А.П. Еремеев, Ю.И. Королев

Анализ и верификация моделей процессов в сложных динамических системах

Аннотация. В работе рассматриваются вопросы моделирования процессов в современных сложных динамических системах. Обосновывается потребность в учете темпоральных (временных) зависимостей между объектами таких систем на этапе моделирования. В качестве инструмента моделирования предлагается модификация раскрашенных сетей Петри с поддержкой темпоральной интервальной логики Аллена. Предложенный в работе подход позволяет адекватно моделировать процессы, протекающие в сложных динамических системах, а также проводить верификацию таких моделей. Он ориентирован на использование в интеллектуальных системах поддержки принятия решений реального времени.

Ключевые слова: искусственный интеллект, принятие решений, реальное время, динамическая система, темпоральная логика, моделирование процессов, сеть Петри.

Введение

В процессе системной интеграции компьютерных средств и коммуникационных технологий, характерном для современного общества, появляется потребность в системах, не только представляющих информацию, но и выполняющих ее предварительный анализ, способных давать полезные советы и рекомендации, осуществлять прогнозирование развития ситуаций, отбирать наиболее перспективные решения из множества альтернатив, т.е. поддерживать процесс принятия решений человеком (ЛПР - лицом, принимающим решения), взяв на себя значительную часть рутинных операций, а также функции предварительного анализа и оценок. Наиболее развитыми системами данного класса являются интеллектуальные системы поддержки принятия решений реального времени (ИСППР РВ) [1] и экспертные системы реального времени (ЭС РВ) [2], основанные на методах и технологии искусственного интеллекта. Понятно, что ИСППР, в том числе и ИСППР РВ, могут базироваться на экспертных знаниях, т.е. конструироваться как ЭС.

ИСППР РВ принято относить к динамическим интеллектуальным системам (ДИС) [3]. Понятие динамической системы вводится, если имеется необходимость в исследовании того, как система развивается во времени, путем установления взаимосвязи между значениями параметров системы в различные моменты времени. Необходимость использования методов и подходов искусственного интеллекта к исследованию поведения сложных динамических технических, организационных, транспортных, социальных, экологических и т.п. систем определяется тем, что зависимости настолько сложны, что не допускают традиционного аналитического представления состояния системы, описываются качественными параметрами, а законы поведения (функционирования) могут основываться на эмпирических и экспертных знаниях.

1. Моделирование процессов в динамических системах

О важности наличия средств представления времени и темпоральных (временных) зависи-

[1] Работа выполнена при финансовой поддержке РФФИ и в рамках проектной части государственного задания Минобрнауки РФ

мостей в интеллектуальных системах говорится практически с момента их появления [4,5]. Однако особенно актуально эта проблема встала в связи с появлением и развитием ДИС, в том числе ИСППР РВ, предназначенных для помощи ЛПР при управлении сложными объектами и процессами в условиях, как правило, достаточно жестких временных ограничений и при наличии различного рода неопределенностей (неполноты, нечеткости, противоречивости и т.п.) как в поступающей извне информации, так и в заложенных в систему знаниях. Одной из главных задач ИСППР РВ является помощь ЛПР для удержания объекта в штатном (нормальном) режиме функционирования. Для этого необходимо реализовать функции, определяемые современными методами и приемами управления в аномальных и критических ситуациях на основе моделей и методов теории аттракторов, теории устойчивости, методов искусственного интеллекта, позволяющих осуществлять поиск решения на основе знаний специалистов-экспертов и схем достоверного и правдоподобного вывода на базе неклассических логик, прецедентов, аналогий и т.д. [6, 7].

Итак, одной из главных задач при создании перспективных ИСППР РВ является задача представления и оперирования темпоральными зависимостями [8, 9]. Учет временного фактора необходим при решении задач диагностики и мониторинга сложного объекта или процесса, планирования действий для достижения поставленной цели, прогнозирования последствий принимаемых решений, управления в реальном времени, а также обучения (тренировки) ЛПР. Современные методы представления темпоральных зависимостей можно разбить на два основных класса - основанные на моделировании изменений во времени (типичный пример STRIPS-системы и ситуационное управление) и на явном моделировании времени. Второй класс обладает большими выразительными возможностями. Методы и модели этого класса можно различать по способу введения фактора времени. В основном это темпоральные логики, основу которых составляют либо модальные логики, если время учитывается в семантике языка путем усложнения интерпретации, либо модификации логики первого порядка, если время учитывается в синтаксисе. В качестве временных примитивов используются моменты (в точечных логиках) или интервалы (в интервальных логиках) времени. Алгоритмы вывода в темпоральных логиках, являющихся расширением логики предикатов первого порядка, могут базироваться на модификациях классических алгоритмов вывода в этих логиках, например, темпоральной резолюции, но эффективность этих алгоритмов вывода довольно низка для их практического применения.

Говоря о текущем состоянии разработок в области ИСППР и ИСППР РВ, следует отметить, что исследования в первую очередь направлены на создание универсальных или специализированных инструментальных средств с целью промышленной разработки ИСППР и ИСППР РВ для различных коммерческих и промышленных приложений. Напомним, что ИСППР РВ, будучи сами ДИС, ориентированы, как правило, на функционирование в динамических и открытых предметных областях и помощь ЛПР при управлении сложными динамическими объектами. Поэтому задача моделирования и исследования процессов и темпоральных закономерностей, определяющих функционирование сложных динамических систем, на сегодняшний день особенно актуальна.

Для исследования процессов, протекающих в системе, необходимо выбрать или создать соответствующую модель. Очевидно, что выбор формального аппарата определяет качество моделирования и глубину анализа процессов. В зависимости от назначения модели в общем случае можно отнести к одному из следующих типов:

- *функциональные модели*, отображающие закономерности функционирования объектов и используемые при проектировании систем для исследования принципов их работы и характера протекающих процессов;
- *структурные модели*, отображающие структурные характеристики объектов и используемые при конструировании систем в ситуациях, когда важными являются связи отдельных частей объектов между собой;
- *гибридные модели*, сочетающие (интегрирующие) возможности обоих перечисленных моделей и обладающие, как правило, большей выразительностью представления, что является важным фактором для их использования в ИСППР РВ.

