концептуализации* → *формализации* → *реализации* рассматривать как единый целостный процесс, в котором этапы формализации и реализации не разделяются. В его основу положены идеи метода ситуационного управления (Поспелов Д.А, Клыков Ю.И. [10,14,15,16]) и методологии концептуального анализа и проектирования (КАиП) систем организационного управления (Никаноров С.П. [11,13]). Напомним кратко их основное содержание.

2. Метод ситуационного управления

Базовые идеи метода: постепенное формирование модели внешнего мира в мозгу человека по мере приобретения им опыта (Пушкин В.Н.) [16]; семиотический подход к описанию сложных и больших систем управления (Шрейдер Ю.А., Скороходько Э.Ф.) [17,18,20,21]; принципы естественно-языкового описания модели ПрО, классификации ситуаций, обучения модели (Поспелов Д.А., Клыков Ю.И.) [10,14,15,16].

Первый принцип состоит в том, что в ситуационных моделях формализация объекта осуществляется на языке, близком к естественному. Основными, сохраняемыми знаками являются знаки двух видов: понятия и отношения. Понятия представляют собой знаки, которые ставятся в соответствие объектам (явлениям, событиям, ситуациям) внешнего мира, а отношения описываются знаками, которые соответствуют связям между ними. По аналогии с грамматикой естественного языка, в языке ситуационных моделей также задается ряд формальных правил порождения производных понятий и отношений.

Второй принцип метода базируется на идее классификации ситуаций. Предполагается, что мощность множества пространства ситуаций существенно больше множества возможных решений по управлению. Суть его в том, что при создании модели управления необходимо разбиение множества всех возможных ситуаций на классы таким образом, чтобы каждому можно было сопоставить либо решение по управлению, либо модель, обеспечивающую порождение допустимых рациональных решений. Выбор наилучшего решения из этого множества может происходить на основе либо эвристических процедур, либо известных точных математических моделей. Типичными примерами таких решений, например, являются: для управления движением на перекрестке – включить красный (зеленый) свет; для управления эскалаторами метро – запустить движение эскалатора вниз (вверх); для принятия решения банком о выдаче кому-то кредита – выдать кредит или нет; для предупреждения граждан о степени террористической напряженности в стране –

на соответствующих табло включить зеленый (красный, желтый и т.п.) цвет и т.п.

Каждый класс ситуаций описывается в обобщенном виде и отнесение конкретной ситуации к одному из классов в процессе принятия решений состоит в определении того обобщенного описания, частным случаем которого является эта ситуация. Таким образом, на основе принципа классификации в процессе принятия решений осуществляется порождение области поиска решений, удовлетворяющих целям функционирования системы. Это, в свою очередь, обеспечивает возможность решения задач управления с неопределенной областью поиска решений, что является принципиально важным и отличает метод от традиционных подходов. Как известно, любую задачу управления можно представить в виде графа, одна из вершин которого соответствует исходному состоянию объекта управления, а остальные – возможным его состояниям. Дуги интерпретируются как команды управления, переводящие объект из одного состояния в другое. Решение задачи состоит в нахождении такого пути на графе, по которому исходное состояние переводится в целевое оптимальным образом. Так могут быть поставлены и решены задачи, для которых граф управления известен заранее. В случаях, когда граф либо не известен, либо он слишком велик, традиционные методы недостаточны и необходим переход к методам решения задач с неизвестным графом, и метод ситуационного управления предлагает один из таких подходов.

Третий принцип метода предполагает наличие аппарата, позволяющего строить модель управления объектом на основе обучения этих моделей принятию решений. Обучение должно осуществляться либо экспертом, либо на основе опыта решения задач управления, накапливаемого ситуационной моделью управления в процессе своего функционирования. При этом обучение состоит в формировании самой модели объекта, а затем модели принятия решений по управлению. Этот принцип обеспечивает создание моделей, способных к усовершенствованию функций принятия решений, к адаптации в изменяющихся условиях работы. Это означает возможность расширения моде-

ли, ее функций и ассортимента решаемых задач, реализации принципов и «выращивания» модели для заданных условий.

Следует отметить, что вопросы инженерии знаний в ситуационном управлении практически не рассматривались и, вряд ли тогда осознавалась во всей их глубине и фундаментальности.

3. Методология концептуального анализа и проектирования

В ее основе лежит идея анализа и проектирования конструкции моделей ПрО и самих БЗ в виде системы инвариантных конструкторов – моделей и задания метода их интеграции в некоторую единую понятийную структуру. Вся работа с экспертами строится сообразно этим моделям. В этом случае возможны мощная инструментальная поддержка, организация процессов обучения, самообучения и развития. Условием решения проблемы данной сложности является *точное, технологичное, эффективное мышление субъектов* этой проблемы, свободное от традиций и познавательных предрассудков. Такое мышление создается путем *сознательного его строительства*, а не путем фиксации сложившихся стереотипов. Масштаб ПрО и глубина исследования не являются ограничением для такого мышления. Выходным продуктом такого технологичного мышления является стандартная по своей форме модель ПрО, основания и выводы которой непосредственно могут контролироваться. Все элементы системы ценностно ориентированы путем задания целей системы, которые явно формулируются. Поэтому язык всей системы естественным образом вытекает из используемых понятий, а не наоборот, когда понятия вытекают из сложившегося языка. Таким образом, работа с экспертом ведется в русле дисциплинированного мышления путем использования готовых интеллектуальных единиц, называемых в логике «конструктами». Модель ПрО, в результате, представляется в виде особым образом связанных конструкторов, их описаний и отношений между ними. При таком подходе появляются возможности сравнения конструкторов и построения более сложных из простых

средствами формальной логики и искусственного интеллекта, которым придается операциональная форма.

Таким образом, конструкт – это особое средство, полученное самим человеком (разработчиком или экспертом), проверенное им на практике, позволяющее ему воспринимать и понимать окружающую действительность, оценивать и прогнозировать события. Конструкты могут быть личностные (субъективные) и модельные. В первом случае набор конструктов выявляется из суждений эксперта, затем проверяется на инвариантность у других экспертов и утверждается в качестве некоторой эталонной структуры. Во втором случае конструкт проектируется исходя из поставленных целей и методов их достижения. Итоговым продуктом концептуального анализа является концептуальная модель предметной области (КМПрО), дающая *в контексте субъектных интересов* исчерпывающее описание ПрО. Возможность управления процессом формирования модели ПрО является важным достоинством КАиП.

Основные достоинства методологии КАиП.

Позволяет постепенно формировать «несущую конструкцию», образующую понятийный каркас того информационного пространства, в рамках которого будут осуществляться все основные процессы по обработке информации и выводу решений.

Организует работу экспертов, помогая им «выложить на лист» знание в конструктивном виде, которым они перед началом работы, вряд ли, обладали, т.к. экспертное знание описывается с помощью специальных конструктов - образцов (прототипов), из которых затем синтезируется ситуационное пространство ПрО.

4. Метод ситуационного анализа и проектирования модели предметной области

Метод ситуационного управления, как было показано в п.2, был разработан применительно к случаю, когда число возможных ситуаций в ПрО объекта управления существенно больше

числа решений, т.е., $S \gg U$. Однако, как показала практика, таких систем управления сравнительно немного. Для большинства вопрос о числе возможных решений и их соотношении с числом возможных ситуаций вообще является открытым. Поэтому можно говорить о случаях, когда число возможных ситуаций либо соизмеримо с числом решений, т.е. $S \approx U$, либо множество U достаточно велико и открыто, чтобы перечислить все возможные решения [1,4,5]. Например, в решении о выдаче кредита не ясными остаются вопросы: размер кредита, время выдачи, условия выдачи, срок кредита. Чтобы ответить на перечисленные вопросы, нужно их рассмотреть как самостоятельные задачи, требующие существенного развития модели ПрО и, соответственно, самого метода. Именно для этих случаев разработан предлагаемый метод.

