



МЕТОД АВТОМАТИЧЕСКОЙ АДАПТАЦИИ РЕШЕНИЙ В ГИБРИДНОЙ ДИНАМИЧЕСКОЙ СЦЕНАРНО-ПРЕЦЕДЕНТНОЙ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЕ

УДК 656.612

ШЕРСТЮК Владимир Григорьевич

к.т.н., доцент кафедры информационных технологий
Херсонского национального технического университета.

Научные интересы: интеллектуальные системы принятия решений реального времени, принятие решений на основе прецедентов, мультиагентные системы, комбинированные логические системы представления знаний.

e-mail: v_sherstyuk@bigmir.net

ВВЕДЕНИЕ

Управляемый оператором динамический объект (ДО) при взаимодействии с другими ДО в пределах некоторого ограниченного пространства формирует открытую полиэргатическую сложную динамическую систему (СДС) [1]. В полиэргатических СДС знания операторов ДО о текущей ситуации, как правило, неполны и неточны, действия операторов взаимодействующих ДО непредсказуемы, а нормативные правила, регулирующие взаимодействие ДО, являются противоречивыми и недостаточно определенными. Кроме того, результаты применения запланированных управляющих воздействий на исполнительные механизмы ДО не всегда приводят к требуемому результату вследствие стохастических воздействий внешней среды, а сами воздействия носят протяженный во времени и пространстве характер.

При наличии в СДС стесненных условий [2] и множественных ситуационных возмущений [3] складываются информационно сложные для оператора ситуации [4], обусловленные неполнотой и неточностью исходной информации, значительными объемами требуемых вычислений, а также серьезными ограничениями во времени, что существенно усложняет принятие опера-

тором адекватных решений по управлению ДО в сложившейся ситуации, а в некоторых случаях приводит к инцидентам и авариям, которые принято классифицировать как «воздействие человеческого фактора» [5].

Снизить зависимость от «человеческого фактора» возможно путем автоматизации процесса принятия решений оператором ДО, используя методы искусственного интеллекта [6].

В [7] предложено с целью компенсации влияния «человеческого фактора» на процесс управления ДО в информационно-сложных ситуациях использовать интеллектуальную систему (ИС) управления ДО на основе сценарно-прецедентного подхода. Как показано в [8], ИС управления ДО должна: а) функционировать в реальном времени; б) автоматически генерировать уместные в сложившейся ситуации решения.

Как показывает практика [9], в реальных прецедентных и сценарно-прецедентных ИС на некоторых этапах функционирования используются интерактивные методы, предполагающие непосредственное участие оператора в отборе, адаптации и верификации решений, в то время как оператор и без того перегружен. Соответственно, синтез сценарно-прецедентных ИС управления ДО требует разработки методов адапта-

ции и верификации решений, не требующих участия человека-оператора.

В данной работе представлен метод автоматической адаптации решений в ИС управления ДО, работающий при наличии жестких временных ограничений (в реальном времени).

Целью данной работы является дальнейшее развитие теории сценарно-прецедентных ИС и решение задач синтеза ИС управления ДО реального времени на основе сценарно-прецедентного подхода.

АНАЛИЗ СУЩЕСТВУЮЩИХ МЕХАНИЗМОВ АДАПТАЦИИ РЕШЕНИЙ

Сценарно-прецедентные ИС (СПИС) являются классом интеллектуальных систем автоматизированного вывода решений, основанным на принципах: а) повторяемости ситуаций; б) возможности использования ранее принятых решений в случае возникновения сходных проблемных ситуаций; в) представления решений в форме сценариев [10].

В отличие от широко распространенных прецедентных ИС типа «ситуация-действие» СПИС работает по принципиальной схеме «ситуация-план-сценарий-действие», и соответственно имеет более высокий уровень сложности.

Однако, используя правдоподобный вывод по прецедентам, не всегда возможно непосредственно получить адекватное решение, поскольку между сложившейся ситуацией и неким эталоном, хранящимся в виде прецедента, существует всего лишь отношение подобия (но не эквивалентности), поэтому сложившаяся и эталонная ситуации принадлежат некоторому классу ситуаций (прототипу), экземпляры которого имеют как схожие (например, структурные) свойства, так и различающиеся (внешними условиями, некоторыми параметрами и т.д.).

Соответственно, решение, принятое ранее в контексте некоторой эталонной ситуации, для использования в контексте вновь сложившейся ситуации необходимо адаптировать. Процесс адаптации предполагает изменение некоторых свойств решения, в СПИС управления ДО – структуры и параметров сценариев управляющих воздействий, составляющих планы компенсации ситуационных возмущений.

Механизмы адаптации решений в планирующих прецедентных системах классифицируются на деривационные и трансформационные [11].

Деривационный механизм адаптации используется в случае, если решение прецедента хранится в хранилище прецедентов (ХП) в символьном виде в форме логической процедуры построения плана [12]. При выборе эталонного прецедента решение формируют в процессе логического вывода, подставляя содержащие необходимые замены условий контекста утверждения, а результатом вывода является построенный для новой проблемной ситуации план.

Ввиду существенной вычислительной сложности процедур логического вывода, неразрешимости существующих логических теорий первого порядка, используемых для представления планов в планирующих прецедентных ИС, символьный способ хранения прецедентов и механизм их адаптации непригодны для СПИС реального времени, от которой требуется найти решение за достаточно малый промежуток времени.

Трансформационный механизм позволяет изменять определенные элементы решения, хранящегося в ХП в форме подготовленного к выполнению плана [13]. Как правило, в планирующих прецедентных ИС используются способы представления решений, соответствующие алгоритмам с полным или частичным упорядочением, иерархическим алгоритмам либо основанные на представлении планов графами [14].

В трансформационном механизме адаптации могут использоваться как формализованные знания, специфичные для конкретной предметной области, так и эвристические методы, инвариантные к предметной области. Существенно, что используемые для адаптации решений знания, как правило, требуют использования в процессе адаптации методов, отличных от вывода по прецедентам. В частности, к наиболее часто используемым для адаптации планов методам можно отнести вывод на основе правил, вывод на основе моделей, метод удовлетворения ограничений, генетические алгоритмы и иные нелинейные методы [15].

