

Темпоральные модели на основе логики ветвящегося времени в интеллектуальных системах¹

Аннотация. Рассматриваются методы организации временных (темпоральных) рассуждений (методы темпорального вывода) в интеллектуальных системах на примере систем поддержки принятия решений. Основное внимание уделяется темпоральной логике ветвящегося времени и ее использованию в системах реального времени.

Ключевые слова: искусственный интеллект, интеллектуальная система поддержки принятия решений, темпоральная логика, логика ветвящегося времени.

Введение

Возможность использования в современных интеллектуальных системах (ИС) моделей рассуждений с учетом фактора времени (временных зависимостей) позволяет перейти на качественно новый уровень решения задач в таких областях как принятие (поиск) решений, планирование, диагностика и мониторинг, вопросно-ответные системы, работа с естественным языком, организация знаний и обработка больших массивов (хранилищ) данных, верификация программ и программирование агентов и т.д. [1, 2].

Актуальность наличия развитых средств представления и оперирования временными (темпоральными) зависимостями неоднократно отмечалась уже в ранних работах по кибернетике и искусственному интеллекту (например, работы [1,3,4]). Особенно необходимы эти средства при создании перспективных ИС семиотического типа, способных развиваться и адаптироваться к специфике предметной области (внешней среды) и решаемым задачам [2,3,5,6]. Обзор различных моделей и методов представления темпоральных зависимостей в плане их применения в ИС и, в частности, ИС поддержки принятия решений реального времени (ИСППР РВ), а также возможности их реализации, содержится в работах [7-11]. Но в

них, как правило, основное внимание уделяется моделям (темпоральным логикам) с линейной структурой времени, т.е. когда существует полностью определенное отношение предшествования для временных примитивов (моментов или интервалов), или, иначе говоря, единственное будущее и единственное прошлое. Однако, говоря о топологии времени, необходимо учитывать и возможность ветвящейся и параллельной структуры времени [12-17]. Ветвящаяся структура времени в противоположность линейной допускает множественность будущего и/или прошлого. Такое время соответствует концепции «возможных миров», а параллельная структура времени определяет различные параллельные миры.

Создавая компьютерные системы темпорального вывода, т.е. системы, моделирующие темпоральные (временные) рассуждения (СВР), необходимо учитывать ряд требований к таким системам [15,16]. Прежде всего это выразительная мощность — язык СВР должен позволять пользователю представлять знания достаточно просто и естественным образом, особенно в случае сложных (ветвящихся) темпоральных выражений. Во-вторых, СВР должна обладать хорошо определенными синтаксисом и семантикой, а также эффективной реализацией (автоматизированной процедурой вывода), характеризующейся

¹ Работа выполнена при финансовой поддержке РФФИ (проект № 11-01-00140).

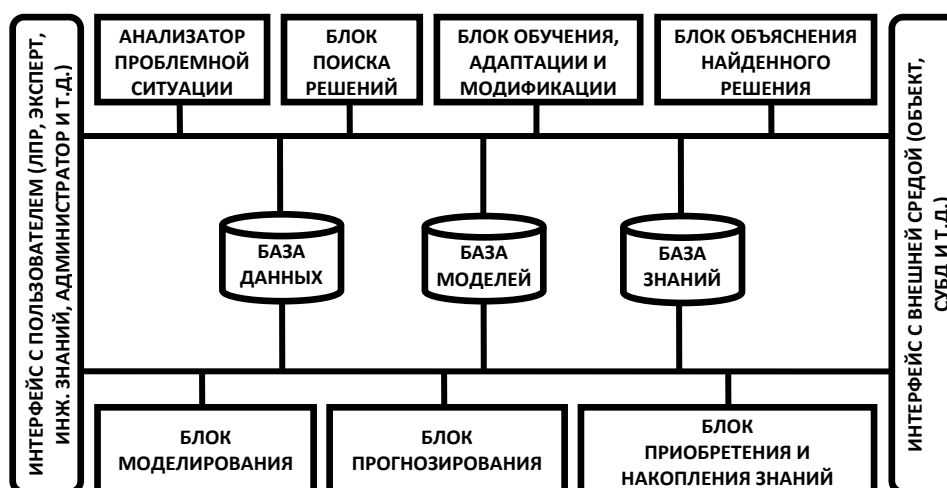


Рис. 1. Базовая архитектура ИСППР РВ

надежностью и полнотой. Учитывая природу времени, язык СВР должен обладать унифицированным подходом к представлению декларативных и процедурных аспектов знаний.

1. Специфика ИСППР РВ

В настоящее время весьма актуальной проблемой в области разработки ИС является проблема создания ИСППР РВ, ориентированных на открытые и динамические предметные области [2,6]. В основе таких систем лежит интеграция способных к адаптации, модификации и обучению моделей представления и оперирования знаниями, учитывающих специфику проблемной (предметной) области и тип неопределенности в имеющейся информации (данных и знаниях). Обобщенная структура ИСППР РВ представлена на Рис. 1.

При реализации методов моделирования рассуждений в ИСППР РВ необходимо учитывать специфику подобных систем:

- необходимость получения решения в условиях временных ограничений, определяемых реальным управляемым процессом;
- необходимость учета временного фактора (зависимостей) при описании проблемной ситуации и в процессе поиска решения;
- невозможность получения всей объективной информации, необходимой для решения, и, в связи с этим, использование субъективной, экспертной информации;
- многовариантность поиска, необходимость применения методов правдоподобного вывода и активного участия в процессе поиска ЛПР;

- наличие неопределенности (недетерминизма), необходимость коррекции и введения дополнительной информации в базу знаний системы при поиске решения.

Необходимость представления знаний и большого объема данных, меняющихся со временем (показаний датчиков, значений управляющих параметров, выполняемых операторами действий и т.д.), возникает при решении многих задач ИСППР РВ, в частности, задач диагностики, мониторинга, планирования, прогнозирования и др. Использование при решении этих задач информации о времени позволяет сократить поисковые пространства и повысить скорость реакции системы [7,8]. Модели (логики) ветвящегося времени могут быть полезны при прогнозировании последствий принимаемых решений или развития некоторой ситуации в условиях дефицита времени, что характерно для ИСППР РВ.

Рассмотрим основные положения темпоральных моделей на основе логики ветвящегося времени в плане их использования в ИС, включая ИСППР РВ.

2. Темпоральная логика ветвящегося времени

Темпоральная логика ветвящегося времени (ветвящаяся темпоральная логика) может быть использована для решения задач обучения, прогнозирования и моделирования в ИС, когда необходимо рассматривать время ветвящимся в будущее. Известно, что темпоральная логика может быть построена на основе модальной

логики [12,13]. Модальный операторы необходимости $\Box p$ может интерпретироваться как «необходимо, что всегда будет p », а модальный оператор возможности $\Diamond p$ – как «возможно, что будет p », или, другими словами, $\Box p$ истинно в момент времени t , если и только если p истинно при всех t' на всякой ветви времени, выходящей из t , а $\Diamond p$ истинно при всех t , если и только если p истинно при некотором t' хотя бы на одной ветви, выходящей из t .

