СИТУАЦИОННО-СОБЫТИЙНЫЙ ПОДХОД К СПЕЦИФКАЦИИ ГИБРИДНЫХ ПРОЦЕССОВ

В. М. Шпаков♦

Санкт-Петербургский институт информатики и автоматизации РАН

СПИИРАН, 14-я линия ВО, д. 39, Санкт-Петербург, 199178

<vlad@iias.spb.su>

УДК 681.3.06

Шпаков В. М. **Ситуационно-событийный подход к спецификации гибридных процессов** // Труды СПИИРАН. Вып. 4. — СПб.: Наука, 2007.

Аннотация. *Рассматривается подход к разработке исполняемых спецификаций совокупностей гибридных процессов, основанный на представлении функций перехода дискретных состояний с помощью правил трансформации ситуаций. Обсуждаются возможные структуры правил с точки зрения их выразительности и пригодности для создания надежных спецификаций и быстродействующих исполняющих процедур. Свойства правил иллюстрируются на примерах спецификации простого гибридного процесса.* — Библ. 3 назв.

UDC 681.3.06

Shpakov V. M. **A Situation-Event Approach to Hybrid Processes Specifications** // SPIIRAS Proceedings. Issue 4. — SPb.: Nauka, 2007.

Abstract. *An approach to development of hybrid process collection executable specifications is considered. The approach is based on presentation of discrete state transition function with help of situation transformation rules. Possible structures of the rules are discussed in the view of their expressiveness and their fitness for creation of reliable specifications and high-performance executing procedures. The rules properties are shown by the examples of simple hybrid process specifications.* — Bibl. 3 items.

1. Введение

Функционирование современных технических, в том числе многих производственных, систем определяется большими совокупностями взаимодействующих процессов различной динамики. Различают процессы непрерывные, дискретно-событийные и гибридные. Все процессы могут быть представлены с помощью переменных состояния. В случае непрерывного процесса изменения носят непрерывный характер и в качестве переменных состояния используются вещественные переменные. В случае дискретно-событийных процессов изменения состояния происходят мгновенно в дискретные моменты времени и могут носить качественный характер. Для их описания используются символьные или лингвистические переменные. В гибридных процессах происходят изменения обоих указанных типов, причем изменения одного типа могут влиять на характер изменений другого типа. В гибридных процессах различают дискретные изменения двух типов. Первые связаны с мгновенными изменениями динамики непрерывной составляющей процесса. Они происходят, например, при включении и выключении нагревателя в системах термостатирования. Такие «качественные» дискретные состояния гибридных процессов называются режимами. Изменения второго типа определяются мгновенными дискретными изменениями непрерывного состояния процесса. Примером процесса такого типа может служить прыгающий мячик. Часто причинами изменений дискретных состояний

♦ Данная работа была частично поддержана грантом Российского фонда фундаментальных исследований за 2005 год (проект № 05-08-18111-а).

событийного процесса и режимов гибридного процесса являются выходы его непрерывных состояний за пределы заданных ограничений. Спецификация таких причинно-следственных отношений основывается на вычислении предикатов от непрерывных состояний. Очевидно, что определение гибридного процесса включает в себя определения дискретно-событийного и непрерывного процессов, которые являются его частными случаями. В первом случае отсутствуют непрерывные изменения, во втором — дискретные.

Необходимость спецификации процессов возникает при разработке программных компонент имитационных моделей динамических систем и при компьютерной реализации управляющих частей систем управления. При этом наибольший интерес представляют исполняемые спецификации процессов, не требующие перекомпиляции после внесения изменений и тем самым делающие удобным итеративный подход к разработке и отладке процессов.

Спецификация гибридного процесса разбивается на две взаимосвязанные части: спецификацию дискретной и непрерывной составляющих процесса. Спецификация непрерывной составляющей связана с реализацией интегро-дифференциальных и функциональных зависимостей для каждого режима. Спецификация дискретных составляющих гибридных процессов основана на формировании правил трансформации дискретных состояний и режимов и использовании исполняющей процедуры. Трансформационные правила определяют для состояний процесса отношения следования. Исполняющая процедура в цикле сканирует правила и вычисляет эти отношения. Трансформационные правила имеют вид: `Условие` $\rightarrow$ `Действие`. Выразительные возможности трансформационных правил, а также надежность и эффективность получаемых на их основе исполняемых спецификаций процессов зависят от структуры как условной, так и исполнительной частей правил.