В то же время в [16] показано, что, несмотря на интенсивные исследования, в области адаптации

решений, сформированных по прецедентам, открытых вопросов значительно больше, чем решенных. В частности, существующие методы адаптации являются по большей части интерактивными, рассчитаны на символическое представление прецедентов и решений, а процесс адаптации сводится к изменению структуры плана удалением или заменой его элементов (шагов).

В то же время, в СПИС изменение контекста сложившейся ситуации относительно контекста эталонной ситуации может (не обязательно) требовать как изменения последовательности действий (добавление или удаление в сценариях некоторых управляющих воздействий), так и (чаще всего) изменения некоторых параметров, передаваемых в сценарии при их выполнении.

Соответственно, синтез СПИС управления ДО предполагает реализацию метода адаптации, выполняющего как структурные корректировки решений, так и параметрические.

В задачи данной работы входит анализ сценарно-прецедентной модели формирования и адаптации решений и разработка в ее рамках метода адаптации решений погружением в контекст.

ПРЕДСТАВЛЕНИЕ ЗНАНИЙ О ПРЕДМЕТНОЙ ОБЛАСТИ

Как показано в [17], любое зафиксированное в наблюдаемом потоке событий A событие может трактоваться в зависимости от точности его наблюдения как более конкретное либо более абстрактное, придерживаясь некоторой заданной иерархии абстракций (рис. 1). Уровень c_1 является абстракцией «движение вправо», c_{11} конкретизирует его как «поворот вправо», а c_{12} – как «разворот вправо». Далее, c_{112} является конкретным событием «поворот на $12,3^\circ$ вправо», а c_{111} – его абстракцией «поворот на один румб вправо».

Аналогично можно построить множественную иерархию абстракций, например на рис. 1 $c_{11} = c_{21}$ отражает и «поворот вправо» из иерархии c_1 , и «движение вперед-вправо» в иерархии c_2 , при этом абстракция c_2 имеет смысл «движение вперед».

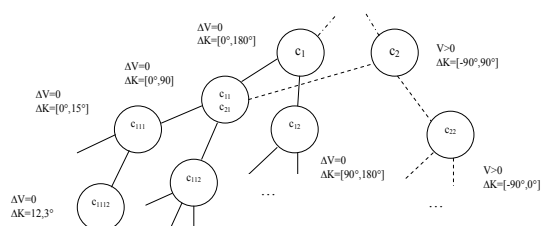


Рисунок 1 – Иерархия абстракций в представлении ситуации:

ΔV – изменение скорости
 ΔK – изменение направления движения

Отношения между уровнями иерархий имеют вид «абстракция-конкретизация» и представляют собой таксономию типа «CLASS-OF», где c_1 является надклассом для c_{11} , c_{11} – надклассом для c_{111} , и далее вниз по иерархии.

В то же время, любое зафиксированное событие входит как составная часть в один или несколько возможных сценариев, которые в свою очередь формируют предполагаемые планы (рис. 2).

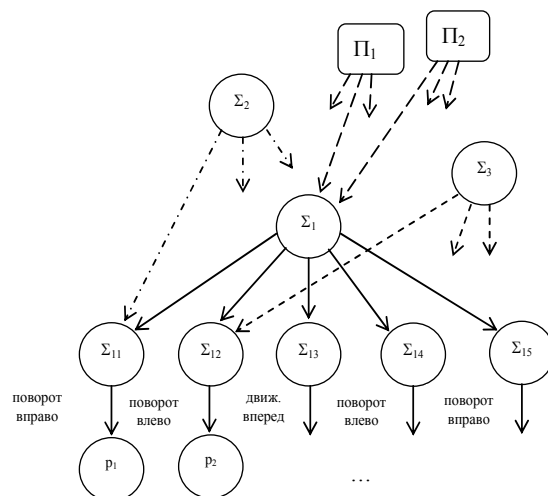


Рисунок 2 – Фрагмент схемы иерархии вложения

На рис. 2 Σ_1 представляет собой сценарий класса «обгон справа», и состоит из пяти последовательных событий, которые происходят упорядоченно во времени исключительно в порядке обхода $\Sigma_{11} - \Sigma_{15}$ слева направо. Отношения между событиями, сценариями и планами являются агрегатными («PART-OF»), а формируемая ими структура представляет собой иерархию вложения (композиционную иерархию).

Задача отображения планов Π на потоки событий c_1, c_2, \dots, c_n является нетривиальной. Так, наблюдаемое событие c_1 через его абстрактный класс c_{11} может

соответствовать началу сценария Σ_1 , а может представлять простой сценарий Σ_2 («отворот вправо»). Событие c_2 класса Σ_{12} также может быть либо частью сценария Σ_1 , либо частью (началом) сценария Σ_3 («уклонение влево»). Сам сценарий Σ_1 может быть составной частью нескольких планов (на рис. 2 Π_1 и Π_2).

Планы и потоки наблюдаемых событий в рамках используемой сценарно-прецедентной модели формирования решений организуются в древовидные сети событий [17], построенные на основе таксономической и композиционной иерархии.

Формализуем базовые понятия событийной модели с явным заданным временем.

Определение 1 Событийной моделью E называется упорядоченная пара:

$$E = \langle v, z \rangle, \quad (1)$$

где v — множество переменных модели;

z — сигнатура.

Определение 2 Сигнатурой событийной модели z называется кортеж вида:

$$z = \langle X, (\wedge_i, I_i, p_i), T, <_T \rangle, \quad (2)$$

где X — множество параметров событий;

I_i — множество элементов иерархии, соответствующей отношению w_i между событиями;

$<_i$ — отношение частичного порядка, заданное над I_i ;

\perp_i — наименьший элемент последовательности $<_i$;

T — множество значений времени;

$<_T$ — отношение полного порядка $<_T$ для T .

Так, для $i=1$ образуем таксономическую иерархию, построенную над отношением таксономии w_1 . Пусть I_1 соответствует множеству классов $C = \{c_1, c_2, \dots, c_n\}$.

Тогда отношение частичного порядка $<_1$ является по определению отношением информационной упорядоченности, т.е. $c_1 <_1 c_2$ означает, что класс событий c_1 несет в себе меньше информации, чем c_2 . Таким образом, класс c_1 является абстракцией класса c_2 , а класс c_2 — конкретизацией класса c_1 , соответственно

отношение $<_1$ задает на C таксономическую иерархию $\langle C, <_1 \rangle$.