В работе [12] анализируются различные подходы к темпоральным высказываниям, являющимся утверждениями о будущем. При одном подходе утверждения о будущем рассматриваются аналогично утверждениям о прошлом и настоящем как констатация положения дел. В этом случае утверждение «когда-нибудь будет p » истинно в момент времени t , если и только если в некоторый момент t' , следующий за t , истинно p . Если будущее состояние однозначно детерминировано настоящим, то утверждения о будущем будут либо истинными, либо ложными. Однако более соответствует реальности ситуация, когда допускается, что будущее не предопределено однозначно настоящим и возможны различные варианты развития событий, т.е. возможно ветвление в будущее. При втором (альтернативном) подходе утверждения о будущих событиях рассматриваются как модальные утверждения. В этом случае выражение типа «всегда будет p », «когда-нибудь будет p », «через n единиц времени будет p », корректные (правильно построенные) в случае первого подхода, не являются корректными.

Правильно построенными выражениями будут:

$G\Box p$ – «необходимо (при любом развитии событий) всегда будет p »;

$G\Diamond p$ – «возможно (при некотором развитии событий) всегда будет p »;

$F\Box p$ – «необходимо когда-нибудь будет p »;

$F\Diamond p$ – «возможно когда-нибудь будет p »;

$F_n\Box p$ – «через n единиц времени необходимо будет p »;

$F_n\Diamond p$ – «через n единиц времени возможно будет p ».

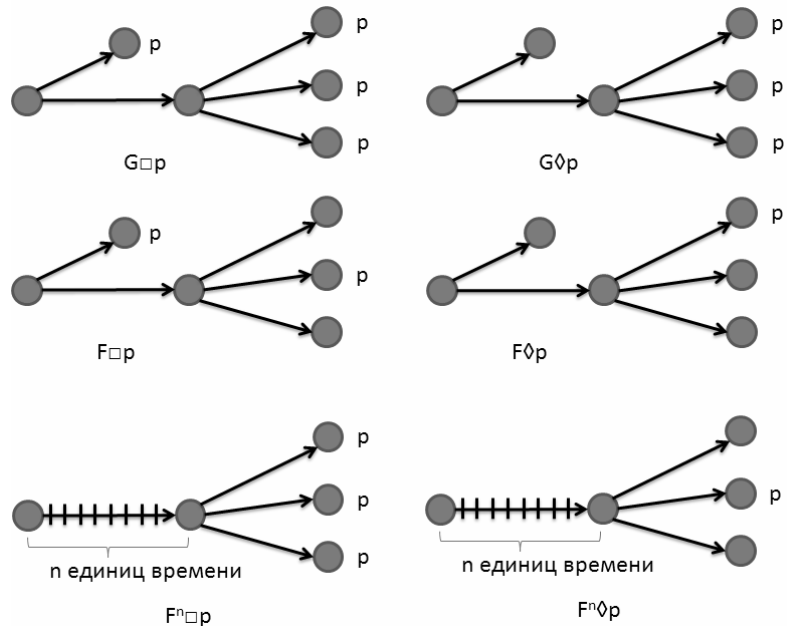


Рис. 2. Графическое представление темпоральных операторов

Для каждого приведенного оператора процесс развития событий представляется в виде дерева, ветвящегося в будущее (Рис.2). Введенные операторы являются единичными, так называемыми модализированными темпоральными операторами, а не комбинацией модальных и темпоральных операторов. Данный подход весьма перспективен в смысле выразительности представления, но не проработан в плане его практической реализации.

3. Пропозициональная темпоральная логика ветвящегося времени

Более проста в плане реализации **пропозициональная темпоральная логика ветвящегося времени (BPTL - Branching-Time Propositional Temporal logic)**, предложенная в [14] и являющейся расширением **пропозициональной темпоральной логики (PTL)**. PTL является модальной темпоральной логикой, построенной на основе классической логики с добавленными модальными операторами для дискретного линейного времени.

Синтаксис PTL задается следующим образом. Языком L_P PTL является счетное множество пропозициональных символов $p, q, r, s, \dots$. Формулы строятся, используя следующие символы:

- Множество пропозициональных символов языка L_P ;

- Классические связки: $T, F, \neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$;
- Темпоральные операторы будущего времени: унарные - $O, \Diamond, \Box$; бинарные - U, W ;
- Темпоральные операторы прошлого времени: унарные - $\otimes, \bullet, \blacklozenge, \blacksquare$; бинарные - J, Z .

Множеством правильно построенных формул (ппф) PTL являются:

- Все пропозициональные символы языка L_P есть ппф;
- Если A и B – ппф, то ппф также будут и следующие выражения: $T, F; \neg A, A \wedge B, A \vee B, A \Rightarrow B, A \Leftrightarrow B; OA, \Diamond A, \Box A, AUB, AWB; \bullet A, \otimes A, \blacklozenge A, \blacksquare A, AJB, AZB$.

Интуитивный (неформальный) смысл модальных операторов следующий. Унарных: O - следующий (next), $\bullet$ – прошлый (last), $\otimes$ - прошлый (last), $\Box$ - всегда в будущем (always in the future), $\blacksquare$ - всегда в прошлом (always in the past), $\Diamond$ - иногда в будущем (sometime in the future), $\blacklozenge$ - иногда в прошлом (sometime in the past); бинарных: U - до тех пор, пока (until), W - пока не (unless), J - с тех пор, как (since), Z - с тех пор, как (zince). Если A и B являются пропозициональными формулами, то интуитивный смысл модальных формул определяется следующим образом: «Ппф OA истинна в данный момент (в данном состоянии), если ппф A истинна в следующий момент (состояние); ппф $\Box A$ («always - всегда» A) истинна в данный момент, если и только если A истинна во все будущие моменты (состояния, включая текущее); ппф $\Diamond A$ («eventually - в конце концов» A) истинна в данный момент, если и только если A истинна в некоторый будущий момент. Строгая until ппф AUB истинна в данный момент (состояние), если и только если ппф B в конце концов будет истинна, т.е. в момент $s > n$, где n – текущий момент, и ппф A истинна для всех моментов t , таких что $n \leq t < s$.

Оператор W есть слабая версия оператора U , когда не гарантируется истинность ппф B в некоторый будущий момент. Темпоральные операторы прошлого времени определяются как строгая версия прошлого времени для соответствующих операторов будущего («будущих» двойников), т.е. прошлое не включает настоящий момент (состояние).

Семантика PTL. Для задания семантики PTL используется семантика возможных миров Крипке (Kripke). Возможный мир рассмат-

ривается как множество состояний во времени, связанных темпоральными отношениями из множества допустимых отношений R . Формально мир задается парой $A=(S,R)$, где S – непустое множество возможных состояний, R – бинарное отношение, $R \subseteq S \times S$.

Рассматривая L_P как множество атомных утверждений, модель мира можно определить как $M=(R,S,V)$, где V – функция означивания (valuation function), задающая отображение $V: S \times L_P \rightarrow \{T, F\}$, т.е. вычисляющая пропозициональное значение для каждого состояния $s \in S$. Вводя различные ограничения на отношение R , получают различные модельные структуры. Например, если ввести ограничение антирефлексивности ($<$), то получим дискретную модель. Для дискретных линейных моделей множество S можно рассматривать как последовательность состояний, R – как отношение следования (next) или преемника (successor). Интерпретация задается парой $\langle M, i \rangle$, где M – модель, а i – целое число, индексирующее состояние $s_i \in S$ в модели.