Как отмечалось, учет темпоральных зависимостей между объектами и состояниями системы является необходимым условием для кор-

ректного моделирования сложной динамической системы. Темпоральные зависимости могут быть *количественными (метрическими),* когда для представления времени используются количественные меры на временной оси, и *качественными,* когда используется только относительное положение во времени событий или действий.

Для более полного моделирования процессов в сложных динамических системах необходимо использование обоих подходов. Представление зависимостей первого типа обычно не вызывает затруднений, однако подобный подход существенно ограничивает возможности моделирования, так как сложно напрямую отобразить темпоральные зависимости между разными объектами одной системы, а также учесть различного типа неопределенности в имеющейся информации (данных и знаниях), что специфично для ИСППР РВ, особенно для наиболее перспективных систем семиотического типа, способных к адаптации и развитию (модификации) [1, 7]. Для моделирования качественных зависимостей необходимо использовать аппарат темпоральных логик.

Таким образом, для моделирования процессов в сложных динамических системах применяемый формальный аппарат (на базе которого строится модель) должен обладать следующими ключевыми свойствами:

- интеграция функционального и структурного подходов;
- визуальная выразительность;
- оперирование с количественными и качественными темпоральными зависимостями, в том числе на основе темпоральных логик.

Выполнение первых двух условий позволяют обеспечить модели на основе графового представления информации об объектах и процессах (так называемые *графоориентированные* модели). В качестве таких моделей рассмотрим сети Петри и опишем их модификацию для возможности оперирования темпоральными зависимостями в плане использования в ИСППР РВ.

2. Модификации сетей Петри для оперирования темпоральными зависимостями

Сети Петри (СП) давно зарекомендовали себя как удобный, наглядный и в то же время математически строгий формализм для моделирования и анализа функционирования сложных систем [10]. Важным его достоинством является представимость многих поведенческих свойств. СП позволяют с необходимой степенью детализации моделировать вычислительные процессы, протоколы взаимодействия, процессы управления, позволяют естественным образом описывать синхронизацию, параллелизм, конфликты и причинную зависимость, а также наглядно представлять структуру и поведение систем.

Можно достаточно просто вводить необходимые для моделируемого объекта (процесса) дополнительные правила и условия в алгоритмы моделирования, получая ту или иную разновидность СП. В частности, была разработана теория *раскрашенных* (*colored*) сетей Петри (РСП) [11]. РСП – это графоориентированный язык для проектирования, описания, имитации и контроля распределенных и параллельных систем. Графическими примитивами показывается течение процесса, а специальными конструкциями языка имитируется необходимая обработка данных. Доказано, что для каждой РСП можно построить обычную СП, и наоборот. Однако на практике РСП представляют собой более компактный и удобный язык моделирования, чем обычные СП (аналогично тому, что языки программирования высокого уровня более удобны для практического программирования, чем ассемблерные языки).

Существенным недостатком как классических СП, так и РСП является отсутствие средств учета фактора времени и соответственно темпоральных зависимостей, что не позволяет эффективно моделировать те процессы, в которых от времени зависит состояние анализируемой системы (процесса). Поэтому были разработаны различные модификации СП, среди которых можно выделить три основных типа [12]:

- *темпоральные (временные) СП (ТСП)*, в которых темпоральные ограничения накладываются на переходы;
- *стохастические СП (ССП)*, основанные на концепции стохастических временных задержек;
- *функциональные СП (ФСП)*, для которых временные задержки определяются как функции аргументов.

Нетрудно установить, что ТСП и ССП являются частными случаями ФСП, поскольку

функции временных задержек позволяют задать как детерминированные и недетерминированные задержки в ТСП, так и стохастические задержки в ССП.

РСП реального времени (РСП РВ) [13] являются темпоральным подклассом РСП, ориентированным на моделирование и анализ систем реального времени. По сравнению с РСП в РСП РВ используется другая модель времени и приоритеты переходов, на которые наложены определенные структурные ограничения. Такие особенности РСП РВ позволяют осуществлять прямое моделирование элементов, типичных для параллельного программирования: приоритеты задач, таймауты и т.д. Формализм РСП РВ предполагает работу с количественными темпоральными зависимостями, однако, как уже отмечалось, выразительность представления увеличится при наличии средств, позволяющих выражать как количественные, так и качественные темпоральные зависимости. Поэтому для эффективного использования РСП РВ как основы для моделирования процессов в ИСППР РВ необходимо решить задачу представления и оперирования качественными темпоральными зависимостями. Для этого предлагается модифицировать РСП РВ, добавив возможность использовать ассоциируемые с местами сети темпоральные интервалы и аппарат темпоральных логик, а именно, активно используемую в искусственном интеллекте интервальную логику Аллена [14].

Логика Аллена характеризуется достаточной выразительностью и наличием полиномиальных алгоритмов вывода, что позволяет использовать ее в ДИС типа ИСППР РВ [9]. В качестве темпоральных примитивов в ней используются интервалы, что важно при моделировании сложных динамических систем, так как они позволяют, в отличие от точечных (оперирующих моментами) темпоральных логик, выражать информацию о положении дел не только в тот или иной момент времени, но и на том или ином временном интервале.

Пусть $B = \{b,bi,m,mi,o,oi,d,di,s,si,f,fi,e\}$ - множество базисных интервальных отношений логики Аллена [14]. Тогда атомарная формула логики Аллена – выражение вида XrY, где X и Y – интервалы, r – базисное отношение из B, а интервальная формула – выражение вида $\varphi = X\{r_1,r_2,\ldots,r_n\}Y$. Интервальная формула φ выполняется в некоторой интерпретации, если в ней выполняются все атомарные формулы Xr_iY, $1 \le i \le n$ (под интерпретацией понимается функция, отображающая интервалы на числовую ось).