4.1. ОСНОВНЫЕ КОНЦЕПТУАЛЬНЫЕ СХЕМЫ МЕТОДА

Исходное *решение* (decision – d) рассматривается как действие, имеющее следующую структуру: *субъект действия – действие – объект действия – компоненты действия, необходимые для реализации решения*. Например, для приведенного выше примера о выдаче кредита, эта структура выглядит так: субъект действия (банк) – действие (выдать) – объект действия (кредит) – компоненты действия, влияющие на решение (положение клиента: семья, здоровье, возраст, наличие собственности и т.п., экономическое положение России, положение в мире, оценка и прогноз устойчивости самого кредитора-банка и т.д.). Таким образом, первоначальное решение может быть развернуто в дерево решений, на каждом уровне которого нужны свои знания. Фактически мы сформировали для рассматриваемого решения метаструктуру вида (рис.1), где X_{as} – субъект действия (action subject – as), X_a – действие (action – a), X_{ao} – объект действия (action object – ao), $X_{ac_1}, X_{ac_2}, \dots, X_{ac_n}$ – компоненты действия (action components – ac). Субъект действия, объект действия и все компоненты действия связаны с самим действием X_a би-

▪ в любом решении по управлению само решение отождествляется с действием, у которого есть субъект и объект, описывается оно как императив в виде глагола, задаваемого в инфинитивной форме (сделать, выдать, построить и т.п.);

▪ решение выступает в роли целеобразующей составляющей задачи по управлению и в этом смысле дерево решений может отождествляться с деревом целей или задач;

▪ двухуровневый граф со структурой (рис.1а) является инвариантом по отношению к произвольной ПрО.

Следуя методу КАиП, структура решения (рис.1а) является искусственной метаструктурой, удобной для организации направленного процесса проектирования, т.е. конструктором, притом сложным, составленным из простых (терминальных) конструкторов типа: субъект действия - действие (action subject - action (as-a)), действие - объект действия (action - action object (a-ao)), действие - компонента действия (action - action component (a-ac)). Назовем этот конструктор *концептуальной структурой единичного решения* (КСЕР). Множество КСЕРов обозначим через

$$D = \{d_1, d_2, \dots, d_m\}, \quad d_i = z_i \cup q_i, \quad i = \overline{1, m},$$

Таким образом, вершины d_i идентифицируют границы ПрО, необходимой для вывода данного решения и проектирования соответствующей модели.

4.2 СТРУКТУРНЫЕ И СЕМАНТИЧЕСКИЕ ОТНОШЕНИЯ

Очевидно, что все вершины КСЕР, кроме вершины действия, могут быть связаны между собой пространственными, логическими, другими типами отношений (как вертикальными, так и горизонтальными), образующими, в свою очередь, простые конструкторы типа: *as-ao*, *as-ac*, *ao-ac*, *ac-ac* со своими двунаправленными бинарными семантическими отношениями - R_{sem} между входящими в них вершинами. Тогда КСЕР превращается в структуру, представленную на рис.1б.

Здесь показан пример, когда между субъектом и объектом, субъектом и компонентами существует полное множество вертикальных, а между объектом и компонентами, компонентами и

компонентами – горизонтальных отношений. Число возможных семантических отношений R_{sem} как горизонтальных, так и вертикальных в такой структуре оценивается как: $N_{R_{sem}}(d) = n(n+1) + 2(n+1) = n^2 + 3n + 2$, где n – число компонент действия. При этом здесь не учитывались структурные отношения $N_{R_{str}}(d) = n + 2$. С их учетом получим: $N_R(d) = n^2 + 4n + 4$. Очевидно, что отношений в любой реальной структуре существенно меньше. Поэтому дальнейшее изложение будем иллюстрировать простыми структурами с двумя компонентами действия, четырьмя структурными и двумя семантическими отношениями (рис.2). На общность выводов это не повлияет.

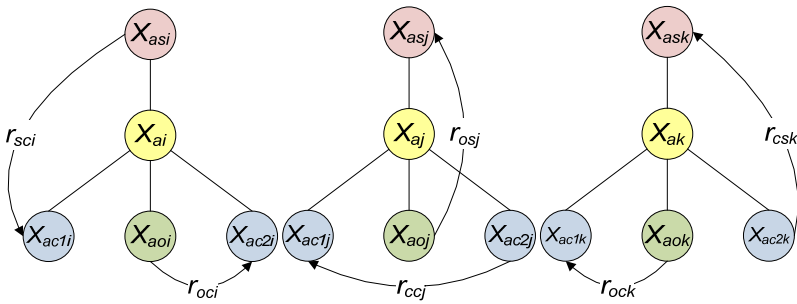


Рис. 2

Теперь КСЕР представляется в виде $d_i = \{X_{d_i}, R_{str d_i}, R_{sem d_i}\}$.

Другими словами, мы получили структуру, состоящую из множества нужных для вывода данного решения-действия имен объектов ПрО и имен отношений между ними.

Известно, что любое отношение может принимать значения – истина (И), ложь (Л), т.е., либо быть, либо нет. Оно устанавливается как истина, если выполняются определенные условия (одно или несколько одновременно). Сами условия формируются на основе отношений между значениями свойств объектов (например, отношений $=$, $\leq$, $\geq$, $\approx$, $\equiv$, $\neq$ и других), обычно пред-

ставляемых в виде левой части продукционных правил, а также проверки истинности других отношений в данной структуре. После выявления необходимых свойств (property – p) объектов ПрО X_{d_i} , единичное решение превращается в структуру, представленную на рис.3 (a,b). На рис. 3а показано, что все элементы КСЕР d_i обладают своим набором свойств $P(d_i)$, образующих множество её концептов – $C(d_i)$. Но, между свойствами концептов объектов X_{d_i} могут существовать свои отношения $R_p(d_i)$, например, отношения $=, \leq, \geq, \approx, \equiv, \neq$ и другие. При этом истинность этих отношений является условием истинности отношений между объектами. (Например, если $(p_{s_i}(X_{as_i}) \neq p_{c_{1i}}(X_{ac_{1i}})) = И$, то $\overline{РЯДОМ}(X_{as_i}, X_{ac_{1i}}) = И$, или $((p_{s_i}(X_{as_i}) \neq p_{c_{1i}}(X_{ac_{1i}})) = И) \rightarrow (\overline{РЯДОМ}(X_{as_i}, X_{ac_{1i}}) = И)$. Читается это, например, так: если координаты субъекта действия X_{as_i} не равны координатам первой компоненты действия $X_{ac_{1i}}$, то X_{as_i} и $X_{ac_{1i}}$ находятся *НЕ РЯДОМ*). В качестве примера на рис.3б показана ситуация:

$$\begin{aligned} &((p_{s_i}(X_{as_i}) \neq p_{c_{1i}}(X_{ac_{1i}})) = И) \rightarrow (r_{sci}(X_{as_i}, X_{ac_{1i}}) = И) \text{ и} \\ &((p_{oi}(X_{aoi}) < p_{c_{2i}}(X_{ac_{2i}})) = И) \rightarrow (r_{oci}(X_{aoi}, X_{ac_{2i}}) = И). \end{aligned}$$

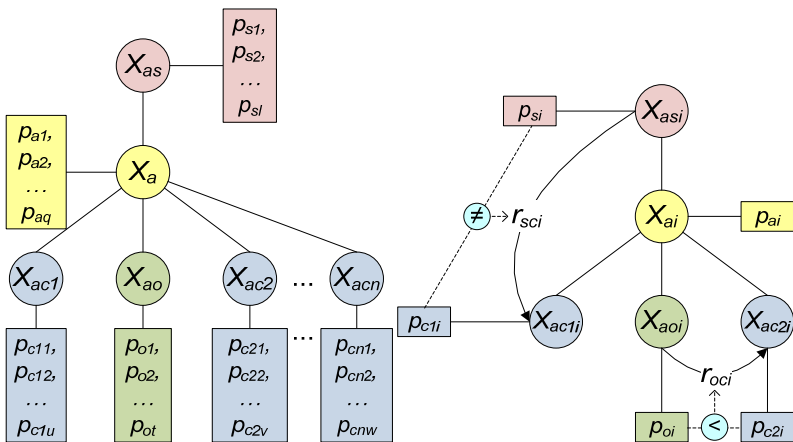


Рис. 3 (a, b)

Соответственно, модель фрагмента ПрО единичного решения расширяется за счет добавления множества свойств объектов и отношений между ними, т.е.:

$$C(d_i) = \{X_{d_i}, R_{str d_i}, R_{sem d_i}, P_{d_i}, R_{p d_i}\}, \quad i = \overline{1, m},$$

где: P_{d_i} – свойства всех объектов названных в КСЕР d_i , $R_{p d_i}$ – отношения между свойствами объектов d_i .