Семантика для темпоральной ппф задается с использованием отношения $\models$ между интерпретацией и ппф. Так утверждение $\langle M, i \rangle \models A$ означает, что ппф A интерпретируется в модели M как состояние с индексом i . Семантика для неограниченной дискретной линейной темпоральной логики задается следующим образом:

- Для утверждений:

$\langle M, i \rangle \models A \Leftrightarrow V(i, p) = T$ для $p \in L_P$, где символ $\Leftrightarrow$ используется как сокращение для «если и только если».

- Для связок:

$\langle M, i \rangle \models T$
 $\langle M, i \rangle \models \neg A \Leftrightarrow \text{not } \langle M, i \rangle \models A$
 $\langle M, i \rangle \models A \wedge B \Leftrightarrow \langle M, i \rangle \models A \text{ и } \langle M, i \rangle \models B$
 $\langle M, i \rangle \models A \vee B \Leftrightarrow \langle M, i \rangle \models A \text{ или } \langle M, i \rangle \models B$.

Семантика для связок $\Rightarrow$ и $\Leftrightarrow$ определяется посредством $\neg$ и $\vee$.

- Для темпоральных модальных операторов:

$\langle M, i \rangle \models OA \Leftrightarrow \langle M, i+1 \rangle \models A$
 $\langle M, i \rangle \models \Box A \Leftrightarrow (\forall j \geq i)(\langle M, j \rangle \models A)$
 $\langle M, i \rangle \models \Diamond A \Leftrightarrow (\exists j \geq i)(\langle M, j \rangle \models A)$
 $\langle M, i \rangle \models AUB \Leftrightarrow (\exists j \geq i)(\langle M, j \rangle \models B \text{ и } (\forall k)((i \leq k < j) \rightarrow \langle M, k \rangle \models A))$
 $\langle M, i \rangle \models \bullet A \Leftrightarrow i=0 \text{ или } \langle M, i-1 \rangle \models A$
 $\langle M, i \rangle \models \otimes A \Leftrightarrow i>0 \text{ и } \langle M, i-1 \rangle \models A$
 $\langle M, i \rangle \models \blacklozenge A \Leftrightarrow (\forall j, 0 \leq j < i)(\langle M, j \rangle \models A)$

$$\begin{aligned} \langle M, i \rangle \models \Diamond A &\leftrightarrow (\exists j, 0 \leq j < i) (\langle M, j \rangle \models A) \\ \langle M, i \rangle \models A \text{ } J \text{ } B &\leftrightarrow (\exists j, 0 \leq j < i) (\langle M, j \rangle \models B \text{ и } \\ (\forall k) ((j \leq k < i) \rightarrow \langle M, k \rangle \models A)) \\ \langle M, i \rangle \models A \text{ } Z \text{ } B &\leftrightarrow \langle M, i \rangle \models A \text{ } J \text{ } B \text{ или } \langle M, i \rangle \models \blacksquare A. \end{aligned}$$

Интуитивная семантика для модальных операторов определяется следующим образом:

- OA – ппф OA истинна в текущем состоянии, если и только если ппф A истинна в следующем состоянии;
- $\Box A$ – ппф $\Box A$ истинна в текущем состоянии, если и только если ппф A истинна во всех состояниях в будущем;
- $\Diamond A$ – ппф $\Diamond A$ истинна в текущем состоянии, если и только если ппф A истинна в каком-либо состоянии в будущем;
- AUB – строгая until ппф AUB истинна в текущем состоянии s_i , если и только если ппф B будет в конце концов истинна, например, в состоянии s_j , $j > i$, и ппф A истинна для всех состояний s_k , $i \leq k < j$;
- AWB – слабая until ппф AWB (называемая unless ппф) означает, что A должна быть истинной до состояния, когда B станет истинной.

Различие между ппф AUB и AWB состоит в том, что для истинности AUB требуется когда-нибудь в будущем истинность B , а для AWB этого не требуется. Модальные операторы прошлого времени (past-time modal operators) определяются как зеркальное отображение модальных операторов будущего. Однако, операторы $\blacksquare$ («всегда в прошлом») и $\Diamond$ («когда-нибудь в прошлом») интерпретируются как строгие, что означает, что текущий индекс не включается в их определение. Кроме того предполагается, что время в прошлом ограничено и имеет начало. Также различаются два других оператора прошлого времени $\otimes$ и $\bullet$. Ппф $\otimes A$ является ложной, в то время как ппф $\bullet A$ истинна, только когда интерпретация дается в начале времени. Ппф $\bullet \text{false}$ истинна, когда интерпретация дается для начала времени, в остальное время она ложна. Следовательно $\neg \bullet \text{true}$ можно использовать для определения начала (начального момента) времени. Справедливо: $\otimes \neg A \equiv \neg \bullet A$.

Понятия *выполнимости* (satisfaction) и *общезначимости* (validity) ппф определяются следующим образом:

- ппф A *выполнима в модели M* в состоянии s_i , если и только если $\langle M, i \rangle \models A$; и ппф A *выпол-*

нима в модели M , если и только если A выполняется в M в состоянии s_0 ;

- ппф A *общезначима* ($\models A$), если и только если A истинна во всех моделях M .

Доказано, что PTL разрешима ($M \models A$) [14].

Аксиоматика PTL. Аксиоматическая система для PTL включает следующие элементы:

- (A_0) – все элементы общезначимой схемы пропозициональной логики;
- аксиомы для оператора O :
 $(A_1): O(A \Rightarrow B) \Rightarrow (OA \Rightarrow OB)$ – схема для нормальной логики K ;
- $(A_2): \neg OA \equiv O \neg A$ – существование и уникальность преемника;
- аксиомы для операторов $\Box$ и $\Diamond$:
 $(A_3): \Diamond \equiv \neg \Box \neg A$ – аксиома двойственности;
- $(A_4): \Box (A \Rightarrow B) \Rightarrow (\Box A \Rightarrow \Box B)$ – K аксиома;
- аксиомы, описывающие связь между операторами $\Box$ и O :
 $(A_5): \Box A \Rightarrow (A \wedge OA \wedge O \Box A)$;
- $(A_6): \Box (A \Rightarrow OA) \Rightarrow (A \Rightarrow \Box A)$;
- аксиомы, характеризующие оператор U :
 $(A_7): AUB \Rightarrow (B \vee (A \wedge O(AUB)))$;
- $(A_8): (F \wedge \Box (F \Rightarrow (B \vee (A \wedge OF)))) \Rightarrow AUB$, где F – произвольная ппф.

Правилами вывода являются правила modus ponens (MP) и необходимости (N): (MP): если $\models A$ и $\models A \Rightarrow B$, то $\models B$; (N): если $\models A$, то $\models \Box A$.

Система аксиом для линейной PTL *непротиворечива* (sound) и *полна* (complete). Доказано, что ппф доказуема в аксиоматической системе линейной PTL, если и только если она общезначима в PTL [14].

В линейной дискретной PTL моделью времени является упорядоченная последовательность натуральных чисел. Следовательно, каждое состояние имеет одного и только одного преемника. В ветвящейся логике BPTL для каждого состояния не обязательно наличие единственного преемника и может быть множество возможных путей из любого данного состояния и, следовательно, возможно несколько различных «будущих». Моделью времени в BPTL является инфинитное дерево, но каждая вершина которого имеет финитное, ненулевое число преемников. Вершина дерева может рассматриваться как возможное состояние, а ветвь или путь – как история возможного мира. Допускается существование лишь одного прошлого, но разрешается будущему быть открытым.