В СПИИРАН разработан исследовательский образец среды имитационного моделирования совокупностей взаимодействующих процессов различной динамики (EnviCon), с помощью которого проводятся исследования структур трансформационных правил. Ниже приводятся возможные структуры правил и их оценки с точки зрения выразительности и надежности получаемых на их основе спецификаций процессов и эффективности исполняющих процедур. Под надежностью спецификаций мы здесь будем понимать только оценку возможности формирования противоречивых правил. Для удобства дальнейшего изложения приведем математическую модель совокупности взаимодействующих процессов [1,2].

2. Математическая модель совокупности процессов различной динамики

Текущее состояние совокупности гибридных процессов может быть задано множеством вещественных переменных $X = \{(x_1,...,x_n) \mid (x_1,...,x_n) \in R^n\}$, представляющих непрерывные составляющие, и множеством символьных переменных W, представляющих дискретные составляющие процессов. Обычно при спецификации процессов в этих множествах выделяют подмножества, различающиеся по способу реализации их элементов. Так, всегда выделяют внешние независимые дискретные и непрерывные воздействия. В составе множества W выделим подмножество внешних воздействий V и подмножество Q, содержащее переменные для представления состояний дискретно-событийных процес-

Труды СПИИРАН. 2007. Вып. 4. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2007. Issue 4. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

сов и режимов гибридных процессов, спецификация которых производится одинаковым способом. Кроме того, в множество W включают подмножество G предикатов от непрерывных состояний, которые могут определять состояние дискретных и режимы гибридных процессов. Таким образом, множество символьных переменных $W = V \bigcup Q \bigcup G$. Нас в данном случае будут интересовать только символьные переменные $q \in Q$.

Для спецификации процессов необходимо задать функции переходов следующих типов:

$\sigma : W \to Q$ — функция трансформации состояний дискретно-событийных процессов и режимов гибридных процессов;

$\delta : W \times X \to X$ взаимозависимость непрерывных состояний для возможных режимов гибридных процессов;

$\gamma : X \to G \times \{False, True\}$ — зависимость значений предикатов от непрерывных состояний процессов. С учетом этого модель процессов может быть представлена в виде следующего кортежа: $(V, Q, G, \sigma, \gamma, X, \delta, q_0, Init)$, где $q_0, Init$ — множества дискретных и непрерывных начальных состояний соответственно.

Реализация функции γ связана с вычислением неравенств от непрерывных состояний. Реализация функции изменений непрерывных состояний δ обычно производится с помощью методов численного решения дифференциальных уравнений. В [3] рассмотрен достаточно эффективный подход к реализации δ, основанный на формировании динамических структурных схем из элементарных динамических звеньев и реализации этих звеньев с помощью транзитивных процедур. Здесь будут рассмотрены только способы реализации функции трансформации дискретных состояний и режимов σ. Поскольку способы их вычисления совпадают, будем для удобства в дальнейшем любые элементы множества Q называть переменными состояния.

В общем случае внешние дискретные воздействия и переменные состояния могут быть представлены символьными переменными. Те из них, которые имеют двухэлементные множества значений (*открыт – закрыт*, *включен – выключен*), могут быть представлены логическими переменными. Использование логических переменных позволяет существенно упростить исполняющую процедуру. Символьную переменную с более чем двумя значениями всегда можно представить с помощью формулы нескольких логических переменных. С учетом этого будем в дальнейшем считать, что все элементы множества W являются логическими переменными.

При использовании логических переменных формирование функции перехода можно производить с помощью логических связок между входными переменными и переменными состояния, т.е. описывать область определения функции с помощью логических формул, что делает описание более наглядным. В принципе, могут быть использованы произвольные логические формулы, однако, на наш взгляд, наиболее удобно использовать элементарные конъюнкции входных воздействий, состояний и режимов и предикатов от непрерывных состояний. Такие конъюнкции логических переменных интуитивно понятным образом могут интерпретироваться как динамические ситуации. При этом любое многозначное (составное) состояние может быть представлено элементарной конъюнкцией нескольких логических переменных, то есть некоторой ситуацией.

Элементарная конъюнкция всех логических переменных спецификации процесса определяет глобальную динамическую ситуацию. Глобальные ситуа-

Труды СПИИРАН. 2007. Вып. 4. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2007. Issue 4. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

ции различаются по тому, какие из переменных представлены в конъюнкции своими значениями, а какие — их отрицаниями. Общее количество различных глобальных ситуаций равно 2^{N_W}, где $N_W = |W|$. Элементарные конъюнкции, составленные из элементов подмножеств W, можно назвать локальными динамическими ситуациями.