Пусть x – переменная, $\Gamma(x)$ – функция, которая определяет тип переменной, т.е. множество всех допустимых значений, которые может принимать переменная x. Пусть y – некоторое выражение, $\nu(y)$ – множество всех переменных в выражении y, $\Gamma(y)$ – тип результата вычисления выражения y, т.е. множество всех возможных значений, которые могут быть получены при вычислении этого выражения.

Определим *РСП РВ с поддержкой темпоральной логики Аллена (РСП РВ ТЛА)* кортежем

$R = <\Sigma,P,T,A,C,G,I,E_M,E_S,M_0,S_0>$,

где Σ – непустое конечное множество непустых типов (множеств цветов), определяющее значения данных, операций и функций, которые могут использоваться в сетевых выражениях (т.е. выражениях дуг, защитных функций и функций инициализации); каждое такое множество имеет, по крайней мере, один элемент;

P – непустое конечное множество мест;

T – непустое конечное множество переходов, $P \cap T = \varnothing$;

$A \subseteq (P \times T) \cup (T \times P)$ – конечное множество дуг;

C: $P \to \Sigma$ – функция типа, помечающая каждое место p типом $C(p)$, т.е. каждый токен p должен иметь значение типа $C(p)$;

G – защитная функция, $\forall t \in T$: $(\Gamma(G(t)) \subseteq Bool, \Gamma(\nu(G(t))) \subseteq \Sigma \cup Int)$, ставящая в соответствие каждому переходу t логическое выражение, в котором все переменные имеют типы, принадлежащие множеству Σ, при этом защитные выражения, всегда возвращающие истину, опускаются;

I: $T \to N \cup \{0\}$ – функция приоритетов;

E_M – функция весовых выражений дуг, $\forall a \in A$: $(\Gamma(E_M(a)) \subseteq C(P(a)), \Gamma(\nu(E_M(a))) \subseteq \Sigma \cup Int)$, ставящая в соответствие каждой дуге выражение типа $C(P(a))$; это означает, что результат вычисления каждого выражения дуги должен быть типа того места, откуда выходит или куда входит эта дуга;

E_S – функция временных выражений дуг, $\forall a \in A$: $(\Gamma(E_S(a)) \subseteq Q \cup \{0\}, \Gamma(\nu(E_S(a))) \subseteq \Sigma \cup Int)$, ставящая в соответствие каждой дуге выражение, результатом вычисления которого является вещественное число – временная задержка перехода;

M_0 – начальная маркировка, $M_0(p) \subseteq 2^{C(p)}$;
S_0: $P \rightarrow Q$ – начальное значение функции временных меток.

Маркировка M определяется как функция на множестве мест P, $\forall p \in P$: $M(p) \subseteq 2^{C(p)}$. Функция временных меток S – функция на множестве мест P, $\forall p \in P$: $S(p) \subseteq Q$. Пусть P – упорядоченное множество. Заменим функцию временных меток вектором временных меток с $|P|$ элементами, тогда состоянием РСП РВ будет пара (M, S), где M – маркировка, а S – временной вектор.

3. Пример моделирования на основе РСП РВ ТЛА

Преимущества нового подхода можно проиллюстрировать, сравнив, например, модель системы экстренного торможения поезда, построенную с помощью аппарата РСП РВ (Рис. 1) [13], с аналогичной моделью, созданной на основе РСП РВ ТЛА [15]. В кабине машиниста каждые 60 секунд загорается световой сигнал, чтобы проверить, контролирует ли он идущий поезд. Если машинист проигнорирует световой сигнал, то через 6 секунд включается звуковой сигнал. Затем, если машинист не дезактивирует его в течение 3 секунд, срабатывает механизм аварийного торможения.

Модель РСП РВ для данного примера содержит шесть мест: *ContrSyst* - элемент, контролирующий систему; *Console* - консоль для отображения сигналов в кабине машиниста; *Brake* - механизм торможения; *Driver* - машинист поезда; *Timer*1; *Timer*2; и пять переходов- *TurnOnLS*: включение светового сигнала; *TurnOnSS* - включение звукового сигнала; *TurnOnBrake* - запуск механизма торможения; *Disactivate* - дезактивация машинистом сигналов; *Activity* - моделирование действий машиниста.

Задана начальная маркировка, начальные значения временных меток равны нулю и опущены. Переход *Disactivate* имеет приоритет 1, остальные переходы – 0 (опущены на схеме). Весовые и временные выражения дуг разделены знаком @. Если временное выражение равно 0, оно опущено. Каждая дуга с двумя стрелками заменяет для наглядности пару дуг.

Применение интервальной логики Аллена в РСП РВ ТЛА (Рис. 2) привело к сужению множества цветов Σ и разбиению сети на две несвязные подсети, одна из которых определяет работу механизма аварийного торможения, а другая моделирует действия машиниста. Формулы логики Аллена использованы в данном случае как защитные функции переходов *DisactLS* и *DisactSS*, обозначающих своевременную реакцию машиниста (ЛПР) на световой и звуковой сигнал соответственно:

- *Driver d LightSign* – переход *DisactLS* сработает, если токен в месте *Driver* появится на временном интервале, когда в месте *LightSign* будет находиться токен *on*;