На основе каждой d_i можно рассчитать мощность множества возможных ситуаций для принятия решения, определяемых данной структурой и его возможных вариантов из допустимых сочетаний свойств, отношений и их значений: $N_{d_i} = k_1 \cdot k_2 \cdot \dots \cdot k_s \cdot 2^t$, где $k_1 \cdot k_2 \cdot \dots \cdot k_s$ – число возможных значений каждого свойства, t – число отношений, участвующих в данной структуре. Например, если $k_1(p_1) = 3$, $k_2(p_2) = 3$, $k_3(p_3) = 2$, $k_4(p_4) = 4$, а число отношений в структуре равно, например, 3, то $N_{d_i} = 3 \cdot 3 \cdot 2 \cdot 4 \cdot 2^3 = 576$ возможных ситуаций на одной структуре d_i (рис.4).



На основе КСЕР d_i генерируется БЗ в виде отображения $F_{k_i} : S_{d_i} \rightarrow X_{a_i}$, т.е. множества прототипов ситуаций или правил, которым соответствует решение-действие X_{a_i} данной d_i . При этом, в зависимости от требований к проектируемой ЭС (статическая, динамическая, прогнозирующая) могут генерироваться правила различных типов: логико-трансформационные для преобразования ситуаций-предусловий в соответствии с принятыми решениями – в ЭС с обратной связью, для экстраполяции ситуаций-постусловий на требуемый период времени – в динамических ЭС [1]. Таким образом, на этом этапе КСЕР d_i превращается в полноценную концептуальную модель предметной области единичного решения (КМПРО ЕР): $C(d_i)$. Здесь под $C(d_i)$ понимается концепт КСЕР – d_i .

13

$C(D) = \bigcup_{i=1}^n C(d_i)$. Однако, это объединение не механическое, т.к.

над d_i выполняется ряд операций аналогичных теоретико-множественным: пересечения, включения, дополнения, разности, объединения и др. Кроме того, между КСЕРами может существовать отношение части и целого в случаях, когда единичное решение может быть развернуто в дерево решений, аналогичное дереву разбиения задачи на подзадачи. Графически такое решение описывается в виде графа редукции (ИИЛИ-графа), вершины которого отождествляются с именами решений – X_{a_i} ($i = \overline{1, n}$), а дуги – с отношением *часть – целое* (*part of while – r_{pow}*).

VII. Перечисленные операции определяются следующим образом.

Разность КСЕР d_i, d_j определяется как $d_{ij} = d_j \setminus d_i$, т.е. та часть d_j , которая отсутствует в d_i , $i, j = \overline{1, m}$.

Объединение КСЕР d_i, d_j – концептуальная структура решения (КСР) $d_{ij} = d_i \cup (d_j / d_i)$, $i, j = \overline{1, m}$.

Пересечение КСЕР d_i, d_j – концептуальная структура решения (КСР) $d_{ij} = d_i \cap d_j$, $i, j = \overline{1, m}$. При этом возможны пересечения следующих типов:

1. $d_i \cap d_j = \emptyset$. В этом случае КМ d_{ij} является простым объединением двух структур (рис.3), т.е.: $C(d_{ij}) = C(d_i) \cup C(d_j)$, $i, j = \overline{1, m}$.

2. При $d_i \cap d_j \neq \emptyset$ возможны следующие варианты.

2.1. Пересечение по одному типу вершин.

2.1.1. Пересечение по субъекту действия: $d_i \cap d_j = X_{as_{ij}}$, $i, j = \overline{1, m}$.

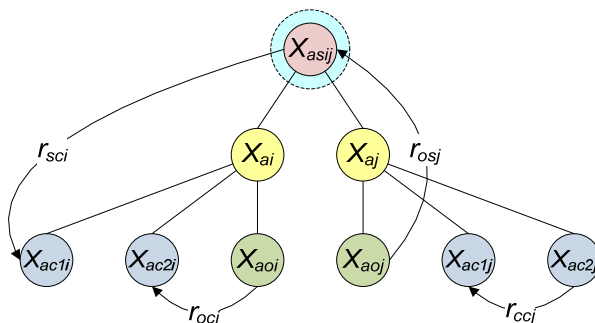


Рис. 5

В этом случае один и тот же субъект совершает разные действия над разными объектами, используя при этом различные компоненты. Это нормально.

2.1.2. Пересечение по самому действию: $d_i \cap d_j = X_{a_{ij}}$, $i, j = \overline{1, m}$.

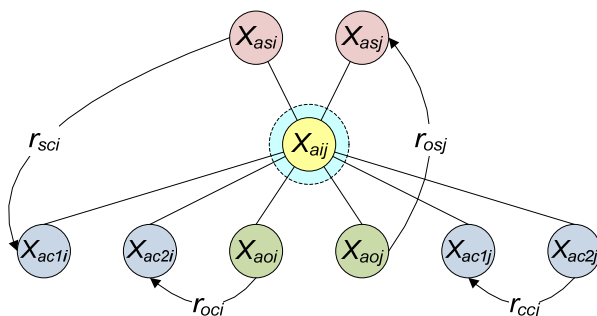


Рис. 6

Здесь очевидно противоречие: одно и то же действие выполняется различными субъектами, в роли которых могут выступать люди, автоматы, программы и т.п. При этом у них объекты и компоненты действий не пересекаются. Это должно насторожить инженера по знаниям. Необходимо дополнительное исследование, в чем тут дело: либо все правильно, но тогда

нужно устранять организационную неувязку уже в самой системе управления (очевидно не инженером по знаниям). И его задача – довести эту информацию до руководства. Либо, во втором случае, имеется путаница в названиях (терминах) соответствующих объектов и её надо тоже устранять.

2.1.3. Пересечение по объекту действия: $d_i \cap d_j = X_{aoij}$, $i, j = \overline{1, m}$.

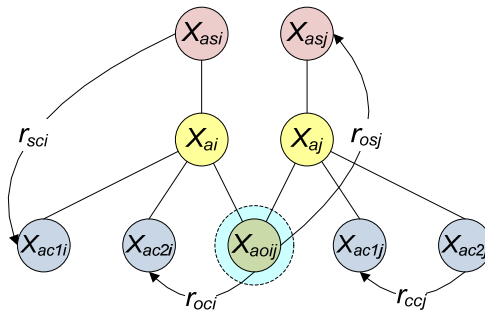


Рис. 7

В этом случае над одним и тем же объектом выполняется два, возможно более двух действий. Это нормально.

2.1.4. Пересечение по одной или нескольким компонентам действия: $d_i \cap d_j = X_{c_{1ij}} \wedge X_{c_{2ij}}$, $i, j = \overline{1, m}$.

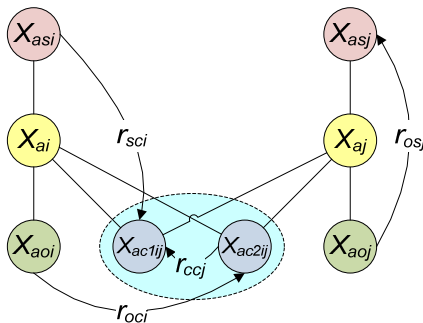


Рис. 8

Здесь показан случай, когда некоторые компоненты участвуют в разных действиях в разных ролях.

2.2. Пересечение по двум типам вершин.