Чтобы специфицировать состояния дерева, для которых ппф является истинной, необходимо идентифицировать путь, содержащий это состояние, и позицию состояния на этом пути. Таким образом делается различие между *формулой пути* (*path formulae*) и *формулой состояния* (*state formulae*). Формула пути является формулой PTL, а формула состояния указывает тот путь, на котором следует интерпретировать формулу пути, т.е. эта формула интерпретируется на данном состоянии пути. Например, если p есть утверждение и s есть состояние, то формула $\exists p$ будет истинной в интерпретации «если существует путь, порожаемый в s и на котором все состояния удовлетворяют p ». Обобщенно, если p есть формула пути, то $\forall p$ и $\exists p$ – формулы состояния. В состоянии s формула $\forall p$ интерпретируется как истинная, если формула p истинна для всех путей, ветвящихся (исходящих) из s , а формула $\exists p$ интерпретируется как истинная, если p истинна хотя бы на одном пути, исходящем из s .

Используя PTL, определены синтаксис и семантика BPTL. Формальная семантика BPTL определяется в терминах модельной структуры $M=(S, R, V)$, где S , R и V определяются аналогично PTL. Концепция ветвящегося времени требует введения условия линейности влево (в прошлое) и транзитивности R . Ппф BPTL являются формулами состояния, а формулы пути – вспомогательные объекты, вводимые для того, чтобы облегчить задание (выражение) семантики формул состояния.

Ппф является *выполнимой* (*consistent*), если она истинна в некотором состоянии при некоторой интерпретации, и *общезначимой* (*valid*), если она истинна в каждом состоянии при каждой интерпретации. Отрицанием общезначимой ппф является *невыполнимая* (*inconsistent*) ппф. Определена система аксиом K_b для BPTL. Доказано, что BPTL полна по отношению ко всем структурам ветвящегося времени, т.е. для любой ппф A справедливо $\vdash K_b A$, если и только если A истинна во всех структурах

4. Реализация темпоральной логики ветвящегося времени

Среди известных темпоральных логик с точки зрения практической применимости выделяются логики, построенные на основе парадигмы согла-

сования ограничений. Среди них наиболее простой и приемлемой по вычислительной сложности для применения в составе современных ИС является точечная темпоральная логика [7,11,18]. Несмотря на то, что в качественной точечной модели базовым примитивом является момент времени, а в множество базовых ограничений входят качественные ограничения, модель обладает большим потенциалом к расширению за счет реализации дизъюнктивных утверждений [19].

Построим на базе точечной темпоральной логики логику ветвящегося времени, применимую в составе современных ИС, включая ИСППР РВ [20-22]. Реализация алгоритмов вывода в точечной логике основывается на переходе к задаче согласования временных ограничений (ЗСВО) [21]. ЗСВО является конкретизацией более общей задачи согласования ограничений (ЗСО), что позволяет использовать для решения ЗСВО методы, применяемые для ЗСО [7].

ЗСВО можно определить набором $Z = (V, D, BTR, C)$ [11]:

- 1) $V = \{V_1, V_2, \dots, V_m\}$ – конечное множество временных переменных (моментов времени);
- 2) D – область значений временных переменных (множество целых чисел);
- 3) $BTR = \{r_1, r_2, \dots, r_n\}$ – конечное множество взаимоисключающих бинарных базовых временных ограничений, полное объединение которых является универсальным ограничением U (не накладывающим каких-либо ограничений);
- 4) $C = \{C_{ij} \mid C_{ij} = \{r_1, \dots, r_k\}; k > 0; r_1, \dots, r_k \in BTR; i, j \leq m\}$, конечное множество ограничений, где C_{ij} – ограничение над временными переменными V_i и V_j , интерпретируемое как $(V_i, r_1, V_j) \vee \dots \vee (V_i, r_k, V_j)$. Если C_{ij} состоит только из одного дизъюнкта, то оно называется *точным*.

Для решения задачи выполнимости (задачи SAT) необходимо найти множество не противоречащих друг другу ограничений $C^* = \{C_{ij}^* \mid C_{ij}^* = \{r_i\}, r_i \in C_{ij}\}$. Если такое множество построить нельзя, то ЗСВО является *несогласованной*. Если ЗСВО имеет по крайней мере одно решение, то она называется *согласованной*.

Ограничения между временными примитивами представлены в виде качественных бинарных ограничений, т.к. известно, что с их помощью можно представить любые ограничения более высокого порядка [7].

Основными операциями над временными ограничениями являются:

- 1) отрицание ($\neg$): $\neg L_{ij} = U \setminus L_{ij}$;
- 2) инвертирование ($\sim$): $\sim(r_1, \dots, r_k) = (\sim r_1, \dots, \sim r_k)$;
- 3) пересечение: $S \cap T = \{r \mid r \in S, r \in T\}$;
- 4) композиция: $T \bullet S = \{t_1, \dots, t_k\} \bullet \{s_1, \dots, s_q\} = \{t_1 \bullet s_1, t_1 \bullet s_2, \dots, t_k \bullet s_q\}$.

Известно [18], что множество всех возможных типов временных ограничений для двух временных примитивов, состоит из $2^{|BTR|}$ типов ограничений, замкнуто относительно операций $\neg$, $\sim$, $\cap$, $\bullet$ и образует *алгебру* временных ограничений.

ЗСВО называют *единичной* ЗСВО (ЕЗСВО), тогда и только тогда, когда в множество S входят только точные ограничения. При этом сама ЗСВО сводится к проверке непротиворечивости (согласованности) ограничений из множества S .

Задачу определения ограничения r , справедливого для переменных V_i и V_j , для которых задано ограничение $C_{ij} = \{r_1, \dots, r_k\}$, при $k > 1$, называют *задачей вычисления неточного ограничения* C_{ij} .

Ограничение C_{ij} *выполнимо* для переменных V_i и V_j тогда и только тогда, когда существует хотя бы одно решение ЗСВО, в котором C_{ij} является ограничением для этих переменных. *Минимальным ограничением* C_{ij}^{min} называется множество, состоящее только из выполнимых ограничений для V_i и V_j . ЗСВО называют *минимальной*, если все ее ограничения минимальны. Задачу вычисления минимальной ЗСВО называют *задачей поиска минимального представления* (задачей *MIN*). Известно, что для любой ЗСВО всегда можно найти эквивалентную минимальную или показать несогласованность ограничений.

Построим на основе определения ЕЗСВО ЗСВО для ветвящегося времени. Сначала определим на базе ЕЗСВО сценарий как $S_i = (V_i, C_i, S_j)$, $i = 0, 1, \dots$, где $S_0 = (V_0, D, BTR, C_0)$ – ЕЗСВО, интерпретируемая как начальный сценарий; S_i ($i > 0$) – наследуемый сценарий, который расширяет множество переменных V_j и множество единичных ограничений C_j сценария S_j , $j < i$, множествами V_i и C_i соответственно; D – область определения переменных (множество целых

чисел), BTR – множество базовых временных ограничений. Таким образом, сценарий является композиционной ЕЗСВО, которую можно построить по иерархии наследования, задаваемой в определении сценария. Для каждого сценария, исходя из определения ЕЗСВО, можно поставить задачи поиска минимального представления и проверки согласованности, т.е. можно говорить о согласованном или несогласованном сценарии и сценарии в минимальном представлении. Будем называть текущими сценариями такие сценарии, для которых не существует наследуемых сценариев.