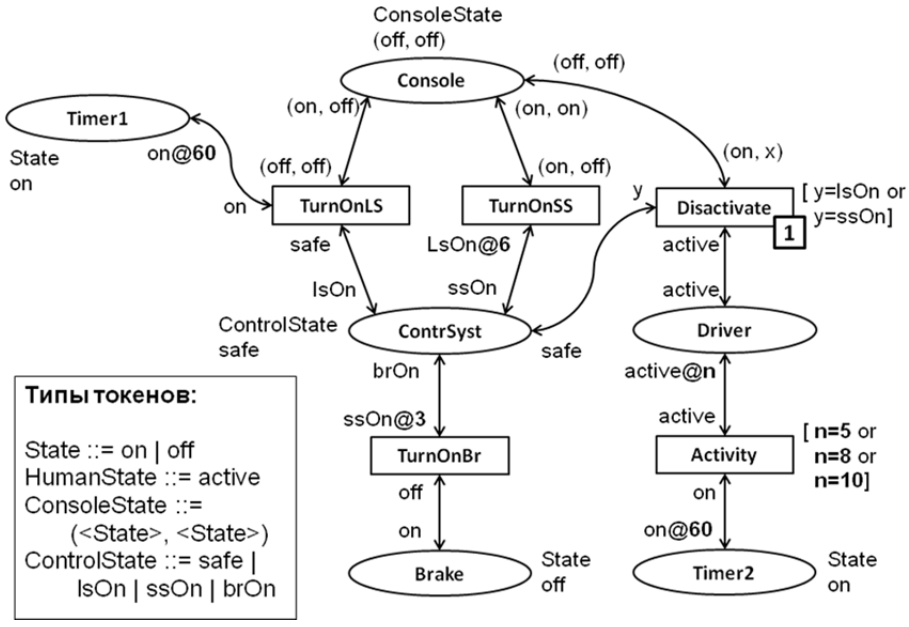

Рис. 1. Модель системы экстренного торможения поезда на основе РСП РВ

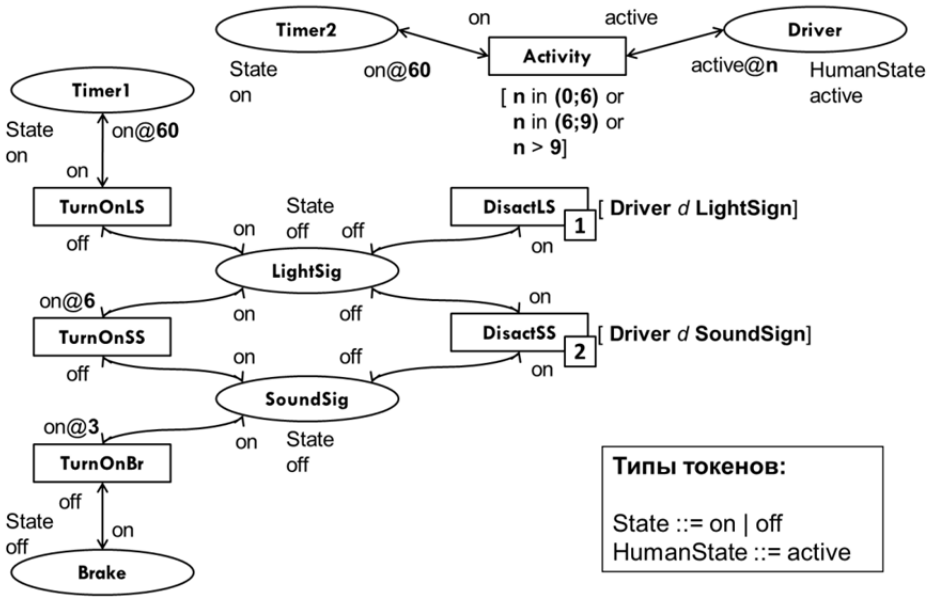

Рис. 2. Модель системы экстренного торможения поезда на основе РСП РВ ТЛА

▪ *Driver d SoundSign* – переход *DisactSS* сработает, если токен в месте *Driver* появится на временном интервале, когда в месте *SoundSign* будет находиться токен *on*.

Еще одним преимуществом РСП РВ ТЛА является возможность задавать не конкретное время реакции, как в случае РСП РВ, а интервалы, на которых ЛПР может дезактивировать систему и каждый из которых определяет дальнейшее поведение модели. Таким образом, включение в модель средств оперирования интервалами позволило корректно отразить неопределенность, присущую исходной задаче, что является одним из основополагающих принципов при проектировании перспективных ИСППР РВ.

4. Анализ моделей

Рассматриваемый подкласс РСП РВ ТЛА представляет собой, по сути визуальный язык программирования с формально определенным синтаксисом [15]. Модели, разработанные с помощью этого аппарата, кажутся полностью формализованными. Однако с точки зрения семантики - это не так. Из самой модели не следует непосредственно полное формальное описание ее поведения. Параллелизм, присущий СП в целом, и учет темпоральных зависимостей, введенный для расширения возможностей моделирования и упрощения процесса разработки моделей для ИСППР РВ, позволяют достаточно эффективно решить многие проблемы создания моделей процессов, протекающих в сложных динамических системах, но зачастую делают целостное восприятие ситуации сложным. Поэтому для анализа функционирования и верификации моделей, построенных с помощью РСП РВ ТЛА, необходимо использовать дополнительные инструменты.

Известны следующие основные группы методов анализа сетей Петри: основанные на построении графов изменения состояний; матричные методы, использующие уравнения сети и инварианты; методы редукции. При работе с раскрашенными и темпоральными сетями Петри последние две группы методов используются редко из-за высокой (по сравнению с классическими СП) сложности формального аппарата для подобных подклассов. Чаще в качестве основного инструмента анализа рассматриваются графы достижимости и покрытия.

Для описания средств анализа РСП РВ ТЛА определим формально ключевые понятия состояния и перехода между состояниями. В качестве примера вернемся к модели системы управления экстренным торможением поезда (Рис. 2). Состояние сети представляет собой пару (M,S), где M – маркировка - функция на множестве мест P, а S – временной вектор, ставящий в соответствие каждому месту сети число – временную метку. Для сети множество мест зададим следующим образом: $P = \{Timer1, LightSig, SoundSig, Brake, Timer2, Driver\}$. Тогда

начальное состояние сети (M_0,S_0) будет следующим: M_0 = (*on*,*off*,*off*,*off*,*on*,*active*), S_0 = (0,0,0,0,0,0).