Соответственно, \perp_1 представляет собой наиболее абстрактный (содержащий минимум информации) класс таксономической иерархии.

Определение 3 Путем $\rho(v_i, \psi_j)$ в заданной модели событий E называется последовательность элементов, ведущая по цепочке от переменной v_i к событию ψ_j .

Перейдем от событийной модели E к реализующей ее формальной модели древовидных сетей событий. Рассмотрим ориентированный связный мультиграф $g = \langle v, e \rangle$, не содержащий циклов, где v — множество вершин, e — множество дуг.

Непустое множество вершин v мультиграфа g разобьем на три непересекающихся множества: множество концевых узлов (листьев) $\tau \subset v$, множество корней, содержащих вершины наивысшего уровня $h \subset v$, и множество узлов b , не являющихся корнями или листьями, причем $g = h \cup b \cup \tau$, $h \cap b = h \cap \tau = b \cap \tau = \emptyset$. Непустое множество дуг e мультиграфа g разобьем на подмножества $e = e_{\triangleright_1} \cup e_{\triangleright_2} \cup \dots \cup e_{\triangleright_n}$, каждое из которых отображает определенное отношение $w_{\triangleright_1}, w_{\triangleright_2}, \dots, w_{\triangleright_n} \in \mathcal{W}$, заданное на v .

Определение 4 Для заданной модели событий E и потока событий $S = [\psi_1, \psi_2, \dots, \psi_n]$, таких что $\forall \psi_i \in E.z$, древовидной сетью событий (ДСС) называется структура \mathcal{G} вида:

$$G = \langle g, f, k, \{>_i\}, t \rangle, \quad (3)$$

где g — ациклический связный мультиграф;

$f: \tau \rightarrow S$ — отображение каждого концевого узла ДСС в событие из потока S ;

$\mathfrak{t}: v \rightarrow 2^{E.z.X}$ — отображение каждого узла ДСС в множество параметров и ограничений из $E.z.X$;

$t: v \rightarrow T$ — отображение каждого узла ДСС на множество значений времени $E.z.T$;

$\{\triangleright_i\}$ — множество отношений частичного порядка \triangleright_i на множестве узлов v , каждое из которых ин-

дуцировано отношением τ_{\triangleright_i} и выражается с помощью подмножества дуг $\epsilon_{\triangleright_i}$.

Корневые узлы ДСС $h \in \mathfrak{h}$ определяют множество последовательностей (потоков), состоящих из всех событий в конечных узлах $r \in \mathfrak{r}$, упорядоченных согласно $\{\triangleright_i\}$ и удовлетворяющих ограничениям, налагаемым \mathfrak{e} .

Неконцевые узлы $b \cup h$, $b \in \mathfrak{b}$, $h \in \mathfrak{h}$ представляют значимые потоки событий, представленных конечными узлами r . Каждый узел ДСС соответствует последовательности событий, представленной h , потомки каждого узла представляют сегмент родительской последовательности.

ДСС может использоваться для формализации частично упорядоченных совокупностей событий, происходящих одновременно и совместно. При этом конечные узлы отображают элементарные (наблюдаемые) события, неконцевые – соответствуют агрегатным событиям (последовательностям элементарных событий, например сценариям), в то время как корневые узлы ведут к оперативным целям соответствующих сценариев.

Таким образом, конечные узлы хранят наблюдаемые события, а неконцевые узлы – ограничения на своих потомков, включая временные отношения, включая порядок их появления.

Дуги ДСС отражают множество отношений \mathfrak{W} , в частности, таксономическую иерархию (CLASS-OF) τ_{\triangleright_1} , композиционную иерархию (PART-OF) τ_{\triangleright_2} , временные отношения \mathfrak{t} .

Важной особенностью формального определения ДСС является возможность расширения множества представимых отношений \mathfrak{W} новыми отношениями τ_k путем дополнения ϵ соответствующим подмножеством дуг ϵ_k и введения в структуру ДСС соответствующего отношения порядка \triangleright_k .

На основе представленной концепции ДСС формализована модель правдоподобных ДСС (ПДСС), в которых каждая дуга $e_j \in \epsilon$ помечается некоторой оценкой правдоподобия l_j , полученной в модели правдоподобия ℓ . Модель правдоподобия может иметь в основе

вероятностную, нечеткую, приближенную трактовку либо их комбинацию.

СЦЕНАРНО-ПРЕЦЕДЕНТНАЯ МОДЕЛЬ ФОРМИРОВАНИЯ РЕШЕНИЙ

В сценарно-прецедентной модели формирования решений ПДСС является формальной основой представления и хранения прецедентов.

Прецедент представляет собой структуру, включающую проблемную ситуацию, ее решение и полученный результат [18]. Описание проблемной ситуации кодируется в структуру, называемую иницирующим прецедентом.

Определение 5 *Иницирующим прецедентом* e_i называется описание проблемной ситуации s_i , содержащее *наблюдаемый поток событий* S и *контекст*, представляющий собой множество переменных $\{v_{e_i}^1, v_{e_i}^2, \dots, v_{e_i}^n\}$, отражающих условия наблюдения.

Формализуем эталонный прецедент. Пусть S – пространство возможных ситуаций, R – пространство возможных решений, G – пространство возможных целей.

Определение 6 *Прецедентом* e называется тройка $\langle s, r, g \rangle = S \times R \times G$, состоящая из описания ситуации $s \in S$, содержащего поток событий \mathcal{R} , называемый *эталонным потоком событий*, и *контекста*, заданный как множество переменных $\{v_e^1, v_e^2, \dots, v_e^n\}$; связанного с S решения $r \in R$, содержащего план действий Π , и связанной с r цели $g \in G$, на достижение которой направлен план $\Pi \in r$.

Цель g отражает предполагаемый результат выполнения решения r прецедента e в ситуации S .

Отметим, что иницирующий прецедент, в отличие от эталонного прецедента, не содержит описание решения и полученного результата.

Множество прецедентов, хранимых в СПИС, представлено в виде сложной двухуровневой структуры данных – ПДСС \mathfrak{E} .