2.2.1. Пересечение по субъекту действия и самому действию: $d_i \cap d_j = X_{as_{ij}} \wedge X_{a_{ij}}$, $i, j = \overline{1, m}$.

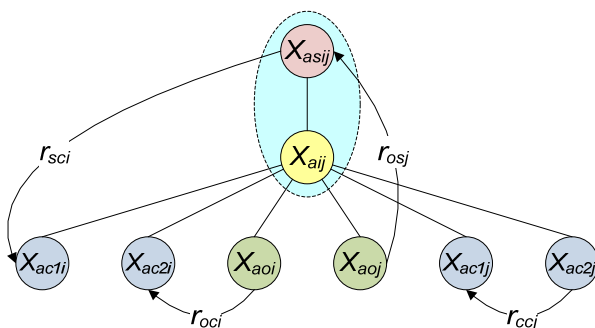


Рис. 9

Т.е., одно и то же действие имеет совершенно разную структуру у одного и того же субъекта его выполнения. По меньшей мере, это выглядит странно и, следовательно, требует дополнительной доработки и устранения неточностей, либо действительно одно и то же действие может выполняться совершенно по-разному, с помощью разных средств, что нормально, но тогда необходимо решать вопрос о приоритетности способов реализации этого действия.

2.2.2 Пересечение по субъекту действия и объекту действия: $d_i \cap d_j = X_{as_{ij}} \wedge X_{ao_{ij}}$, $i, j = \overline{1, m}$.

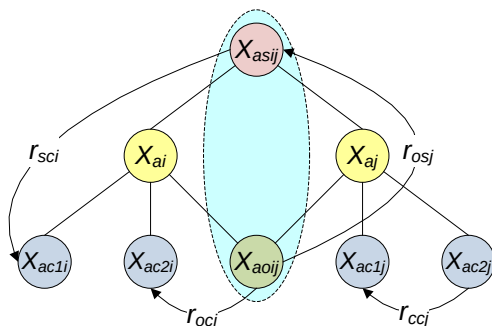


Рис. 10

Здесь - два разных действия выполняются над одним и тем же объектом, что вполне нормально, но возникает дополнительный вопрос, в каком порядке они должны выполняться (ИЛИЛИ): или они являются альтернативными. В последнем случае необходимо фиксировать знание о приоритетности направления выбора. Если это все-таки альтернатива, это значит, что на уровне субъекта действия между дугами существует отношение ИЛИ (рис.11).

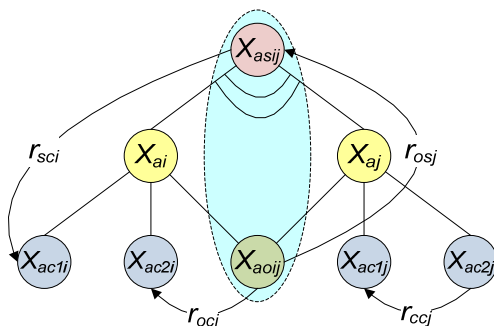


Рис. 11

2.2.3 Пересечение по субъекту действия и одной или несколькими компонентами действия: $d_i \cap d_j = X_{as_{ij}} \wedge X_{c_{1ij}} \wedge X_{c_{2ij}}$, $i, j = \overline{1, m}$.

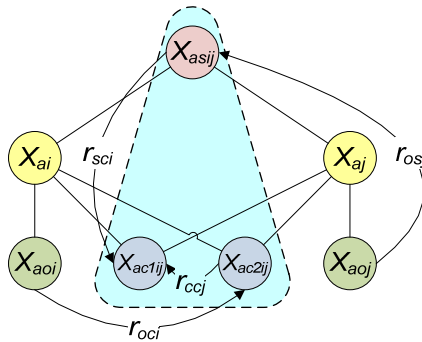


Рис. 12

В этой структуре явных противоречий нет.

2.2.4. Пересечение по самому действию и объекту действия: $d_i \cap d_j = X_{a_{ij}} \wedge X_{ao_{ij}}$, $i, j = \overline{1, m}$.

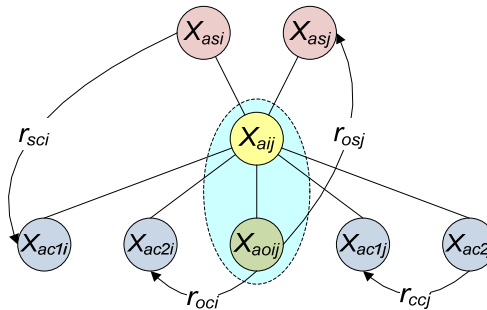


Рис. 13

Данная структура противоречива, т.к. два разных субъекта выполняют одно и то же действие над одним и тем же объектом, но разными средствами. Нужно искать ошибку. Она – либо на уровне субъектов (организационная путаница), либо на уровне объектов (их нужно по-разному назвать), либо разделить действие над ними на два.

2.2.5. Пересечение по самому действию и одной или несколькими компонентам действия: $d_i \cap d_j = X_{a_{ij}} \wedge X_{ac1_{ij}} \wedge X_{ac2_{ij}}$, $i, j = \overline{1, m}$.

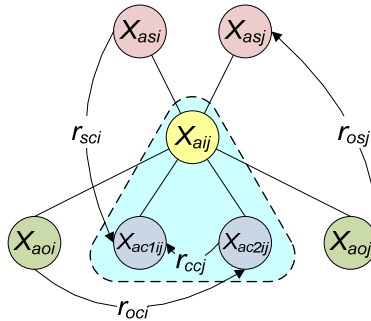


Рис. 14

Эта ситуация комментируется следующим образом: два разных субъекта выполняют одно и то же действие над разными объектами с применением одних и тех же средств, что вполне нормально.

2.2.6. Пересечение по объекту действия и одной или несколькими компонентам действия: $d_i \cap d_j = X_{ao_{ij}} \wedge X_{c1_{ij}} \wedge X_{c2_{ij}}$, $i, j = \overline{1, m}$.

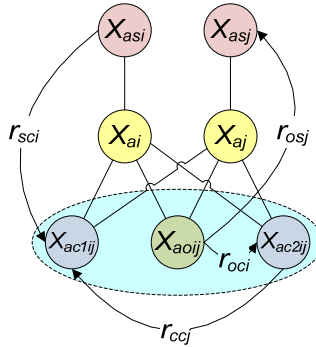


Рис. 15

Здесь ситуация похожа на предыдущую: разные субъекты выполняют разные действия над одним и тем же объектом с применением одних и тех же средств (одного или нескольких), что вполне нормально, но тогда возникает проблема очередности в выполнении действий и использовании требуемых средств.

2.3. Пересечение по трем типам вершин.

2.3.1. Пересечение по субъекту действия, самому действию и объекту действия: $d_i \cap d_j = X_{as_{ij}} \wedge X_{a_{ij}} \wedge X_{ao_{ij}}$, $i, j = \overline{1, m}$.

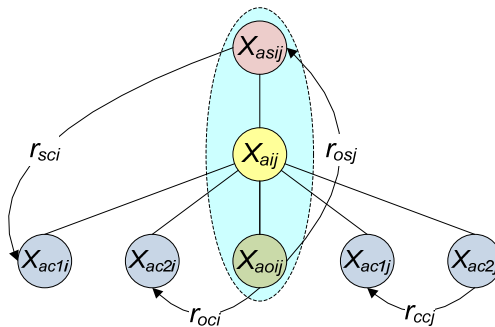


Рис. 16

Здесь комментарий прост: один и тот же субъект выполняет одно действие над одним объектом, но средства описания дейст-

вия оказались разными у тех экспертов, которые строили эти структуры. Здесь надо разбираться с экспертами: либо они что-то напутали, либо один из них дал более подробную структуру и тогда все нормально.