Переход от одного состояния сети к другому может быть обусловлен двумя причинами:

- срабатыванием перехода $t \in T$ в подстановке b (подстановка – функция, которая замещает каждую переменную в защитной функции $G(t)$ и функциях весовых и временных значений дуг E_M, E_S, влияющих на переход t, значением соответствующего типа);
- течением времени - постепенное уменьшение каждой временной метки на фиксированную величину, пока не появится переход, который может сработать.

Следует отметить, что безусловный приоритет при смене состояний сети имеет событие срабатывания перехода. Течение времени позволяет только дожидаться момента, когда может сработать очередной переход. Для рассматриваемого примера в начальном состоянии могут сработать два перехода: *TurnOnLS* и *Activity*. Рассмотрим изменение состояния сети при срабатывании первого. На переходе *TurnOnLS* защитная функция всегда принимает значение *true*, поэтому переход срабатывает в тривиальной подстановке b=(). Результатом срабатывания перехода *TurnOnLS* в начальном состоянии будет состояние (M_1,S_1): M_1 = (*on*,*on*,*off*,*off*,*on*,*active*), S_1 = (60,0,0,0,0,0).

При срабатывании перехода фишки-токены извлекаются и помещаются в места, связанные с переходом, в соответствии со значениями весовых выражений дуг, временные метки входных мест обнуляются, а временные метки выходных мест определяются в соответствии со значением временных выражений дуг, идущих из перехода к этим местам. В соответствии с условиями, накладываемыми защитной функцией, переход *Activity* может сработать в трех различных подстановках: $b1=(x/n)$, $b2=(y/n)$ и $b3=(z/n)$, где $x \in (0;6)$, $y \in (6;9)$, $z>9$. Для примера возьмем x=5, y=8, z=10. Результатом срабатывания перехода *Activity* в подстановке $b2$ будет состояние (M_2,S_2): M_2 = (*on*,*on*,*off*,*off*,*on*,*active*), S_2 = (60,0,0,0,60,8).

Ни один переход не может сработать в этом состоянии. Необходимо подождать τ=6 секунд, чтобы временная метка в месте *Console* позволила сработать переходу *TurnOnSS* (машинист не реагирует на световой сигнал): S_2' = (54,-6,-6,-6,54,2).

После срабатывания перехода *TurnOnSS* в подстановке b=() сеть перейдет в состояние (M_3,S_3): M_3 = (*on*,*on*,*on*,*off*,*on*,*active*), S_3 = (54,0,0,-6,54,2).

Полученную последовательность смены состояний сети можно представить графически (Рис. 3).

Будем считать, что состояние (M',S) достижимо из состояния (M,S), если существует конечная последовательность переходов, начинающаяся с состояния (M,S) и оканчивающаяся состоянием (M',S). Обозначим $R(M,S)$ множество всех состояний, достижимых из состояния (M,S). Анализ РСП РВ ТЛА можно проводить, используя *граф достижимости (ГД)*, вершинами которого являются элементы множества $R(M_0,S_0)$, а каждая дуга отображает изменение состояния (M_i,S_i) на (M_j,S_j) по прошествии времени $\tau \geq 0$ и срабатывания перехода t в подстановке b.

Фрагмент ГД для последовательности смены состояний сети представлен на Рис. 4.

Анализ свойств сети может осуществляться с помощью маркировки узлов ГД и меток дуг.

$$(M_0,S_0) \xrightarrow{(TurnOnLS,(\,))} (M_1,S_1) \xrightarrow{(Activity,(8/n))} (M_2,S_2) \rightarrow$$
$$\xrightarrow{\tau=6} (M_2,S_2') \xrightarrow{(TurnOnSS,(\,))} (M_3,S_3) \longrightarrow \ldots$$

Рис. 3. Последовательность смены состояний сети

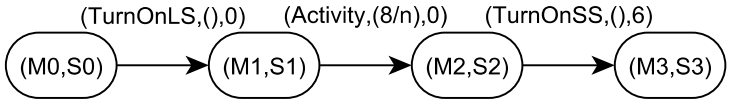

Рис. 4. Пример фрагмента графа достижимости

Каждая метка дуги представляет собой тройку, состоящую из перехода, его подстановки и значения временного промежутка перед его срабатыванием. Последний параметр позволяет определить время, затраченное на переход от одного состояния к другому. Используя алгоритмы поиска минимального и максимального пути между двумя узлами мультиграфа, можно найти минимальное и максимальное время перехода из одного состояния в другое.

Если продолжить строить ГД для анализируемой сети, можно убедиться, что из-за непрерывного уменьшения значений временных меток он оказывается бесконечным. Подобная ситуация возникает при анализе практически любой РСП РВ, в том числе РСП РВ ТЛА, причем степень связанности сети не влияет на конечность ГД. Очевидно, что в этом случае данный формализм неудобно использовать для анализа сетей. Введение дополнительных условий на временные метки позволяет трансформировать бесконечный граф в конечную структуру.

Одним из главных преимуществ РСП РВ ТЛА является возможность представления множества достижимых состояний с помощью конечного графа, построенного на основе отношения *покрытия (coverability)* (ГП), которое позволяет определять эквивалентные по своим характеристикам состояния сети. Считается, что два состояния покрываемы друг другом, если их маркировки совпадают, а временные метки либо совпадают, либо не превышают максимального времени доступа места $p \in P$, т.е. такого значения временной метки, когда токены становятся доступными для всех выходных переходов места p. При этом удовлетворяются условия рефлексивности, симметричности и транзитивности, т.е. отношение покрытия есть отношение эквивалентности (~) на множестве $R(M_0,S_0)$.