Определение 7 *Хранилищем прецедентов* \mathcal{M} называется множество упорядоченных относительно $\{\triangleright_i\}$ прецедентов $\{e_1, e_2, \dots, e_n\}$, каждый из которых

$e_j, j \in [1..n]$ является кустом ПДСС $\tilde{\mathcal{E}}_1$, т.е. $e_j \triangleleft^* \tilde{\mathcal{E}}_1$, где \triangleleft^* - отношение вложения.

Другими словами, каждый прецедент является последовательным кустом ПДСС [17], а ПДСС рассматривается как упорядоченная последовательность кустов, соответствующая множеству эталонных ситуаций. Первый (нижний) уровень составляет ПДСС $\tilde{\mathcal{E}}_1$, ассоциированная с моделью событий \mathcal{E} и содержащая множество эталонных потоков событий $\{\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_n\}$, $\mathcal{R}_i \in \mathcal{E}, i \in [1..n]$. Второй уровень $\tilde{\mathcal{E}}_2$ содержит ПДСС, представляющую таксономическую структуру контекстов ситуаций $\text{Ctx}_{e_i} \triangleleft^* \tilde{\mathcal{E}}_2$, где $\text{Ctx}_{e_i} = \{v_{e_i}^1, v_{e_i}^2, \dots, v_{e_i}^n\}$, $v_{e_i}^j \in \mathcal{E}, j \in [1..n]$.

С точки зрения поиска и отбора прецедентов, подобных иницирующему e_i , далее нас интересует именно ПДСС $\tilde{\mathcal{E}}_1$. Последовательная структура \mathcal{M} отражает множество прототипов последовательностей событий, которые вызваны одним и тем же источником в каждой конкретной ситуации.

Введем определения подобного, уместного и опорного прецедентов.

Определение 8 Подобным прецедентом называется прецедент e_s , находящийся с иницирующим прецедентом e_i в отношении подобия с оценкой, большей заданного порога θ , так что $\text{SIM}(e_i, e_s) > \theta$.

Определение 9 Уместным прецедентом называется подобный иницирующему прецеденту e_i прецедент e_p , находящийся в отношении уместности с оценкой, большей заданного порога ϱ , так что $\text{SIM}(e_i, e_p) > \theta \wedge \text{REL}(e_i, e_p) > \varrho$.

Определение 10 Опорным прецедентом называется прецедент e_o , содержащий в r_o возможный план Π_o решения для проблемной ситуации s_i .

Сопоставляя иницирующий прецедент e_i с множеством прецедентов \mathcal{M} , накопленных в ХП, СПИС ищет на основании динамической оценки подобия потоков событий SIM множество подобных прецедентов $\{e_s^i\}, i \in [1..m]$.

Данную фазу процесса функционирования СПИС принято называть *отбором* прецедентов [19].

На следующей фазе, которую обычно называют *оценочной* [19], из множества подобных прецедентов $\{e_s^i\}, i \in [1..m]$ подбирается множество наиболее уместных (релевантных) проблемной ситуации прецедентов-кандидатов $\{e_p^j\}, j \in [1..k]$.

Для принятия решений в сложившейся ситуации может использоваться план $\Pi_p^* = \Pi_p^j$, содержащийся в решении единственного наиболее уместного прецедента-кандидата e_p^j . В случае, если процесс оценивания не привел к единственному наиболее уместному прецеденту-кандидату, далее рассматривается множество уместных прецедентов-кандидатов $\{e_p^j\}$.

Соответственно, план Π_p^* может быть получен путем выполнения операций пересечения, объединения, выделения частей над планами Π_p^j , содержащимися в решениях прецедентов, составляющих множество $\{e_p^j\}$.

В результате формируется опорный прецедент e_o , содержащий сформированное для проблемной ситуации s_i решение в виде плана $\Pi_o = \Pi_p^*$.

Определение 11 Критерием отбора прецедентов \mathbb{R} называется кортеж

$$\mathbb{R} = \langle (\text{SIM}), \theta, (\text{REL}), \varrho \rangle, \quad (4)$$

где **(SIM)** – динамическая оценка подобия потоков событий;

θ – пороговое значение минимально допустимой оценки подобия потоков событий;

(REL) – оценка уместности (релевантности) прецедента наблюдаемому потоку событий;

ϱ – пороговое значение минимально допустимой оценки релевантности прецедента наблюдаемому потоку событий.

Критерий отбора прецедентов может быть *неадаптивным*, если он задан статически в процессе разработки СПИС, и может быть *адаптивным*, если допускает уточнение по мере функционирования системы и накопления опыта решения проблемных ситуаций. СПИС может также включать множество критериев отбора прецедентов, подбирая в различные моменты времени критерий отбора сообразно, например, сложности проблемной ситуации.

Поскольку контекст иницирующего прецедента e_I , как правило, отличается от контекста опорного прецедента e_O , далее производится *адаптация* плана Π_O к новым условиям контекста $\Pi_O \rightarrow \Pi_A$, осуществляющая привязку полученного решения к новому контексту.

Определение 12 *Адаптированным прецедентом* называется прецедент e_A , полученный из опорного прецедента e_O привязкой его решения r_O к контексту иницирующего прецедента e_I , $r_O|s_O \rightarrow r_A|s_I$.

Далее производится *верификация* адаптированного плана Π_A – сможет ли данное решение успешно решить возникшую проблему в контексте проблемной ситуации. Верификация может производиться автоматически на заданной модели или БЗ предметной области, либо вручную с помощью оператора.

Процессы адаптации и верификации решения принято называть фазой *проверки* [19].

Решение r_A , признанное в процессе верификации несоответствующим проблемной ситуации s_I , возвращается на фазу оценки либо адаптации для корректировки или поиска нового уместного решения.

Окончательное подтверждение адаптированного решения r_A может производиться СПИС автоматически либо с помощью оператора.

Определение 13 *Порожденным прецедентом* называется прецедент e_R , полученный из адаптированного прецедента e_A , решение r_A которого успешно прошло верификацию и признано соответствующим проблемной ситуации s_I .

СПИС для установления соответствия прецедента проблемной ситуации может использовать некоторое заданное на этапе разработки множество оценок \mathcal{V} , например,

$$\mathcal{V} = \{\text{оптимальный, субоптимальный, подходящий}\}.$$

После того как выявлена проблемная ситуация s_I и в результате работы СПИС сформировано подходящее решение r_R на основе множества имеющихся прецедентов \mathcal{M} , соответствующая информация $\langle s_A \sim s_I, r_A, g_A \rangle$ упаковывается в порожденный прецедент e_R , и выполняется *сохранение* в ХП \mathcal{M} .