2.3.2. Пересечение по субъекту действия, самому действию и одной или нескольким компонентам действия:

$$d_i \cap d_j = X_{as_{ij}} \wedge X_{a_{ij}} \wedge X_{c_{1ij}} \wedge X_{c_{2ij}}, \quad i, j = \overline{1, m}.$$

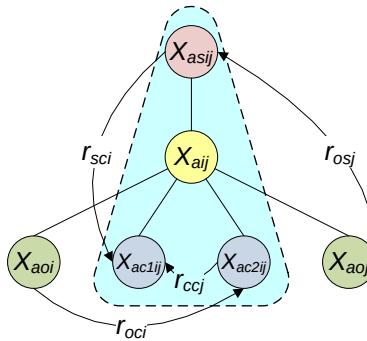


Рис. 17

2.3.3. Пересечение по субъекту действия, объекту действия и одной или нескольким компонентам действия:

$$d_i \cap d_j = X_{as_{ij}} \wedge X_{ao_{ij}} \wedge X_{c_{1ij}} \wedge X_{c_{2ij}}, \quad i, j = \overline{1, m}.$$



2.3.4. Пересечение по самому действию, объекту действия и одной или нескольким компонентам действия:

$$d_i \cap d_j = X_{a_{ij}} \wedge X_{ao_{ij}} \wedge X_{ac1_{ij}} \wedge X_{ac2_{ij}}, \quad i, j = \overline{1, m}.$$


23

ем части или всех одинаковых средств. Очевидна противоречивость этой структуры и здесь должна существовать организационная неувязка. Кроме того, различные эксперты, видимо, строили эти структуры и у них возникли противоречия (если не все средства совпали) в определении самих структур. Здесь тоже нужно дополнительно разбираться.

Включение d_i, d_j друг в друга (частичное или полное):

$d_{ij} = d_i \subseteq d_j, i, j = \overline{1, m}$ (рис.8). При полном совпадении объектов и компонент действия, следует дополнительно рассмотреть их имена. Не являются ли они синонимами одного и того же? Если являются, то одна из структур убирается из списка D . При частичном совпадении итоговая структура d_{ij} будет иметь вид 1.2.4 (рис.20).

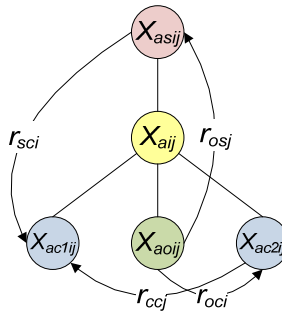


Рис. 20

Остальные операции определяются аналогично.

Операции пересечения, объединения КСЕР d_i, d_j, d_k коммутативны и ассоциативны, но не дистрибутивны. Операция разности КСЕР d_i, d_j не коммутативна и не ассоциативна:

$$\begin{aligned}
d_i \cap d_j &= d_j \cap d_i, \\
(d_i \cap d_j) \cap d_k &= d_i \cap (d_j \cap d_k) \\
d_i \cup d_j &= d_j \cup d_i, \\
(d_i \cup d_j) \cup d_k &= d_i \cup (d_j \cup d_k), \\
d_i \cup (d_j \cap d_k) &\neq (d_i \cup d_j) \cap (d_i \cup d_k), \\
d_i \cap (d_j \cup d_k) &\neq (d_i \cap d_j) \cup (d_i \cap d_k), \\
d_i / d_j &\neq d_j / d_i, \\
(d_i / d_j) / d_k &\neq d_i / (d_j / d_k).
\end{aligned}$$

Далее формируются полные описания всех элементов структур действий (КСЕРов) в виде понятийных структур (концептов) субъектов, объектов и компонент действий, самих структур действий и полной КСР путем выполнения над ними операций ($\cup, \cap, /$).

Очевидно, что результаты операции объединения концептов действий определяются типом пересечения соответствующих им структур.

Так для ситуации, когда $d_i \cap d_j = \emptyset$, d_i , d_j не зависимы, их КМ $C(d_{ij})$ является простым объединением двух структур (рис.5): $C(d_{ij}) = C(d_i) \cup C(d_j)$, $i, j = \overline{1, m}$.

Для ситуаций типов 2.1.1-2.1.4 при формировании КМ d_{ij} концепты всех вершин КСЕР d_{ij} наращиваются (содержание $C(d_{ij})$) разностью в концептах d_i , d_j , путем добавления к концептам соответствующих объектов новых свойств и отношений. Например: для случая, когда в пересечение попадает вершина «субъект действия», концепт этой вершины представляется, как: $C(X_{as_{ij}}) = C(X_{as_i}) \cup (C(X_{as_j}) \setminus C(X_{as_i}))$, $i, j = \overline{1, m}$, где

$C(X_{as_j}) \setminus C(X_{as_i})$ понимается как разность контекстов объектов X_{as_j} , X_{as_i} , т.е., в итоговую модель добавляется та часть содержания X_{as_j} , которая отсутствует в X_{as_i} .

Одна или несколько компонент действия d_i могут сами являться объектами одного или нескольких действий в других КСЕРах, т.е. объекты-компоненты могут играть одновременно в данной ПрО несколько ролей и для каждой из них важны разные свойства и отношения, т.е. у них должны быть разные концепты. При этом, действие X_{a_j} , очевидно, не может быть выполнено, пока не будет выполнено требуемое действие над компонентой $X_{ac_{2i}}$ действия X_{a_i} . Это означает, что между действиями структур d_i , d_j автоматически проявляется неявное отношение «действие–поддействие» (рис.21). Если таких компонент несколько (рис.22,23), то вершины-действия структур d_j , d_k ($j, k = \overline{1, m}$), связаны отношением редукции *ИЛИЛИ*. Для случая *ИЛИ* необходимо устанавливать приоритеты выполнения альтернативных действий. Для случая *И* вопросы пересечения структур d_j , d_k , рассматриваются аналогичным 2.1.1-2.1.7 образом.

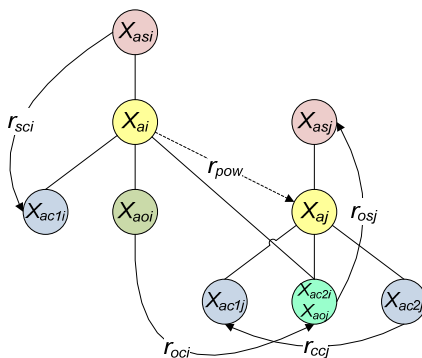


Рис. 21

Объектами действия могут быть одна или несколько компонент действия одновременно. Рассмотрим их взаимодействие, предположив, что две компоненты действия d_i являются, в свою очередь объектами действий d_j , d_k (рис.22).

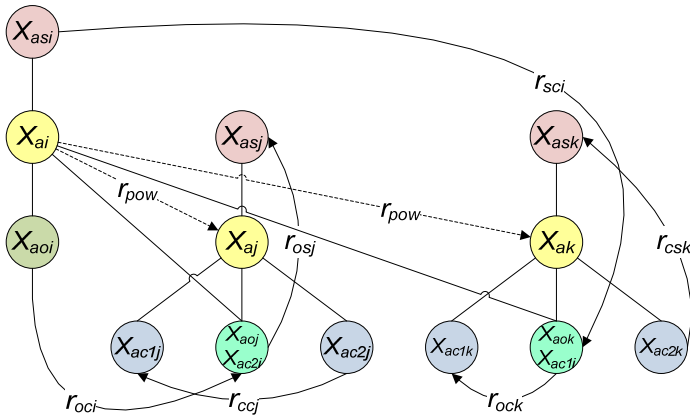


Рис. 22

Возможны следующие варианты взаимодействия разноразрядных структур.

1. Субъекты действия в структурах d_i , d_j не совпадают, т.е., $X_{as_i} \neq X_{as_j}$ (рис.23). Это означает, что эти действия выполняются разными субъектами, между которыми должно существовать отношение организационной иерархии (подчинения). Если этого нет, то и не может быть уверенности в выполнении данного действия. Следовательно, требование наличия организационного подчинения проявляется автоматически. И если его на самом деле нет, то его необходимо ввести в какой-то форме. Будем интерпретировать это отношение как отношение «часть-целое» (r_{pow}). Допустим также, что для X_{as_i} и X_{as_j} построены

соответствующие КСЕРы d_i , d_j и их концептуальные модели:

$C(d_i)$, $C(d_j)$. Тогда КМ d_{ij} есть:

$$C(d_{ij}) = C(d_i) \cup C(d_j) \wedge r_{pow}(X_{asi}, X_{asj}) \wedge r_{pow}(X_{aoi}, X_{aoj}),$$

$i, j = \overline{1, m}$.