Определим ветвящуюся ЗСВО как множество альтернативных сценариев, унаследованных от одного начального сценария S_0 , $VZ = \{S \mid S - \text{сценарий}\}$. Определим для ветвящейся ЗСВО следующие подзадачи:

- проверка согласованности – проверка существования как минимум одного текущего согласованного сценария $S_i \in VZ$;
- проверка истинности каких-либо утверждений для конкретного текущего сценария;
- преобразование всех ЕЗСВО, соответствующих согласованным сценариям, к минимальному виду;
- проверка истинности каких-либо утверждений для всех текущих сценариев или хотя бы для одного текущего сценария.

Рассмотрим пример (Рис. 3). Для ЗСВО имеется 16 сценариев, из которых 5 являются несогласованными. Следует отметить, что, исходя из практических соображений, следует допускать наследование только согласованных сце-

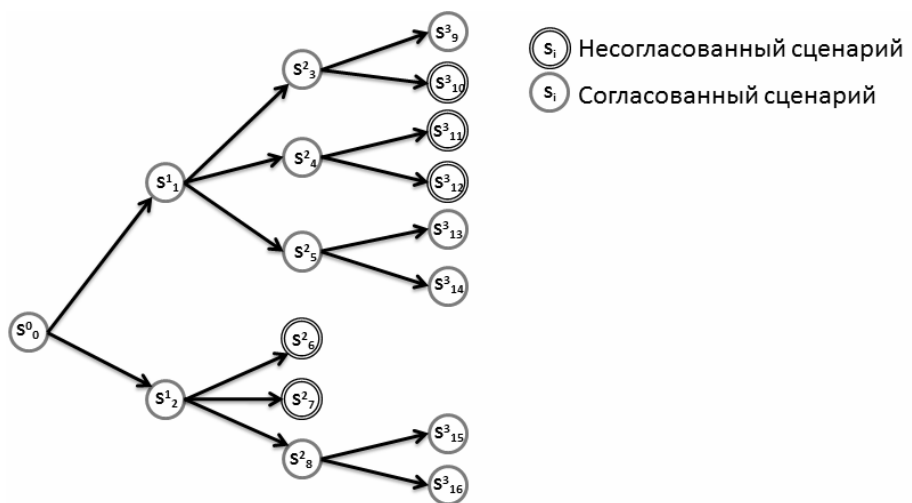


Рис. 3. Пример на ветвящуюся ЗСВО

нариев. Поэтому сценарии S_6^2 и S_7^2 не участвуют в порождении сценариев-наследников.

Таким образом, каждый сценарий соответствует возможному варианту развития событий, задаваемому соответствующим множеством ограничений над моментами времени.

Существуют разнообразные стратегии формирования сценариев в ветвящейся ЗСВО. Источниками ветвления на каждом шаге являются дизъюнктивные утверждения. Не дизъюнктивные утверждения добавляются к каждому текущему сценарию и не приводят к ветвлению. Если вносятся дизъюнктивные утверждения, то к каждому сценарию добавляется ряд наследников. Например, при добавлении в ветвящуюся ЗСВО ограничения типа $C_{ij} \vee C_{xy}$ для каждого текущего сценария должны быть добавлены три сценария: сценарий, в котором обязательно наличие ограничения C_{ij} , сценарий, в котором обязательно наличие C_{xy} и сценарий, в котором обязательно наличие как C_{ij} так и C_{xy} . Естественно, что при такой постановке ветвление является существенной проблемой, т.к. ведет к сильному росту числа возможных сценариев (особенно при большом числе дизъюнктов). Однако существует ряд приемов, которые могут быть использованы для ограничения ветвления. На практике может быть предусмотрен вариант внесения ограничений, связанных «исключающим или», т.е. в рассмотренном примере нет необходимости порождения сценария, в котором обязательно наличие комбинации C_{ij} и C_{xy} . Другой возможной стратегией является независимая достройка сценариев (когда каждый сценарий достраивается и разветвляется так, как это необходимо, исходя из задачи). В этом случае внесение дизъюнктивного ограничения приводит к разветвлению конкретно сценария и может не затронуть оставшиеся сценарии.

Алгоритмы решения задач ветвящейся ЗСВО могут быть построены на базе алгоритмов решения ЕЗСВО [11, 18, 21, 22]. В частности каждую ЕЗСВО в процессе решения ветвящейся ЗСВО можно решать с помощью перехода к задаче на графе времени (ТЛ-графе). Однако подобный подход не является оптимальным, т.к. на практике придется хранить в памяти ТЛ-граф и дополнительный Time-граф для каждого сценария. С учетом возможного роста числа сценариев предпочтительно по-

строить такое внутреннее представление задачи, которое позволяло бы экономить ресурсы за счет учета наследования сценариев, заложенного в само определение ветвящейся ЗСВО. Также при построении алгоритмов следует учесть, что создание новых сценариев на каждом шаге должно выполняться как можно быстрее, так как во многих приложениях типа ИСППР РВ большая глубина анализа (т.е. рассмотрение как можно большего числа сценариев) способствует получению более качественного решения. В работе [23] предлагаются алгоритмы, позволяющие экономить вычислительные ресурсы при последовательном дополнении ЕЗСВО ограничениями за счет анализа производимых изменений. В случае ветвящейся ЗСВО такие алгоритмы хорошо подходят для решения ЕЗСВО для каждого сценария-наследника $S_i=(V_i, C_i, S_i)$, так как наследники определяются как расширение множества ограничений в сценарии-предке. Коротко рассмотрим этот подход.

Для представления ЕЗСВО используется ТЛ-граф, определяемый как $G=(W, E, L)$, где W – непустое множество вершин (соответствующее моментам времени), E – множество направленных связей вида (w_i, l, w_j) , где $w_i, w_j \in W$, $l \in L$, $L=\{<, \leq\}$. Путем $\pi_{xy}=\langle x_0, x_1, \dots, x_k \rangle$ из вершины x в вершину y в графе G называется последовательность вершин, в которой $x_0=x$, $x_k=y$ и $(x_i, x_{i+1}) \in E$ для всех $0 \leq i < k$. В качестве веса пути примем $w(\pi_{xy}) = \bigcap_{i=0}^{k-1} w_{x_i x_{i+1}}$, где $w_{x_i x_{i+1}}$ – пометка ребра (x_i, x_{i+1}) . В случае если при вычислении веса w_{xy} в графе G существует направленная связь (y, l, x) , то в качестве w_{xy} принимается $\sim l$. Будем считать путь π_{xy} *существенным*, если $w(\pi_{xy}) \neq U$ и $w(\pi_{xy}) \neq \emptyset$.