Определение 14 *Компетенцией* ε называется множество проблемных ситуаций, для которых СПИС способна сформировать решение, используя имеющееся множество прецедентов \mathcal{M} .

Определение 15 *Уровнем компетентности* ε называется отношение мощности компетенции СПИС ε к мощности множества возможных ситуаций S : $\varepsilon = |\varepsilon|/|S|$.

Поскольку мощность множества возможных ситуаций в СДС неизвестна, в СПИС можно использовать оценки ε -покрытия или ε -достижимости [20].

СПИС может также самостоятельно оценивать эффективность своего функционирования, используя заданную оценку эффективности ϑ .

Определение 16 *Критерием качества* Q хранилища прецедентов \mathcal{M} называется кортеж вида $Q = \langle \varepsilon, \vartheta \rangle$.

СПИС, работающие в автоматическом режиме, должны поддерживать заданный уровень компетенции ε и оценку эффективности ϑ , в необходимых случаях используя методы машинного обучения [21].

Результатом использования методов машинного обучения могут быть дополнения и изменения, вносимые:

- в БЗ предметной области;
- в структуру конкретного прецедента e_R , например в используемый сценарий Σ_R или план решения Π_R ;
- в структуру ХП \mathcal{M} , например, для формирования нового класса прецедентов $c_R \in C$, $C \in \mathcal{E}_Z$;
- в используемую систему критериев отбора подобных и уместных прецедентов \mathbb{R}^* ;
- в используемые критерии качества Q ХП \mathcal{M} .

Задачи сохранения, поддержания компетентности и эффективности, обучения традиционно решаются на фазе *поддержки* [19].

АДАПТАЦИЯ СФОРМИРОВАННОГО РЕШЕНИЯ

В результате поиска прецедентов, подобных текущей (*проблемной*) ситуации, и производимого далее отбора уместных прецедентов, СПИС формирует опорное (*базовое*) решение, представляющее собой базовый план действий, построенный в соответствии с принятой моделью адаптивного планирования. Базовый план

представляет собой решение, которое ранее было использовано для достижения аналогичной цели в некоторой опорной ситуации, связанной с текущей ситуацией отношением *сходства* (но не эквивалентности).

Непосредственно использовать базовый план для проблемной ситуации не всегда представляется возможным из-за различия между контекстом опорной ситуации и контекстом текущей ситуации. Данный факт составляет основную проблему ИС, основанных на прецедентах – решение, апробированное ранее в контексте опорной ситуации, необходимо *адаптировать* к контексту текущей ситуации.

Поскольку контекст инициирующего прецедента e_i , как правило, отличается от контекста опорного прецедента e_o , необходима адаптация плана Π_o ($\Pi_o \rightarrow \Pi_A$), соответствующая изменению контекста ситуации ($\text{Ctx}_{e_i} \rightarrow \text{Ctx}_{e_o}$) и осуществляющая привязку сформированного решения к новому контексту.

Поскольку план имеет структуру, а элементы плана (сценарии и управляющие воздействия) имеют параметры, использовать трансформационный механизм для метода адаптации решений следует как с точки зрения изменения структуры решений (структурная адаптация), так и с точки зрения изменения параметров различных элементов решения (параметрическая адаптация).

Проблема состоит в том, что адаптация решения, принятого по прецеденту, требует использования альтернативных формализмов и методов, имитирующих иные способы принятия решений [22].

Интеллектуальную систему, в которой для решения задачи используется более одного метода имитации интеллектуальной деятельности ЛПР, принято называть *гибридной интеллектуальной системой*. Гибридизация ИС понимается как процесс интеграции двух или более подсистем, каждая из которых может иметь различные языки представления и методы вывода, при этом подсистемы объединяются вместе семантически и взаимодействуют каждая с каждой [23].

Правдоподобный вывод в представленной модели СПИС может выполняться не только над прецедентами, но и над другими фрагментами знаний. Формализм ПДСС, который используется в СПИС для представления совокупности прецедентов ХП, комплементарен логи-

ческому формализму правдоподобных дескрипционных логик (ПДЛ), обладающих важным для ИС реального времени свойством разрешимости и являющихся неким компромиссом между выразительными способностями и вычислительной сложностью [24].

Таким образом, ПДСС может составлять интеграционную основу для гибридизации СПИС, необходимой с точки зрения дальнейшей адаптации и верификации полученного базового решения. Интеграция может объединить подсистемы вывода *на прецедентах, на правилах и на моделях* [25].

Параметрическая адаптация решений связана с изменением параметров $\text{Var}_{\Sigma_i} = \{V_1, V_2, \dots, V_m\}$ сценариев Σ_i управляющих воздействий, входящих в опорный план Π_o .

Метод параметрической адаптации состоит в поиске приемлемых подстановок параметров. Зависимость конкретного параметра сценария $V_j(\Sigma_i)$ от переменной контекста ситуации $v_{\text{Ctx}(e_i)}^k$ может быть выражена с помощью роли DependOn в нотации ПДЛ и введения соответствующего отношения $\omega_{>_3}$ между узлами ПДСС \tilde{G} :

$$\text{DependOn}(V_j(\Sigma_i), v_{\text{Ctx}(e_i)}^k) \Leftrightarrow \exists rol_1, \dots, rol_n : (rol_1 \circ \dots \circ rol_n (V_j(\Sigma_i), v_{\text{Ctx}(e_i)}^k)) \quad (5)$$

Наличие зависимости конкретного параметра от того или иного условия контекста может быть выражено композицией ролей $rol_1 \circ \dots \circ rol_n$, представляющей собой путь $\rho(V_j(\Sigma_i), v_{\text{Ctx}(e_i)}^k)$, присутствующий в ПДСС \tilde{G} .

Пусть v_1 – параметр сценария, v_2 – параметр контекста. Если существует аксиома $\text{DependOn}(v_1, v_2)$, причем $v_2 \in \text{Ctx}_{e_o}$, $v_1 \in r_o$, задача параметрической адаптации сводится к нахождению такого v'_1 , что сохраняется выполнимость аксиомы $\text{DependOn}(v'_1, v'_2)$ и $v'_2 \in \text{Ctx}_{e_i}$. Результатом параметрической адаптации будет замена значения параметра сценария v_1 значением v'_1 , привязанным к новому контексту ситуации.