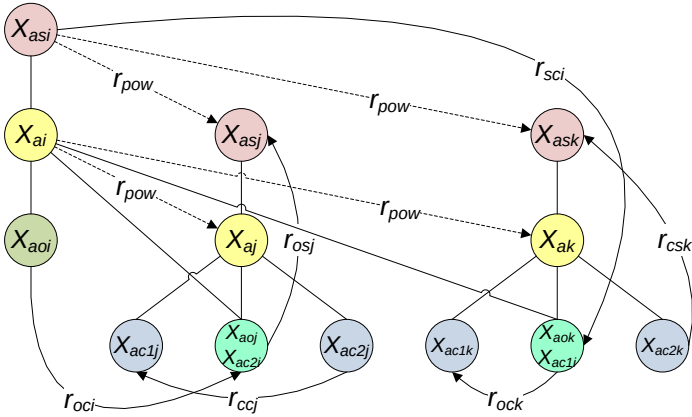


Рис. 23

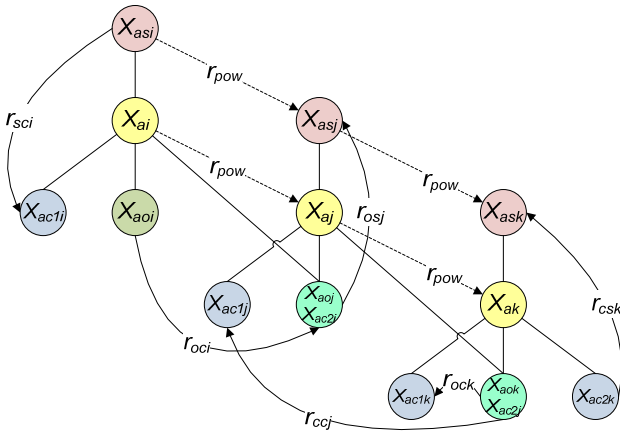


Рис. 24

2. При совпадении субъектов действия в структурах d_i, d_j , т.е. когда $X_{as_i} = X_{as_j}$ (рис.25), необходимо рассматривать цепцы X_{as_i} и X_{as_j} на предмет унификации.

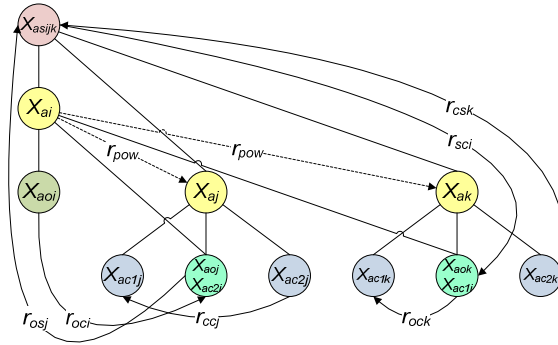


Рис. 25

Цепочка вложенных КСЕРов, образующаяся по отношениям *действие-поддействие, субъект-надсубъект, объект-подобъект* может продолжаться до тех пор, пока не будет достигнут уровня терминальных действий-решений, т.е. элементарных действий. В результате мы получим концептуальную структуру решений всей ПрО (КСРПрО) вида (1):

$$D = \bigcup_{i=1}^m d_i \cup G(X_{as}, R_{pow}) \cup G(X_a, R_{pow}) \cup G(X_{ao}, R_{pow}),$$

где $G(X_{as}, R_{pow})$, $G(X_a, R_{pow})$, $G(X_{ao}, R_{pow})$ – графы-деревья иерархии: субъектов действий (субъектный уровень описания); самих действий (операциональный уровень описания); объектов (объектный уровень описания), соответственно.

КСР всей ПрО, таким образом, представляется в виде трех-слойного графа: первый слой – иерархия субъектов действий; второй слой – иерархия действий, третий слой описывает уровень взаимодействия объектов, компонентов действий и отношений между ними.

Именно он определяет структуру ситуационного пространства решений. На рис. 20 КСР ПрО представлены эти три уровня в цветах – розовый, желтый и зеленый, соответственно.

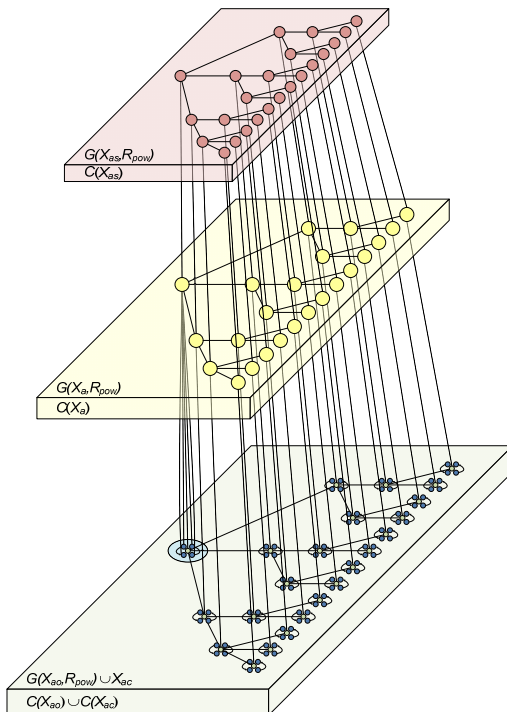


Рисунок 26

На этом рисунке: «цветок»  изображает «юбку» d_i (семантические отношения между объектами и компонентами действий); пунктирными вертикальными линиями показаны структурные отношения между парами: $(X_{as} - X_a), (X_a - X_{ao}), (X_a - X_{ac})$; пунктирные круги вокруг «цветков» – группы семантических отношений в рамках каждой d_i (для упрощения рисунка показан только один такой «цветок», а они должны быть вокруг всех вершин типа X_{oc}); сплошными

линиями показаны отношения иерархии на уровне субъектов действия, действий и компонент действий.

Соответственно, КМПРО описывается выражением (2):

$$G(D) = \bigcup_{i=1}^m C(d_i) \cup G(X_{as}, R_{pow}) \cup G(X_a, R_{pow}) \cup G(X_{ao}, R_{pow}),$$

где $\bigcup_{i=1}^m C(d_i)$ – отображение $F_D : S_{d_i} \rightarrow U_{X_{a_j}} \ (i, j = \overline{1, m})$ или БЗ

данной ПРО, остальные члены – иерархии субъектов действий, действий и объектов действий.

Граф КСР ($G(D)$), образующийся в результате сопоставления действиям соответствующих КСЕР, состоящих из задачи управления (функциональной части), и обеспечивающей части и их связывания через отношения иерархии, по существу, задает все множество путей (или сценариев) многошагового вывода решений. Вершины третьего типа на своем уровне связываются как отношениями иерархии, так и семантическими отношениями типа: место, средство, способ и т.п. Назовем первый уровень графа – субъектным, второй – операциональным, а третий – объектным. Каждый из этих слоев описывается своим графом: первый и второй уровни имеют один и тот же вид – вид дерева и различаются лишь интерпретацией вершин; третий является ориентированным графом, в котором каждая дуга имеет своё имя. Операциональный граф даёт полное представление об иерархии действий, их перечне на каждом уровне принятия решений и соответствует классификатору действий, описываемому в виде: $D = (X_a, r_{pow})$, где X_a – множество действий; r_{pow} – отношение «часть-целое», интерпретируемое здесь как отношение действие-поддействие. Число уровней классификатора действий зависит от глубины конкретизации, необходимой для достижения цели, определяемой корнем дерева, и достигнутой в исследуемой ПРО степени автоматизации процессов управления. Очевидно, что чем он выше, тем этих уровней меньше.