Определим множество P всех направленных путей между всеми вершинами в графе G . Число возможных элементов в P не превышает значение $|P| = e^2$, где e – число связей, входящих в E . Определим функцию $\mathfrak{Z}: P \rightarrow \{true, false\}$, как $\mathfrak{Z}(\pi_{xy}) = (w(\pi_{xy}) \neq U) \wedge (w(\pi_{xy}) \neq \emptyset)$. Определим: P' – множество существенных путей в графе G , $P' \subseteq P$; множество входящих в вершину x путей $L_p(x) = \{\pi_{lx} \mid l \neq x, \pi_{lx} \in P'\}$ и множество выходящих из вершины x путей $R_p(x) = \{\pi_{xm} \mid m \neq x, \pi_{xm} \in P'\}$. Если в множество P входят все пути между всеми вершинами, то может быть построен алгоритм вычисления выполнимого ограничения между временными переменными (алгоритм 1), сложность которого оценивается

величиной $O(\max(|R_p(x)|, |R_p(y)|, |L_p(x)|, |L_p(y)|))$, при условии, что множества R_p и L_p могут быть вычислены за время $O(k)$, где k константа.

Алгоритм 1. Алгоритм вычисления выполняемого ограничения

Входные данные: x, y – моменты времени; P' – множество всех существенных путей в графе G .

Выходные данные: выполняемое ограничение между x и y .

```
01:  $ForwardPatches \leftarrow R_p(x) \cap L_p(y)$ 
02:  $RearwardPatches \leftarrow R_p(y) \cap L_p(x)$ 
03:  $R_{xy} = \bigcap_{\pi_{ij} \in ForwardPatches} w(\pi_{ij})$ 
04:  $R_{yx} = \bigcap_{\pi_{ij} \in RearwardPatches} w(\pi_{ij})$ 
05: return  $R_{xy} \cap R_{yx}$ 
```

Алгоритмы создания и удаления связей строятся таким образом, что их выполнение производится над множеством существенных путей, и что связи, которые могут привести к несогласованности, отсекаются на этапе внесения и не участвуют в порождении путей.

Ниже приводится алгоритм, применяемый для создания связей (алгоритм 2), сложность которого составляет $O(|R_p(y)| \cdot |L_p(x)|)$. Требуемый для работы объем памяти – $O(e^2)$. Существенным отличием данного алгоритма от рассмотренных ранее является то, что согласованность проверяется на этапе создания связей, поэтому несогласованность определяется сразу при внесении приводящих к ней связей.

Алгоритм 2. Алгоритм создания связей

Входные данные: (x, r, y) – связь для создания, $r \in \{<, >, \leq, \geq\}$; P' – множество всех существенных путей.

```
01:  $r' \leftarrow \text{Алгоритм 1}(x, y, P')$ 
02: if  $(r' \cap r = \emptyset)$  return inconsistent
03: if  $(r' \gg r)$  return // Существующее ограничение сильнее вносимого
04: foreach  $(\pi_{lx} \in L_p(x))$  {
05:   if  $(\exists(\pi_{ly})) P = P \cup \pi_{ly}$  //  $\pi_{ly}$  существенный
06:   foreach  $(\pi_{ym} \in R_p(y))$ 
07:     if  $(\exists(\pi_{lm})) P = P \cup \pi_{lm}$  //  $\pi_{lm}$  – существенный
08:   } // foreach  $\pi_{lx} \in L_p(x)$ 
09: foreach  $(\pi_{ym} \in R_p(y))$ 
10:   if  $(\exists(\pi_{xm})) P = P \cup \pi_{xm}$  //  $\pi_{xm}$  – существенный
```

Таким образом, алгоритмы создания связей в результате своей работы вносят в множество

существенных путей пути, которые образуются в TL-графе в результате создания связи (w, l, v) . Рассмотрим алгоритм удаления ограничений. Будем считать, что путь π_{xy} вложен в путь π_{ab} ($\pi_{xy} \subseteq \pi_{ab}$), тогда и только тогда, когда π_{ab} представим в виде $\pi_{ax}\pi_{xy}\pi_{yb}$. Ниже приводится алгоритм 3, применяемый для удаления ограничений. Вместе с удаляемым ограничением (x, r, y) удаляются все существенные пути, в которые элементарный путь π_{xy} входит в качестве подпути. Сложность алгоритма удаления ограничения существенно выше сложности алгоритма внесения ограничений и оценивается в худшем случае величиной $O(e^2)$, т.к. при этом требуется анализ всех путей в множестве P' , однако существует путь уменьшения этой величины при организации транзакционного режима

Алгоритм 3. Алгоритм удаления связей

Входные данные: (x, r, y) – связь для удаления; P' – множество всех существенных путей в графе G .

Выходные данные: P' – множество всех существенных путей в графе G .

```
01: foreach  $(\pi_{ab} : \pi_{xy} \subseteq \pi_{ab}, \pi_{ab} \in P')$ 
02:    $P' \leftarrow P' \setminus \pi_{ab}$ 
```

Пошаговые алгоритмы построены так, что они позволяют поддерживать в актуальном состоянии множество всех существенных путей на TL-графе G , соответствующему ЕЗСВО, после каждого изменения. Получаемый в результате пошагового построения TL-граф не содержит циклов, содержащих две вершины, связанные ограничением $<$ или $\neq$. В результате может быть получен только согласованный TL-граф. Основным преимуществом рассмотренного подхода является то, что после каждого изменения не требуется вызов дополнительных алгоритмов проверки согласованности и вычисления неявных ограничений.

Таким образом, для каждого текущего сценария можно хранить множество существенных путей и для достройки сценария использовать приведенные выше алгоритмы. Следует отметить, что применяемая внутренняя структура данных (множество существенных путей) может быть достаточно просто адаптирована для ветвящейся ЗСВО за счет хранения для каждого сценария-наследника $S_i = (V_i, C_i, S_j)$ только тех существенных путей, которые появились при