Для выполнения замены необходимо (рис. 3):

- 1) найти кратчайший путь $\rho(v_1, v'_2)$ между вершинами ПДСС, соответствующими v_1 и v'_2 ;
- 2) используя построенный кратчайший путь $\rho(v_1, v'_2)$, найти такую замену v'_1 в окружении параметра v_1 , которой можно было бы заменить v_1 (и которая была бы связана композицией ролей $rol_1 \circ \dots \circ rol_n$ с параметром контекста v'_2).

Решение первой задачи сводится к решению задачи нахождения кратчайшего пути в направленном ациклическом графе g [26]. Решение второй задачи сводится к рекурсивному (относительно последовательности всех ролей, композиция которых составляет путь $\rho(v_1, v'_2)$) решению логической проблемы выборки всех экземпляров заданного концепта в ПДЛ [24].

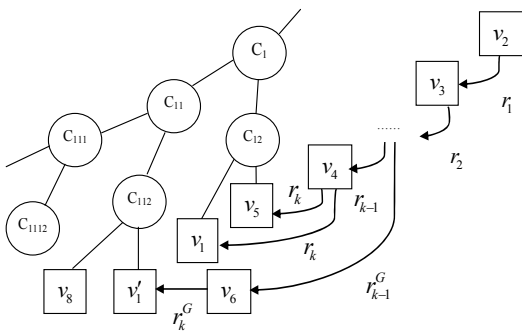


Рисунок 3 – Построение пути $\rho(v_1, v'_2)$ в ПДСС $\tilde{\Theta}$

Для выполнения параметрической адаптации используется процедура *Bind*, принимающая в качестве параметров v_1 и v'_2 , находящая в ПДСС $\tilde{\Theta}$ путь $\rho(v_1, v'_2)$, и возвращающая v'_1 :

$$v'_1 = Bind(v_1, v'_2, \langle rol_1, \dots, rol_n \rangle) \quad (6)$$

Поскольку существование прямого пути $\rho(v_1, v'_2)$ между v_1 и v'_2 не гарантировано, процедура *Bind* может в процессе поиска пути $\rho(v_1, v'_2)$ поднимать уровень абстракции концептов v_1 и v'_2 ($root(v_1)$, $root(v'_2)$), т.е. выполнять на ПДСС известную в дескрипционной логике операцию обобщения (абстракции) [24] до тех пор, пока искомым параметр v'_1 не будет достигнут.

Структурная адаптация решений связана с изменением структуры решения – в том числе последовательности и состава сценариев в плане, последовательности управляющих воздействий в сценарии Σ и т.д.

Метод структурной адаптации решений состоит в последовательном выполнении множества правил трансформации решения (адаптационных правил) $\{AR\}_{v_2}$, зависящих от параметра контекста v_2 .

Структурная зависимость целевой вершины g от контекста Ctx_e может быть представлена с помощью роли *Depend By* в нотации ПДЛ, связывающей вершину g с вершиной ar , связанной с множеством правил $\{AR\}$ ролью *HasRule*:

$$Depend\ By(g, ar) \mid \exists AR_1, \dots, AR_m : \forall AR_i\ HasRule(ar, AR_i) \quad (7)$$

Соответствующая структура ролей представлена на рис. 4. В ПДСС $\tilde{\Theta}$ имплементировано соответствующее отношение $\omega_{\triangleright_4}$ между его узлами.

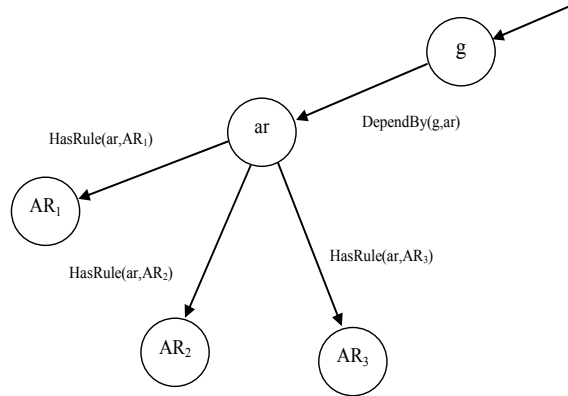


Рисунок 4 – Структура ролей, выражающих зависимость целевой вершины g

от контекста с помощью множества правил $\{AR_1, AR_2, AR_3\}$

Каждое трансформационное правило AR_i имеет вид продукции $CR \Rightarrow ZR$.

Условие CR определяется правильно построенной формулой ϕ языка ПДЛ, причем для ϕ выбор переменных ограничен множеством параметров контекста ситуации $Ctx_{e_A} = \{v_{e_A}^1, v_{e_A}^2, \dots, v_{e_A}^n\}$.

Действие ZR описывает одну из трех возможных операций трансформации решения, преобразующего куст Υ_g ПДСС $\tilde{\Theta}$, соответствующий целевой вершине g :

- удаление куста Υ_k , последовательного кусту Υ_j ($-Cluster(\Upsilon_k, \Upsilon_j)$), $\Upsilon_k \triangleleft \Upsilon_j$, $\Upsilon_j \triangleleft^* \Upsilon_g$;
- добавление куста Υ_k к кусту Υ_j ($+Cluster(\Upsilon_j, \Upsilon_k)$), $\Upsilon_j \triangleleft^* \Upsilon_g$, $\Upsilon_k \triangleleft \Upsilon_j$;
- замена куста Υ_j другим кустом Υ_k ($Replace\ Branch(\Upsilon_j, \Upsilon_k)$), $\Upsilon_j \triangleleft^* \Upsilon_g$.

Выполнение указанных операций основывается на известных методах теории графов [27].

Для выполнения структурной адаптации используется процедура $Fire$, принимающая в качестве входного параметра ссылку C на целевую вершину ПДСС g . Процедура $Fire$ находит все вершины ar_i , такие что $\forall ar_i ar_i \triangleleft g$, и последовательно (в порядке, заданном ранжированием правил по степени их уместности) обходит все вершины $AR_j \triangleleft ar_i$, проверяя условие $CR_j \in AR_j$ и в случае его истинности выполняя операции трансформации, составляющие $ZR_j \in AR_j$.