Графы ГД и ГП строятся одинаково. Различие заключается только в способе добавления новой вершины в графы. Для ГП после определения нового состояния сети необходимо проверить, есть ли в графе вершина, которая отображает состояние, покрываемое новым. Если есть, то необходимо добавить только новую дугу, которая идет к найденной вершине. В противном случае вершина нового состояния добавляется в граф вместе с соответствующей дугой. ГП содержит только одну вершину для каждого класса эквивалентности по отношению покрытия. ГП, построенный для анализируемой сети, приведен на Рис. 5.

Каждая вершина ГП помечена элементом фактор-множества $R(M_0,S_0)/\sim$, причем не существует двух или более вершин, помеченных одним и тем же элементом, а количество вершин ГП совпадает с мощностью множества $R(M_0,S_0)/\sim$. Конечность множества $R(M_0,S_0)/\sim$ обеспечивается следующими факторами:

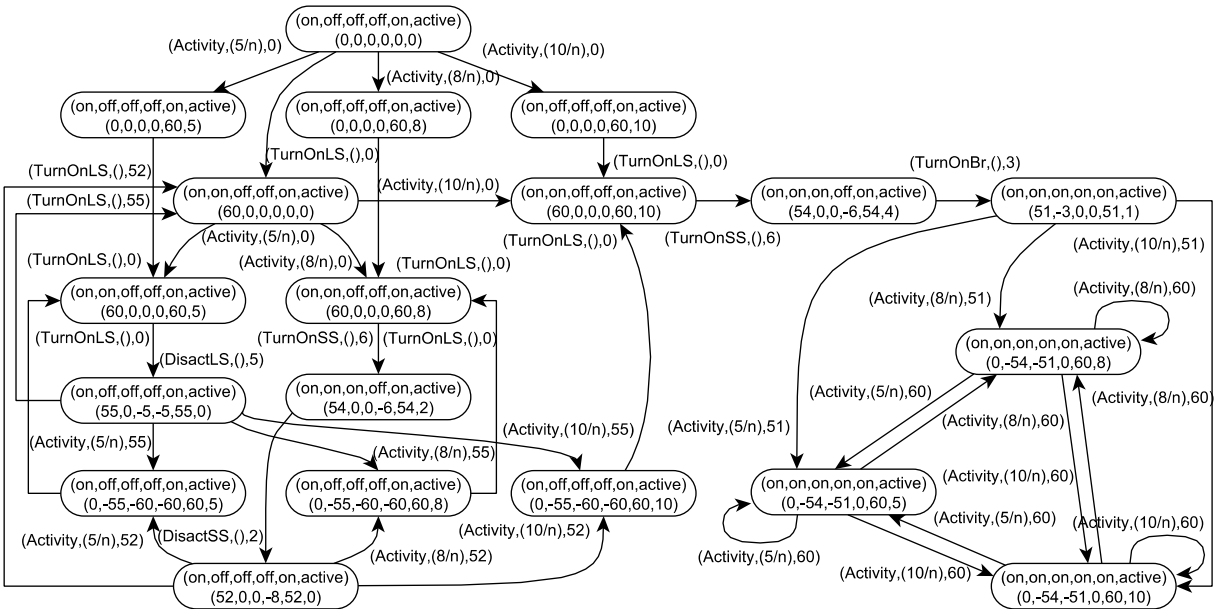

Рис. 5. Пример полного графа покрытия для анализируемой РСП РВ ТЛА

- множество всех возможных маркировок M на конечном множестве мест сети P конечно;
- максимальный возраст доступа каждого места $p \in P$ задается в общем случае вещественным числом.

Таким образом, ГП для РСП РВ ТЛА всегда конечен.

Граф ГП для РСП РВ ТЛА предоставляет такие же возможности анализа сетевых свойств, как и полный ГД. Чтобы найти минимальное и максимальное время перехода из одного состояния в другое, можно использовать те же алгоритмы, что и для ГД. Граф ГП позволяет увидеть все состояния сети с точностью до временных меток. Анализируя его, разработчик оценивает корректность выполнения поставленной задачи. Отметим, что ГП и для сравнительно малых РСП РВ ТЛА может достигать довольно больших размеров. Поэтому прямые исследования сетей путем их компьютерного моделирования могут упростить задачу разработчика.

В настоящий момент ведется разработка соответствующего программного обеспечения на языках высокого уровня, в которое планируется включить также возможность автоматической верификации и создания сложных систем иерархической структуры по принципу декомпозиции. Прототип инструментария для моделирования сложных динамических систем на основе РСП РВ и РСП РВ ТЛА, предназначенный как для включения в состав ИСППР РВ, так и для самостоятельного применения, разработан с помощью инструментального комплекса конструирования систем реального времени *G*2 (*Gensym Corp*., США) [15, 16].

5. Верификация моделей

Проблема обеспечения правильности программных и аппаратных компонентов систем управления и поддержки принятия решений приобретает сегодня первостепенное значение. Надежность и предсказуемость поведения таких систем зачастую являются более важными свойствами, чем производительность, модифицируемость и т.п.

Параллельные, распределенные и многопоточные программы, характерные для многих систем управления и ИСППР РВ, крайне подвержены ошибкам. Хорошо известно, что даже в тех случаях, когда функционирование каждой из параллельных взаимодействующих компонент системы абсолютно ясно, человеку трудно понять работу всей системы в целом. Известно, что такие системы, которые работают правильно «почти всегда», длительное время могут содержать «тонкие» ошибки, проявляющиеся в исключительных ситуациях. Их непосредственными причинами являются некорректные спецификации, неправильное понимание спецификации разработчиками, несогласованность параллельных ветвей процессов и многое другое. Очевидным и широко распространенным методом проверки правильности программных систем является тестирование – проверка работы построенной системы в различных ситуациях и при различных исходных данных. Однако в случае с параллельными системами обычно нельзя заранее определить все возможные траектории функционирования. Поэтому в качестве основного метода повышения качества разработки применяется верификация – формальная проверка того, что система (модель) удовлетворяет сформулированным заранее требованиям [17]. Методы верификации различаются в зависимости от того, какой аппарат лежит в основе проверяемой системы.