Объектный граф представляет собой семантическую сеть со структурными и семантическими отношениями. По отношениям

между вершинами сети всегда можно установить все роли, в которых может выступать каждый объект. Очевидно, что каждой вершине классификатора действий (X_{a_k}) соответствует свой подграф вывода – $G_k(d_k)$, ($k = \overline{1, m}$), который, по существу, образует сценарий вывода решения, содержащий КСЕРы всех действий, лежащих на пути к целевой вершине. Таким образом, граф $G(D)$ можно рассматривать как множество возможных сценариев вывода алгоритмов решений $G_k(d_k)$, – соответствующих d_k -ой вершине КСР $G(D)$. Порядок вывода может быть как сверху-вниз (обратная стратегия вывода «от цели - к ситуации»), так и снизу-вверх (прямая стратегия вывода «от ситуации - к цели») и регламентируется классификатором действий.

Например, в сценарии задачи обработки роботом заготовки на станке предполагается, что на вход модели поступают сообщения о состоянии объектов роботизированного участка цеха. Распознавание состояний станков, тары, манипулятора является самостоятельными задачами, решаемыми на основе показаний различных датчиков информации, устанавливаемых на этих объектах. Если принять подобное понимание, то вывод решений по управлению участком становится двухуровневой задачей, причём задача «обработка заготовки» становится частью общей задачи распознавания ситуации на участке. В связи с тем, что регистрация информации на объектах участка ведётся автономными средствами, действия, соответствующие задачам распознавания, не пересекаются. Таким образом, каждая компонента КСЕР «обработка заготовки» в свою очередь, является объектом действия другой операции – распознавания (рис. 20). Можно представить и более глубокий уровень принятия решений в случае, например, когда распознавание состояния ведётся не на уровне всего станка сразу, а на уровне его отдельных узлов, и затем станка в целом.

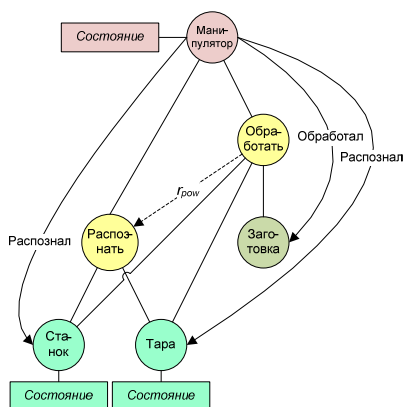


Рис. 27

Построением графа $G(D)$ завершается процесс формирования общей структуры ситуационного пространства ПрО, в котором есть все необходимые знания для принятия решений на любом уровне конкретизации. Реализация сборки графа $G(D)$ выполняется на основе инструментальной программной системы «Помощник эксперта» [6,7]. В результате логической обработки и формирования модели ПрО на выходе выдается граф $G(D)$ в виде понятийно-объектной модели (ПОМ), в которой описываются все его элементы и их концепты с указанием иерархии, т.е. понятий-надпонятий, типов данных, области значений и т.д.

Отметим возможные особенности элементов знания (свойств, отношений между различными структурными единицами $G(D)$).

Каждое свойство задается своим типом данных (вещественный, целый, логический, строковый, символьный и т.д.) и множеством значений. Например, объект (*яблоко*) определяется свойствами (сорт (тип данных – строковый), множество значений – (антоновка, мельба, белый налив и т.п.), вкус (тип данных – строковый), множество значений – (сладкий, кислый, кисло – сладкий и т.п.)), вес (тип данных – вещественный, область зна-

чений – (от 100 до 400 граммов)). Однако, возможно определение свойств в виде лингвистических переменных с соответствующим определением функций принадлежности и нечетких множеств.

Точно также отношения могут задаваться своими значениями (истина, ложь), либо как нечеткие с соответствующими операциями над ними, Т.о., возможно построение нечеткой модели ПрО или модели смешанного типа.

Каждый объект внешнего мира представляется в $G(D)$ именем (знаком), концептом, т.е. содержанием (множеством его существенных свойств объекта с этим именем) и денотатом (объемом этого знака), т.е. множеством своих экземпляров. Каждый экземпляр, в свою очередь, может иметь свои особенности, не позволяющие, например, передавать все его свойства по наследованию (например, пингвин – птица, но он не летает). Такого рода данные определяются на последнем этапе формирования системы знаний о ПрО и учитываются инструментальной программной системой при их внесении.

Далее с помощью встроенной программы осуществляется автоматическая генерация продукционных правил БЗ. Другими словами, вся модель ПрО представляется в другой форме – в виде правил-продукций, с которыми далее на этапе эксплуатации работает инструментальная программная система «Решатель Проблем» [7,19]. Однако, такая форма не единственная. При наличии, например, инструментальной системы, работающей с фреймами или с семантическими сетями, ПОМ дает для этого сеть фреймов. Если внимательно посмотреть на приведенный фрагмент ПОМ, то станет понятно, что итоговая модель $G(D)$ представляется в виде гибридной модели продукционно-семантико-фреймового типа, легко трансформируемого в любую нужную модель представления знаний.

Подчеркнем еще один важный момент. Описание метода ситуационного анализа начиналось с определения понятия конструкта, введения простых и составных конструктов, ставших затем «кирпичиками» для всей структуры ситуационного

пространства ПрО и её модели. Кстати, при таком подходе правила, фреймы, семантическая сеть, а также иерархии по отношениям «часть – целое» (r_{pow}), «класс – экземпляр класса» (IsA – Is A), «вид - род» (АКО – A Kind Of), также являются составными конструктами. При этом проектирование осуществляется по четкому и жесткому, но инвариантному сценарию, который, благодаря этому, позволяет создать необходимую технологию и соответствующую инструментальную программную поддержку. При этом метод гарантирует получение качественного результата на выходе в виде БЗ ПрО, а также концептуальной структуры пространства вывода решений. В эту технологию естественным образом вплетаются знания экспертов или их групп, регламенты субъектов управления, знания из текстов и т.п. Очевидно также, что сам процесс проектирования может проходить параллельно (деление процесса проектирования по вертикали, по горизонтали, по кустам и т.д.)

5. Вопросы инженерии знаний

Рассмотрим проблемы получения знаний на разных этапах проектирования и методы инженерии знаний, которые при этом могут быть наиболее эффективными.

1. Любой конструкт по определению является концептуальной схемой, поскольку работает на уровне классов объектов и отношений между ними. Основной метод вывода решений – дедуктивный. Это предъявляет высокие требования к качеству (уровню) мышления инженеров по знаниям (инженеров – аналитиков). Т.е., требуется наличие у инженеров системного и концептуального мышления, которое формируется на основе большого практического опыта проектирования СППР для действительно сложных систем управления и под руководством очень опытных специалистов.

Ни того, ни другого у нас в стране нет. О причинах не стоит говорить, они всем понятны. Самое плохое, что факт необходимости осознанного и нацеленного обучения такому мышлению мало осознается. Решение видится в организации

обучения, параллельно нацеленного на формирование навыков системного и концептуального мышления. И данный метод дает прекрасную основу для этого.

2. В настоящее время возникли системы управления нового типа, обладающие сложной структурой: многофункциональных иерархических, локально и глобально распределенных, корпоративных. Именно для них особенно нужны ЭС ППР. Но необходимых знаний практически нет, поскольку системы новые и необходимая статистика и опыт еще не осознаны и не накоплены. Соответственно, нужного качества экспертов тоже очень мало. Однако они могут быть весьма полезными при соответствующей технологии и технике работы с ними, в режиме не только извлечения знаний, но и их проектирования. Выход видится в подходе к проблеме БЗ, как к проектированию практически с **чистого листа**.

Ни того, ни другого у нас в стране нет. О причинах не стоит говорить, они всем понятны. Самое плохое, что факт необходимости осознанного и нацеленного обучения такому мышлению мало осознается. Решение видится в организации обучения, параллельно нацеленного на формирование навыков системного и концептуального мышлений. И данный метод дает прекрасную основу для этого.