Выходным параметром процедуры $Fire$ (результатом) является куст Υ'_g ПДСС с вершиной g , подвергнутый структурной трансформации.

МЕТОД АДАПТАЦИИ РЕШЕНИЙ ПОГРУЖЕНИЕМ В КОНТЕКСТ

Процесс адаптации решения базового прецедента e_o основан на парадигме погружения плана Π_o в контекст ситуации Ctx_{e_a} , построенный на основе контекста проблемной ситуации Ctx_{e_t} .

Вначале выполняется расчет параметров всех моделей, связанных с целевой вершиной g , содержащей куст Υ_g ПДСС $\tilde{\Theta}_1$, который и представляет собой план Π_o .

Связь определенных моделей с целевой вершиной g задается с помощью роли $Depend\ As$ в нотации ПДЛ, связывающей вершину g с вершиной m ,

связанной с множеством моделей $\{M\}$ ролью $Has\ Model$:

$$Depend\ As(g, m) \mid \exists M_1, \dots, M_n : \forall M_k Has\ Model(m, M_k) \quad (8)$$

Соответствующая структура ролей связки $Depend\ As - Has\ Model$ аналогична структуре ролей в связке $Depend\ By - Has\ Rule$, представленной на рис. 4.

Расчет параметров моделей выполняется процедурой Run , аналогично процедуре $Fire$ принимающей в качестве входного параметра ссылку C на целевую вершину ПДСС g и контекст проблемной ситуации Ctx_{e_t} :

$$Ctx_{e_a} = Run(g, \{v_{e_t}^1, v_{e_t}^2, \dots, v_{e_t}^n\}) \quad (9)$$

Процедура Run находит все вершины m_j , такие что $\forall m_j m_j \triangleleft g$, и последовательно обходит все вершины $M_k \triangleleft m_j$, иницируя каждую модель M_k соответствующими значениями параметров $v_{e_t}^1, v_{e_t}^2, \dots, v_{e_t}^n$ из контекста Ctx_{e_t} .

Выходные параметры моделей M_k последовательно переносятся в множество переменных $\{v_e^1, v_e^2, \dots, v_e^n\}$, создающих контекст Ctx_{e_a} , который и является выходным параметром процедуры Run .

В результате расчета параметров моделей получаем контекст Ctx_{e_a} , над которым далее производим структурную адаптацию.

Поскольку целевая вершина g ПДСС $\tilde{\Theta}_1$ известна, для выполнения структурной адаптации вызывается процедура $Fire(g)$, в результате чего структура Π_o , соответствующая кусту Υ_g ПДСС $\tilde{\Theta}_1$, трансформируется относительно контекста Ctx_{e_a} .

Далее производится параметрическая адаптация. При этом производится последовательный обход всех сценариев $\Sigma_i \in \Pi_o$, для каждого сценария просматривается список переменных $Var_{\Sigma_i} = \{V_1, V_2, \dots, V_m\}$ и по каждой переменной проверяется существование зависимостей, выраженных связями $Depend\ On$.

Если для переменной V_j обнаруживается роль DependOn, вызывается процедура $Bind(V_j, v_i)$, $v_i \in \text{Ctx}_{e_A}$. В результате выполнения параметрической адаптации все переменные V_j всех сценариев Σ_i плана Π_A становятся привязанными к контексту Ctx_{e_A} , а прецедент e_O становится адаптированным прецедентом e_A и передается на верификацию.

Следует заметить, что представленный метод адаптации погружением в контекст не свободен от недостатков. Так, множества правил, связанных с целевой вершиной ролями Depend By - HasRule, могут образовывать известное в теории продукционных систем конфликтное множество [28], а последовательный способ инициирования моделей может привести к тому, что выходные параметры некоторой модели могут быть перекрыты выходными параметрами последующей модели.

С целью преодоления указанных недостатков для управления ХП может быть использован консервативный принцип, заключающийся в том, что правильно построенное ХП (ПДСС $\tilde{\Phi}$) содержит в каждой целевой вершине ссылку не более чем на m правил и не более чем на n моделей. При правильно подобранных значениях m и n достигается баланс между размерами ПДСС и ветвлением в целевых вершинах, – т.е. между параметрами, находящимися в обратной зависимости.

От наличия баланса в ХП зависит конечная производительность СПИС: чем раскидистее ветви в целевых вершинах, тем сложнее выполнять перебор связей в процессе адаптации, но чем больше размер ПДСС, тем сложнее (и дольше) поиск подобных прецедентов.

Для управления ХП может быть использован и либеральный принцип, разрешающий неограниченное ветвление в целевых вершинах. Такой подход

обеспечивает минимальный размер ПДСС, но при этом адаптация прецедентов значительно усложняется. Кроме того, при либеральном подходе требуется обеспечить непротиворечивость множества моделей $\{M\}$ и множества правил $\{AR\}$ для каждой целевой вершины g [29].

ОСНОВНЫЕ РЕЗУЛЬТАТЫ И ВЫВОДЫ

В результате проведенного исследования представлен метод адаптации сформированных решений погружением в контекст проблемной ситуации, предназначенный для использования в гибридных динамических СПИС управления ДО. Показаны способы структурной и параметрической адаптации сценариев управляющих воздействий, составляющих план компенсации ситуационных возмущений, к условиям контекста проблемной ситуации.

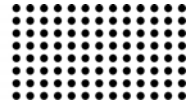
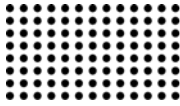
Благодаря использованию формализма ПДСС, метод работоспособен в условиях неполноты и неточности исходной информации, недостаточной компетентности ХП, жестких временных ограничений на формирование решений (в реальном времени).

Использование метода адаптации сформированных решений погружением в контекст проблемной ситуации приводит к гибридизации ИС управления ДО, что влечет необходимость дополнительной реализации в СПИС методов вывода на основе правил и на основе моделей.

В то же время, значительное усложнение структуры гибридной СПИС сполна компенсируется возможностью автоматизированной подготовки решений оператору ДО в режиме реального времени. Дальнейшее развитие метода возможно в направлении инкапсуляции различных моделей удовлетворения ограничений (CSP, Constraint Satisfaction Problem) и расширения множества трансформационных операторов для изменения структуры решений имеющихся прецедентов.