Для верификации технических систем свойства их поведения должны быть выражены формально логическими утверждениями, которые обеспечат их простую, лаконичную и недвусмысленную запись. Для спецификации таких свойств необходимы логические утверждения, истинность которых зависит от времени. Обычная логика высказываний является плохо пригодной для формулировки утверждений о поведении сложных динамических систем при изменении их состояний во времени. Формализация даже простейшего примера типа «Любой посланный запрос когда-нибудь позже будет обслужен» с помощью логики предикатов первого порядка приводит к громоздкому утверждению [18]:

$$(\forall t \geq 0)\,(\text{Послан}(\text{Запрос},t) \rightarrow (\exists t' > t)(\text{Обслужен}(\text{Запрос},t'))).$$

Поэтому при верификации темпоральных конструкций используются выражения темпоральных логик. Если нет необходимости подробно описывать закономерности поведения системы и взаимодействие ее объектов, целесообразнее применять не сложные интервальные логики, а простые расширения обычной логики высказывания. Традиционно при верификации

используются темпоральные логики линейного времени *LTL* (*Linear Temporal Logic*) [18] и ветвящегося времени *CTL* (*Computational Tree Logic*) [19].

Перспективным методом верификации РСП РВ ТЛА является метод *Model Checking* (*MC*) [17]. Другие методы в общем случае не могут быть полностью автоматизированы, что негативно сказывается на возможности их практического применения для ИСППР РВ. С другой стороны, исследования в области *MC* привели в последнее время к разработке эффективных алгоритмов верификации, позволяющих проверять реальные, разрабатываемые промышленностью программно-аппаратные системы. В частности, в работе [17] приводятся алгоритмы *MC*, позволяющие проверить, что формула темпоральной логики *LTL* или *CTL*, выражающая некоторое свойство поведения динамической системы во времени, выполняется (является истинной) на модели системы с конечным числом состояний, в качестве которой используется структура Крипке, формально задаваемая пятеркой $K = (W, W_0, H, AP, L)$,

где W – конечное непустое множество состояний;

$W_0 \subseteq W$ – непустое множество начальных состояний;

$H \subseteq W \times W$ – множество переходов, удовлетворяющее требованию: $(\forall w \in W)(\exists w' \in W)((w,w') \in H)$;

AP – конечное множество атомарных предикатов;

L: $W \to 2^{AP}$ – функция пометок, сопоставляющая каждому состоянию множество истинных в нем атомарных предикатов.

Стандартными шагами доказательства того, что поведение моделируемой системы обладает некоторым свойством, являются следующие:

1) для верифицируемой системы строится соответствующая структура Крипке, т.е. система переходов с конечным числом состояний; поведение реальной системы представляется разверткой - деревом вычислений для построенной структуры;

2) с помощью переменных и параметров верифицируемой системы строятся интересующие разработчика атомарные предикаты – логические выражения, которые могут принимать значения «истина» или «ложь» в каждом состоянии системы;

3) проверяемое свойство выражается формулой темпоральной логики *LTL* или *CTL* с использованием атомарных утверждений, темпоральных операторов и кванторов;

4) проверяется с помощью полностью автоматизированной процедуры истинность утверждения, что структура Крипке является моделью формулы.

Нетрудно убедиться, что рассмотренный выше формализм анализа ГП для РСП РВ ТЛА является структурой Крипке, так как:

- W соответствует множество вершин ГП;
- W_0 соответствует вершина начального состояния (M_0,S_0);
- H соответствует множество переходов между вершинами ГП, требование существования перехода из любой вершины обеспечивается правилом построения ГП;
- AP соответствует конечное фактор-множество $R(M_0,S_0)/\sim$, содержащее по одному элементу для каждого класса эквивалентности по отношению покрытия состояний РСП РВ ТЛА;
- L соответствует функция пометок вершин ГП элементами множества $R(M_0,S_0)/\sim$.

Таким образом, верификация РСП РВ ТЛА с помощью метода МС является естественным расширением начального анализа сетей с помощью графов состояний. Обобщенная схема процесса верификации приведена на Рис. 6.

Общепризнано, что ни тестирование, ни верификация по отдельности не могут гарантировать достаточного уровня правильности (корректности) разрабатываемых моделей. Существует множество примеров, когда в тщательно проверенных и оттестированных реализациях с помощью верификации впоследствии обнаруживались ошибки. С другой стороны, нельзя надеяться и только на верификацию. Часто причина ошибок кроется в том, что при разработке и доказательстве неявно выдвигаются неправильные предположения о функционировании системы, а значит, используются неадекватные формальные модели.

Следовательно, как тестирование, так и верификация обладают своими преимуществами и недостатками, поэтому эти подходы нужно считать взаимодополняющими и использовать совместно, особенно если речь идет о разработке и функционировании ДИС типа ИСППР РВ.