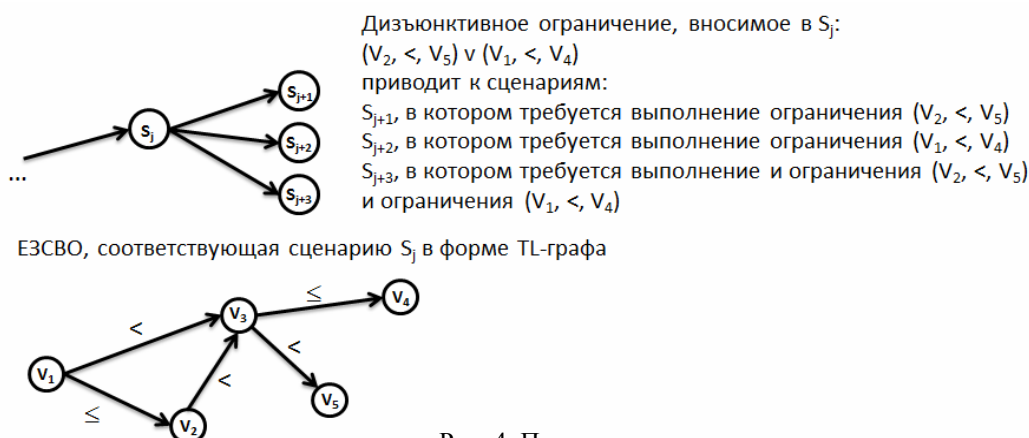


Рис. 4. Пример на ограничение ветвления

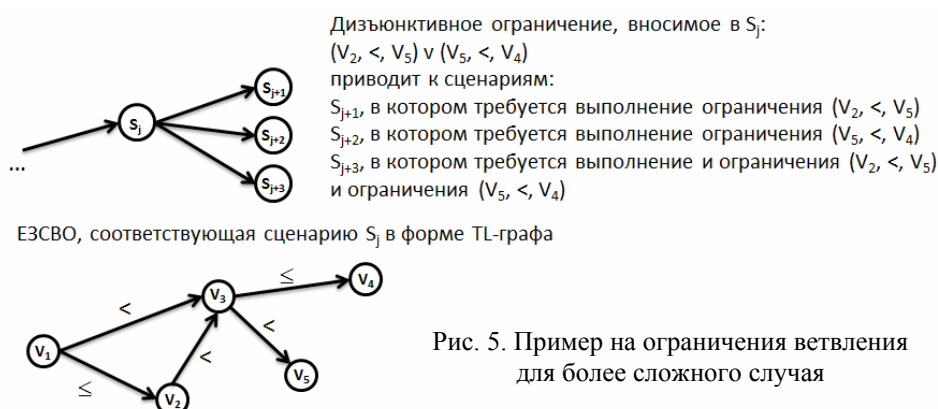


Рис. 5. Пример на ограничения ветвления для более сложного случая

дополнении ЕЗСВО сценария-предка S_j входящими в S_i ограничениями. Подход просто реализуется и позволяет значительно уменьшить объем памяти, требуемой для решения задачи.

Рассмотрим приемы, позволяющие уменьшить ветвление. В примере на Рис. 4 показана ситуация, когда в ветвлении нет необходимости. Следует отметить, что все три сценария S_{j+1} , S_{j+2} и S_{j+3} будут идентичны S_j , так как ограничения $(V_2, <, V_5)$ и $(V_1, <, V_4)$ следуют из ЕЗСВО, соответствующей сценарию S_j .

Более сложной является ситуация, когда в дизъюнктивном ограничении не все элементарные ограничения следуют из S_j . Но и в этом случае можно уменьшить ветвление. Рассмотрим пример на Рис.5. Сценарии S_{j+2} и S_{j+3} будут аналогичными, так как из ЕЗСВО, соответствующей S_j , следует ограничение $(V_2, <, V_5)$. Сценарий S_{j+1} идентичен S_j , но сохранен в качестве текущего сценария, чтобы не потерять вариант, когда выполнение ограничения $(V_5, <, V_4)$ не требуется.

Приведенные алгоритмы позволяют организовать решение ветвящейся ЗСВО и ее подзадач. В частности, т.к. при любых изменениях поддер-

живается множество текущих сценариев и для каждого сценария за счет применения приведенных ранее алгоритмов для получения множества существенных путей поддерживается актуальное решение соответствующей ЕЗСВО, автоматически решается подзадача проверки согласованности. Все ЕЗСВО, соответствующие согласованным сценариям, преобразованы к минимальному виду. В любой момент для любого из текущих сценариев можно проверить истинность какого-либо утверждения.

5. Программная реализация

Предлагаемые алгоритмы реализованы в СВР Point Time [21,22], построенной по модульному принципу с учетом требований расширяемости и простоты интеграции в ИС.

Чтобы обеспечить простое подключение СВР к более крупным приложениям, СВР организована в виде отдельного законченного модуля, который может общаться с системами-клиентами через интерфейс связи (Рис. 6). В этом модуле сконцентрированы механизмы

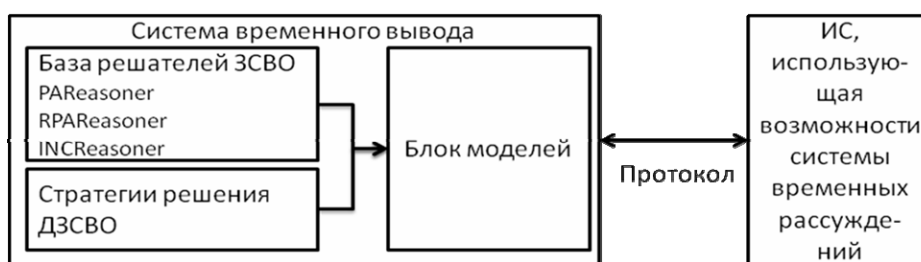


Рис. 6. Схема взаимодействия СВР с другими системами

представления и обработки информации о времени. Интерфейс связи в полной мере обеспечивает набор необходимых системе-клиенту возможностей для оперирования темпоральными зависимостями.

Язык СВР позволяет пользователю представлять знания просто и естественным образом, особенно в случае гнездовых (ветвящихся) темпоральных выражений. Язык обладает хорошо определенными синтаксисом и семантикой. СВР реализует унифицированный подход к представлению декларативных и процедурных знаний.

Рассмотрим принцип работы решателя ветвящейся ЗСВО, построенного на базе описанных ранее алгоритмов 1-3. Основными компонентами решателя являются: таблицы путей, словарь темпоральных (временных) переменных, блок транзакций и индексы (Рис. 7).

Словарь временных переменных хранит соответствие строковых имен переменных числовым кодам. Сценарии хранятся в списке сценариев. Каждый сценарий имеет свой идентификатор и ссылку на соответствующую ему таблицу путей. Каждая таблица путей хранит существенные пути и ссылку на родительскую таблицу (соответствующую сценарию-предку). Каждый путь представляется в виде записи в таблице и представляет собой совокупность следующих характеристик:

- *ID* – уникальный идентификатор пути (целое число);
- *S* – вершина, в которой начинается путь (целое число);
- *T* – вершины, в которую ведет путь (целое число);
- *W* – вес пути (вещественное число);
- *L* – индекс присоединенного пути слева;
- *R* – индекс присоединенного пути справа;
- *P* – порождающий путь.



Рис. 7. Компоненты пошагового решателя ЕЗСВО

Индексы служат для ускорения поиска по таблице путей (в частности, для ускорения выполнения запроса на поиск всех существенных путей, выходящих из вершины *v*) и ускорения поиска текущих сценариев.

Решатель работает по следующей схеме: каждая временная переменная кодируется целым числом, это соответствие помещается в словарь. Для внесения нового ограничения используется алгоритм, порождающий сценарии и существенные пути. Стек транзакций служит для поддержки транзакционного режима. С помощью применения транзакций можно разработать улучшенную версию алгоритма 3 удаления ограничений. Отличительной особенностью данного решателя является то, что он достаточно просто реализуем на основе современных промышленных СУБД. Среди программных средств, поставляемых совместно с СВР, можно выделить редактор сетей временных ограничений и монитор нагрузки СВР. Монитор позволяет оценить степень загрузки системы путем измерения таких параметров, как число обновлений, транзакций и возвратов в секунду, а также объем использованной оперативной памяти, количество временных переменных и ограничений и некоторые другие характеристики.

Заключение

В работе затронуты некоторые актуальные вопросы, связанные с реализацией методов моделирования рассуждений с учетом фактора времени для ИС типа ИСППР РВ. Проблема компьютерной реализации темпоральных логик, особенно логики ветвящегося времени, состоит в нахождении полиномиальных подклассов алгоритмов вывода, в общем случае характеризующихся NP- или экспоненциальной сложностью. Предложенная ветвящаяся темпоральная логика на базе точечной логики является менее выразительной, но практически реализуемой.