ЛИТЕРАТУРА:

1. Шерстюк В.Г. Логико-вероятностный подход к оцениванию состояния динамической системы /В.Г. Шерстюк //Вестник Херсонского национального технического университета. – 2010. – №3(39). – С.514-520.
2. Шерстюк В.Г. Информационная технология поддержки принятия решений по управлению движением судна /В.Г. Шерстюк //Вестник Херсонского национального технического университета. – 2008. – №4(33). – С.180-189.
3. Мальцев А.С. Маневрирование судов при расхождении /А.С. Мальцев. – Одесса: Морской тренажерный центр, 2004. – 212 с.



4. Сиек Ю.Л. Принципы синтеза интеллектуальных систем управления морскими динамическими объектами /Ю.Л. Сиек, Соэ Мин Лвин //Искусственный интеллект. – 2009. – №4. – С.448-456.
5. Топалов В.П. К проблеме человеческого фактора в судоходстве /В.П. Топалов, В.Г. Торский, Ю.В. Торский //Судовождение. – 2004. – Вып. 8. – С.94-102.
6. Шерстюк В.Г. Принципы интеллектуальной поддержки принятия решений по управлению движением судна /В.Г. Шерстюк //Вестник Херсонского национального технического университета. – 2009. – №3(36). – С.133-141.
7. Шерстюк В.Г. Интеллектуальные системы поддержки принятия решений по управлению судном в условиях неполной и противоречивой информации /В.Г. Шерстюк, А.П. Бень //Судовождение. – 2007. – Вып. 14. – С.141-144.
8. Мальцев А.С. Интеллектуальные гибридные системы поддержки принятия решений при расхождении судов /А.С. Мальцев //Судовождение. – 2006. – Вып. 11. – С.74-86.
9. Варшавский П.Р. Моделирование рассуждений на основе прецедентов в интеллектуальных системах поддержки принятия решений /П.Р. Варшавский, А.П. Еремеев //Искусственный интеллект и принятие решений. – 2009. – №1. – С.45-57.
10. Шерстюк В.Г. Сценарно-прецедентный подход к формированию управляющих воздействий в системе управления морского подвижного объекта /В.Г. Шерстюк //Проблемы информационных технологий. – 2009. – №2 (006). – С.69-77.
11. Munoz-Avila H. Case-based Plan Adaptation: An Analysis and Review /Hector Munoz-Avila, Michael T. Cox //IEEE Intelligent Systems. – 2008. – Vol. 23. – No. 5. – Pp.75-81.
12. Ihrig L. Derivational replay for partial order planning /L.H. Ihrig, S. Kambhampati //Proceedings of the Twelfth National Conference on Artificial Intelligence AAAI-94. – Seattle, WA. – 1994. – Pp.116-125.
13. Cox M. Case-based planning /M.T. Cox, H. Muñoz-Avila, R. Bergmann //Knowledge Engineering Review. – 2006. – Vol. 20. – No.3. – Pp.283-287.
14. Gereveni A. Fast Plan Adaptation through Planning Graphs: Local and systematic search techniques /A. Gereveni, I. Serenia //Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling (AIPS-2000). – Breckenridge, CO. – 2000. – Pp.112-121.
15. Grech, A., & Main, J. (2004). Case-Base Injection Schemes to Case Adaptation Using Genetic Algorithms In Advances in Case-Based Reasoning: 7th European Conference, ECCBR 2004. Berlin: Springer.
16. Serina I. Kernel functions for case-based planning /Ivan Serina //Artificial Intelligence. – 2010. – Vol. 174. – No. 16-17. – Pp.1369-1406.
17. Шерстюк В.Г. Использование деревьев событий для представления знаний в динамических прецедентных интеллектуальных системах /В.Г. Шерстюк //Вестник Херсонского национального технического университета. – 2011. – №2 (41). – С.100-111.
18. Шерстюк В.Г. Формальная модель гибридной сценарно-прецедентной СППР /В.Г. Шерстюк //Автоматика. Автоматизация. Электротехнические комплексы и системы. – 2004. – №1. – С.114-122.
19. Aamodt A. Case-based reasoning: foundational issues, methodological variations, and system approaches /A. Aamodt, E. Plaza //AI Communications. – 1994. – Vol. 7. – №1. – Pp.39-59.
20. Lopez de Montaras R. Retrieval, reuse and retention in case-based reasoning /R. Lopez de Montaras, D. Leake, B. Smyth, M.T. Cox, A. Aamodt, I. Watson etc. //Knowledge Engineering Review. – 2006. – Vol. 20. – No. 3. – Pp.215-240.
21. Lopez B. Case-based learning of strategic knowledge /B. Lopez, E. Plaza //Blanes: CSIC Research Report GRIAK 90/14, 1990. – 83 p.
22. Pal S.K. Foundation of Soft Case-Based Reasoning /S.K. Pal, S.C.K. Shiu. – New Jersey: J. Wiley & Sons, 2004. – 274 p.
23. Гаврилов А.В. Гибридные интеллектуальные системы /А.В. Гаврилов. – Новосибирск: Изд-во НГТУ, 2003. – 168 с.
24. Baader F. The Description Logic Handbook: Theory, Implementation, Applications /F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, P.F. Patel-Schneider. – Cambridge University Press, Cambridge, UK, 2003. – 574 p.
25. Hunt J. Hybrid case-based reasoning /J. Hunt, R. Miles //The Knowledge Engineering Review. – 1994. – Vol. 9. – Pp.383-397
26. Левитин А.В. Алгоритмы: введение в разработку и анализ /А.В. Левитин. – М.: «Вильямс», 2006. – 576 с.
27. Кристофидес Н. Теория графов. Алгоритмический подход. /Н. Кристофидес. – М.: Мир, 1978. – 429 с.
28. Рассел С. Искусственный интеллект: современный подход /С. Рассел, П. Норвиг. – М.: Вильямс, 2006. – 1408 с.
29. Поспелов И.Г. Динамическое описание систем продукции и проверка непротиворечивости производственных экспертных систем /И.Г. Поспелов, Л.Я. Поспелова //Изв. АН СССР: Техническая кибернетика. – 1987. – №1. – С.184-192.