2. В настоящее время возникли системы управления нового типа, обладающие сложной структурой: многофункциональных иерархических, локально и глобально распределенных, корпоративных. Именно для них особенно нужны ЭС ППР. Но необходимых знаний практически нет, поскольку системы новые и необходимая статистика и опыт еще не осознаны и не накоплены. Соответственно, нужного качества экспертов тоже нет и вряд ли они появятся в ближайшем будущем. Из существующих экспертов, или тех людей, которые могут что-то передать, вытаскивать практически нечего. Однако они могут быть весьма полезными при соответствующей технологии и технике работы с ними, когда они будут работать не только в режиме извлечения знаний, но и их проектирования. Выход видится в подходе к

проблеме БЗ, как к проектированию практически с **чистого листа**.

3. Современные организации (особенно корпоративные) накопили огромные объемы данных (БД, ХД), однако не умеют эффективно использовать их в процессах управления. Нет единой методологической основы, которая помогала бы им в этом. Представляется, что данный метод может помочь и здесь, поскольку нетрудно сформулировать задачу порождения гипотез для обнаружения скрытых (неявных) знаний из данных.

4. Перечисленные проблемы имеют общий характер и их решение, соответственно, требует общего подхода. Как показала жизнь, принятые методы и технологии обучения специалистов подобного рода (инженеров по знаниям, аналитиков) в ВУЗах не достаточны. Необходимы специальные подходы, способствующие дать положительную мотивацию студентам, активизировать их творческий потенциал, воображение. Одновременно, эти подходы должны дать возможность чувственно пережить конкретный опыт работы с достаточно серьезными объектами управления. Такими возможностями обладают игровые подходы и область проектирования нового знания, новых систем знаний и управления является благодатной почвой для них.

Именно поэтому представляется, что наиболее эффективным методом инженерии знаний в процессе проектирования на основе предложенного подхода, является игровой, при котором на разных этапах проектирования могут применяться игры различных типов: индивидуальные, групповые, с компьютерной поддержкой и без нее [2,3,12].

6. Заключение

Предложенный метод прошел многолетнюю апробацию при проектировании реальных ЭСППР, внедрен в учебный процесс в ряде вузов для выполнения лабораторных, курсовых и дипломных работ, и соответственно был накоплен огромный банк демонстрационных и исследовательских ЭСППР.

Литература

1. БОЛОТОВА Л.С., ИНОЗЕМЦЕВ С.П., ЯЩЕНКО Н.Ю. *Модели и методы представления знаний в системах искусственного интеллекта*. – СПб.: ВИКИ им. А.Ф. Можайского, 1991. – 116 с.
2. БОЛОТОВА Л.С., МОРОЗ Ю.В. *Игровые подходы к обучению инженера по знаниям* // Управление большими системами: Сб. тр. – Вып. 26. – М.: ИПУ РАН, 2009. – С.
3. БОЛОТОВА Л.С., МОРОЗ Ю.В. *Обучение построению модели предметной области в процессе концептуальной игры* // Теоретические вопросы вычислительной техники и программного обеспечения: Межвуз. сб. научн. тр. – М.: МИРЭА, 2008. – С. 5-10.
4. БОЛОТОВА Л.С., СМЕРНОВ Н.А., СМОЛЪЯНИНОВ А.А. *Системы искусственного интеллекта. Теоретические основы СИИ и формальные модели представления знаний*: Учеб. пособие. – М.: МИРЭА, 2003. – 124 с.
5. БОЛОТОВА Л.С., СМОЛЪЯНИНОВ А.А. *Неформальные модели представления знаний в системах искусственного интеллекта*: Учеб. пособие. – М.: МИРЭА, 1999. – 99 с.
6. БОЛОТОВА Л.С., СМОЛЪЯНИНОВА В.А., СМЕРНОВ С.С. *Понятийно-объектная модель как формализованное представление концептуального пространства предметной области* // 51-я научно-техническая конференция: Сб. тр. – М.: МИРЭА, 2002. – С. 86-90.
7. БОЛОТОВА Л.С., СМОЛЪЯНИНОВА В.А., СМЕРНОВ С.С. *Теоретические основы концептуального проектирования модели предметной области в корпоративных информационных системах* // Теоретические вопросы вычислительной техники и программного обеспечения: Межвуз. сб. научн. тр. – М.: МИРЭА, 2006. – С. 125-131.
8. ГАВРИЛОВА Т.А., ГУЛЯКИНА Н.А. *Визуальные методы работы со знаниями: попытка обзора* // Искусственный интеллект и принятие решений. – 2008. – №1. – С. 15-21.

9. ГАВРИЛОВА Т.А., ЧЕРВИНСКАЯ К.Р. *Извлечение и структурирование знаний для экспертных систем.* – М.: Радио и связь, 1992. – 200 с.
10. КЛЫКОВ Ю.И. *Ситуационное управление большими системами.* – М.: Энергия, 1974. – 135 с.
11. КУЧКАРОВ З.А., НИКАНОРОВ С.П., СОЛНЦЕВ Г.В., ШАБАРОВ В.Н. *Концептуальный анализ и проектирование. Исследование социально-экономических систем. Методология. Теория. Следствия.* – М.: Концепт, 2007 – 844 с.
12. МОРОЗ Ю.В. *Игровое обучение концептуализации предметной области // Программные продукты и системы.* – 2009. – № 2(86) – С. 193-196.
13. НИКАНОРОВ С.П., НИКИТИНА Н.К., ТЕСЛИНОВ А.Г. *Введение в концептуальное проектирование АСУ: Анализ и синтез структур.* – М.: РВСН, 1995. – 234 с.
14. ПОСПЕЛОВ Д.А. *Большие системы. Ситуационное управление.* – М.: Знание, 1975. – 64 с.
15. ПОСПЕЛОВ Д.А. *Ситуационное управление: Теория и практика.* – М.: Наука, 1986. – 284 с.
16. ПОСПЕЛОВ Д.А., ПУШКИН В.Н. *Мышление и автоматы.* – М.: Сов. радио, 1972. – 224 с.
17. СКОРОХОДЬКО Э.Ф. *Информационно-поисковая система БИТ.* – Киев: Наук. думка, 1968. – 120 с.
18. СКОРОХОДЬКО Э.Ф. *Семантические сети и автоматическая обработка текста.* – Киев: Наук. думка, 1983. – 218 с.
19. СМIRНОВ С.С., БОЛОТОВА Л.С., СМОЛЬЯНИНОВА В.А. *Организация логического вывода на понятийно-объектной модели предметной области в системах поддержки принятия решений // Теоретические вопросы вычислительной техники и программного обеспечения: Межвуз. сб. научн. тр.* – М.: МИРЭА, 2003. – С. 123-133.
20. ШРЕЙДЕР Ю.А. *Семиотические основы информатики.* – М.: ИПКИР, 1975. – 80 с.
21. ШРЕЙДЕР Ю.А., ШАРОВ А.А. *Системы и модели.* – М.: Радио и связь, 1982. – 152 с.

GAME APPROACH TO KNOWLEDGE ENGINEER TRAINING

Lyudmila Bolotova, Moscow State Institute of Radioengineering, Electronics and Automation, Doctor of Science, Professor
(lubolotova@mail.ru).

Yuliya Moroz, Moscow State Institute of Radioengineering, Electronics and Automation, Postgraduate Student
(moroz@mirea.ru)

Sergey Smirnov, Moscow State Institute of Radioengineering, Electronics and Automation, Senior Lecturer
(silicon_logic@rambler.ru).

Valeriya Smolyaninova, Moscow State Institute of Radioengineering, Electronics and Automation, Senior Lecturer
(valerysmol@mail.ru).

Abstract: The method of arbitrary domain model situational analysis and designing for decision support expert systems is considered. Situational control of large systems and conceptional analysis and designing of organizational control systems underlie the method. Fundamentals of the method, logical-formal apparatus of singular decision conceptual frameworks analysis and domain conceptual framework generation are given.

Keywords: expert system, knowledge engineering, situational control, situational analysis, domain model, conceptional analysis, designing, conceptualization, formalization, decision, action, conceptual framework, knowledge base, knowledge control system.