В настоящее время на кафедре прикладной математики МЭИ (ТУ) в плане разработки моделей, методов и базовых инструментальных средств конструирования ИСППР РВ на основе нетрадиционных логик разрабатывается СВР, интегрирующая различные модели представления темпоральных зависимостей (для метрического, интервального, смешанного представления времени, для линейной и ветвящейся структур времени, количественных и качественных зависимостей) применительно к прототипу ИСППР РВ для мониторинга и управления сложными техническими и организационными объектами. В рамках этой работы выполнена программная реализация базовых алгоритмов СВР с использованием среды разработки Microsoft Visual Studio 2010 (ОС Windows® и Linux). Разработанные программные средства прошли апробацию на задачах компании «ААМ Автоматик» [24].

Литература

1. Кандрашина Е.Ю., Литвинцева Л.В., Поспелов Д.А. Представление знаний о времени и пространстве в интеллектуальных системах. / Под ред. Д.А. Поспелова. М.: Наука. Гл. ред. физ.-мат. лит., 1989.
2. Вагин В.Н., Еремеев А.П. Некоторые базовые принципы построения интеллектуальных систем поддержки принятия решений реального времени // Изв. РАН. Теория и системы управления. 2001. № 6. – С. 114-123.
3. Поспелов Д.А. Логико-лингвистические модели в системах управления. - М.: Энергоиздат, 1981.
4. McCarthy J. M., Hayes P. J. Some Philosophical Problems from the Standpoint of AI. // Readings in AI. TP Co., Palo Alto, 1981.
5. Поспелов Д.А., Осипов Г.С. Введение в прикладную семиотику // Новости искусственного интеллекта. 2002. № 6. – С. 3-7.
6. Геловани В.А., Башлыков А.А., Бритков В.Б., Вязилов Е.Д. Интеллектуальные системы поддержки принятия решений в нештатных ситуациях с использованием информации о состоянии природной среды // – М.: Эдитореал УРСС, 2001.
7. Еремеев А.П., Троицкий В.В. Модели представления временных зависимостей в интеллектуальных системах поддержки принятия решений // Изв. РАН. Теория и системы управления. 2003. № 5. – С. 75-88.
8. Еремеев А.П., Куриленко И.Е. Реализация временных рассуждений для интеллектуальных систем поддержки принятия решений реального времени // Программные продукты и системы. 2005. № 2. – С. 8-16.
9. Виньков М.М., Фоминых И.Б. Формализмы немонотонных рассуждений на базе темпоральной линейной логики с часами // Тр. Междунар. конф. «Интеллектуальные системы» (IEEE AIS'03)» и «Интеллектуальные САПР» (CAD-2003). В 3-х томах. Том 1.– М.: Физматлит, 2003. – С. 236-242.
10. Еремеев А.П., Куриленко И.Е. Применение временных рассуждений в интеллектуальных системах реального времени / Интеллектуальные системы. Колл. монография. Выпуск второй. / Под. Ред. В.М. Курейчика. – М.: Физматлит. 2007. – С. 14-30.
11. Еремеев А.П., Куриленко И.Е. Применение темпоральных моделей в интеллектуальных системах / Интеллектуальные системы. Колл. монография. Выпуск третий. / Под. Ред. В.М. Курейчика. – М.: Физматлит. 2009. – С. 124-139.
12. Смирнов В.А. Логические системы с модальными временными операторами // Материалы II Советско-финского colloquium по логике «Модальные и временные логики». – М.: Институт философии АН СССР. 1979. – С. 89-98.
13. Логический подход к искусственному интеллекту: От модальной логики к логике баз данных: Пер. с франц. / Тейз А., Грибомон П., Юлен Г. и др. – М.: Мир, 1998.
14. Torsun I.S. Foundations of Intelligent Knowledge-Based Systems // ACADEMIC PRESS, London, 1998.
15. Еремеев А.П. Логика ветвящегося времени и ее применение в интеллектуальных системах поддержки принятия решений // Десятая национальная конференция по искусственному интеллекту с международным участием КИИ-2006 (25 - 28 сентября 2006 г., г. Обнинск): Тр. конф. В 3-х т. Т.3. М.: Физматлит, 2006. - С. 746-754.
16. Еремеев А.П., Куриленко И.Е. Логика ветвящегося времени и возможности ее реализации // Труды конгресса по интеллектуальным системам и информационным технологиям «AIS-IT'10». Научное издание в 4-х томах. – М.: Физматлит, 2010. – Т. 1/ - С. 130-139.
17. Карпов Ю.Г. MODEL CHECKING. Верификация параллельных и распределенных программных систем. – СПб.: БХВ-Петербург, 2010.
18. Gereviny A. and Schubert L. Efficient Algorithms for Qualitative Reasoning about Time. Technical report 496, Department of Computer Science, University of Rochester, Rochester, NY, 1993.
19. Еремеев А.П., Куриленко И.Е. Реализация механизма временных рассуждений в современных интеллектуальных системах // Изв. РАН. Теория и системы управления, 2007, № 2. - С. 120–136.
20. Еремеев А.П., Куриленко И.Е. «Моделирование временных рассуждений в интеллектуальных системах реального времени» // Вестник МЭИ. 2008. №1. - С. 114-123.

21. Еремеев А.П., Куриленко И.Е. Компонента временных рассуждений для интеллектуальных систем поддержки принятия решений реального времени // Искусственный интеллект и принятие решений. 2009. №1. - С. 31–45.
22. Куриленко И.Е. Система временного вывода для интеллектуальных систем поддержки принятия решений реального времени // Сб. док. междунар. научно-практ. конф. «Интегрированные модели и мягкие вычисления в искусственном интеллекте» в 2 т. – Т.1. – М.: ФизМатЛит. 2009. – С. 241 – 252.
23. Куриленко И.Е. О методах улучшения алгоритмов вывода для системы временных рассуждений // Сб. тр. IV–й междунар. научно–практической конф. Интегрированные модели и мягкие вычисления в искусственном интеллекте в 2 т. – Т.1. – М.:ФизМатЛит, 2007 – С.149–156.
24. Куриленко И.Е. Применение временной логики при построении интеллектуальной системы управления крупными парковочными комплексами // Сб. док. научно-практ. конф. «Интегрированные модели и мягкие вычисления в искусственном интеллекте» 2009 в 2 т. – Т.2 – М.:ФизМатЛит, 2009. - С. 171 -180.

Еремеев Александр Павлович. Заведующий кафедрой прикладной математики ГОУВПО «Московский энергетический институт (Технический университет)» (МЭИ (ТУ)). Окончил Московский энергетический институт (МЭИ) в 1972 году. Доктор технических наук, профессор, лауреат премии Президента РФ в области образования. Автор 217 печатных работ. Область научных интересов: искусственный интеллект, принятие решений, интеллектуальные системы поддержки принятия решений.

Куриленко Иван Евгеньевич. Доцент кафедры Прикладной математики ГОУВПО МЭИ (ТУ). Окончил ГОУВПО «Московский энергетический институт (Технический университет)» в 2005 году. Кандидат технических наук. Автор 45 печатных работ. Область научных интересов: интеллектуальные системы поддержки принятия решений, темпоральные модели, моделирование правдоподобных рассуждений.