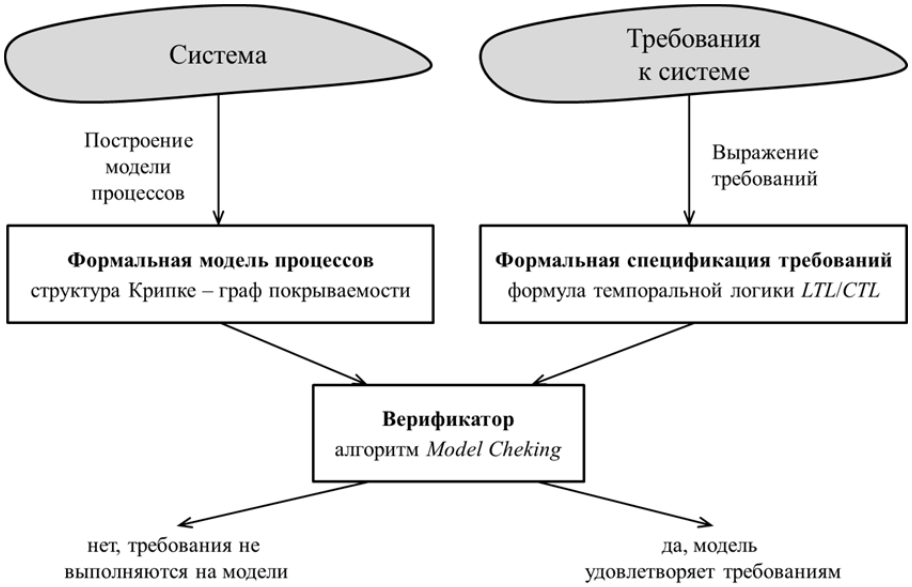

Рис. 6. Схема процесса верификации на основе алгоритма Model Checking

Заключение

Рассмотренные в работе подходы и методы позволяют достаточно эффективно моделировать процессы, протекающие в сложных динамических объектах и системах, в том числе и в ДИС типа ИСППР РВ. Приведены алгоритмы для анализа и верификации таких моделей – построение соответствующих графов ГД и ГП, метод *Model Checking*. Созданный в среде *G*2 для разработки ЭС РВ прототип показал возможность компьютерного моделирования процессов для ИСППР РВ [15], однако высокая стоимость комплекса *G*2 (как и других подобных систем) приводит к необходимости создания отдельного приложения для разработки моделей. В настоящий момент ведется разработка такого программного комплекса, в котором предусмотрена возможность автоматической верификации моделей.

Литература

1. 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.
2. Рыбина Г.В. Теория и технология построения интегрированных экспертных систем. М.: Научтехлитиздат, 2008.
3. Осипов Г.С. Динамические интеллектуальные системы // Искусственный интеллект и принятие решений, 2008, № 1. С. 47-54.
4. Shoham Y. Reasoning About Change: Time and Causation from the Standpoint of Artificial Intelligence. - The MIT Press: Cambridge, MA, 1988.
5. 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.
6. Вагин В.Н., Головина Е.Ю., Загорянская Н.А., Фомина М.Б. Достоверный и правдоподобный вывод в интеллектуальных системах / Под ред. В.Н. Вагина, Д.А. Поспелова. – 2-е изд., испр. и доп. - М.: ФИЗМАТЛИТ, 2008.
7. Вагин В.Н., Еремеев А.П. Исследования и разработки кафедры прикладной математики по конструированию интеллектуальных систем поддержки принятия решений на основе нетрадиционных логик // Вестник МЭИ, 2008, № 5. С. 16-26.
8. 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.
9. Еремеев А.П., Куриленко И.Е. Средства темпорального вывода для интеллектуальных систем реального времени // В кн.: Интеллектуальные системы. Коллективная монография. Выпуск 4. -М.: Физматлит, 2010. С. 222-252.
10. Котов В.Е. Сети Петри. М.: Наука, 1984.
11. Jensen K. Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Vol 1-3. Springer-Verlag, 1992-1997.
12. Еремеев А.П., Королев Ю.И. Реализация интеллектуальных систем реального времени на основе сетей Петри с поддержкой темпоральных зависимостей // Программные продукты и системы, 2013, № 3. С. 88-94.
13. Szpyrka M. Modelling and Analysis of Real-Time Systems with RTCP-Nets // In: Petri Net, Theory and Applications. I-Tech Education and Publishing, 2008. P. 17-40.
14. Allen J.F. Maintaining knowledge about temporal intervals // Communications of the ACM, 1983, Vol. 26, No. 11. P. 832-843.

15. Еремеев А.П., Королев Ю.И. Инструменты и методология разработки интеллектуальных систем реального времени на основе цветных сетей Петри // Вестник Ростовского государственного университета путей сообщения, 2013, № 3. С. 53-60.
16. Еремеев А.П., Королев Ю.И. Темпоральные сети Петри и их применение в интеллектуальных системах поддержки принятия решений реального времени // International Journal "Information models and analyses", 2013, V. 2, № 4. P. 336-344.
17. Карпов Ю.Г. Model Cheking. Верификация параллельных и распределенных программных систем. - СПб.: БХВ-Петербург, 2010.
18. Pnueli A. The temporal logic of program // Proc. of the 18th Anny. Symp. on Foundation of Computer Science. 1977. P. 46-57.
19. Clarke E.M., Emerson E.A., and Sistla A.P. Automatic verification of finite-state concurrent systems using temporal logic specifications // ACM Trans. Program. Lang. Syst., 1986, Vol. 8, No. 2. P. 244-263.

Еремеев Александр Павлович. Заведующий кафедрой прикладной математики Национального исследовательского университета «Московский энергетический институт» (НИУ «МЭИ»). Окончил МЭИ в 1972 году. Доктор технических наук, профессор, лауреат премии Президента РФ в области образования. Автор более 280 печатных работ, в том числе 6 монографий. Область научных интересов: искусственный интеллект, принятие решений, интеллектуальные системы поддержки принятия решений, экспертные системы. E-mail: eremeev@appmat.ru

Королев Юрий Ильич. Аспирант кафедры прикладной математики Национального исследовательского университета «Московский энергетический институт» (НИУ «МЭИ»). Окончил МЭИ в 2012 году. Автор 24 печатных работ. Область научных интересов: искусственный интеллект, системы поддержки принятия решений, моделирование процессов, темпоральные зависимости. E-mail: korolevyu@gmail.com