Процедуру, порождающую множество локальных динамических ситуаций, можно задать с помощью следующего индуктивного определения:

1. Любая логическая переменная $w \in W$ (входное дискретное воздействие, переменная состояния, предикат от непрерывных состояний) или её отрицание является динамической ситуацией;
2. Любая элементарная конъюнкция динамических ситуаций также является динамической ситуацией.

С учетом этого и вводя обозначение S_j для некоторой динамической ситуации, будем иметь:

$$S_j = s_{j_1},...s_{j_i},...,s_{j_n}, \text{ где } s_{j_i} = w_{j_i} \text{ или } s_{j_i} = \neg w_{j_i},\ w_{j_i} \in W, n = 1...N_W, N_W = |W|$$

Общее количество локальных динамических ситуаций очень быстро растет при увеличении числа логических переменных. Однако количество ситуаций, определяющих развитие процессов, на практике оказывается достаточно ограниченным. Такие ситуации можно назвать значимыми для спецификации процесса ситуациями.

3. Правила трансформации динамических ситуаций

Обозначая множество значимых локальных ситуаций S, тип функции трансформации динамических ситуаций можно определить как $\sigma : S \to Q \times \{False, True\}$. Такая функция может быть специфицирована совокупностью правил типа "`if…then`", левая часть (условие) которых является локальной ситуацией, а правая (действие) — совокупностью (списком) переменных состояния. При возникновении данной ситуации (значение соответствующей ей конъюнкции равно *True*) переменным состояния из исполнительной части правила должны быть присвоены специфицированные значения (*True* или *False*). Реализация функции производится исполняющей процедурой, которая в цикле сканирует правила, вычисляет значения условных частей правил и в зависимости от этих значений выполняет присваивание специфицированных значений переменным из исполнительной части правил. Причем в ходе сканирования списка правил новые значения переменных запоминаются, а изменения значений соответствующих координат вектора состояния производятся одновременно в конце каждого шага алгоритма обновления состояния. Если изменение значений переменных производить непосредственно в ходе сканирования правил, то спецификации процессов окажутся зависимыми от расположения правил в списках, что усложнит программирование процессов.

Возможны три разновидности или типа правил трансформации ситуаций, различающиеся по выразительным возможностям спецификации процессов и по надежности получаемых спецификаций. Условная часть всех типов правил представляет собой динамическую ситуацию. В исполнительные части правил 1-го и 2-го типов могут включаться как сами логические переменные состояния, так и их отрицания. Правила этих типов имеют вид:

Труды СПИИРАН. 2007. Вып. 4. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2007. Issue 4. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

$$S_j \to q'_{j_1},...q'_{j_i},...,q'_{j_m}\ ;\ q'_{j_i} \in Q \times \{False, True\},\ m \in N\ .$$

Штрихи у переменных состояния означают, в соответствии с общепринятым соглашением, тот факт, что эти переменные принимают свои значения на следующем после возникновения ситуации шаге исполняющей процедуры.

Правила 3-го типа отличаются тем, что в их исполнительную часть помещаются только те переменные, которым необходимо присвоить значения *True*. При использовании этих правил исполняющая процедура сканирует правила, запоминает идентификаторы переменных, которым, в соответствии с ситуациями, должны быть присвоены значения *True* и в конце каждого шага алгоритма обновления состояния присваивает эти значения соответствующим координатам вектора состояния, а всем остальным координатам присваивается значение *False*. В этом случае исключается возможность появления противоречивых правил, однако несколько усложняются правила для присваивания переменным значения *False*.

Правила 1-го и 2-го типов различаются по действию исполняющей процедуры в случаях, когда условная часть правила имеет значение *False*. Исполняющая процедура вычисляет значение условной части правила и в случае ее истинности присваивает переменным из исполнительной части правила специфицированные значения. Если условная часть имеет значение *False*, то правило 1-го типа присваивает переменным из исполнительной части значения противоположные специфицированным, а правило 2-го типа в этом случае ничего не делает и значения переменных из исполнительной части не меняются. При использовании таких правил пользователь имеет возможность присваивать переменным состояния любое из значений истинности. Это расширяет выразительные возможности спецификации, но вместе с тем повышает требования к их логической корректности, так как допускает формирование противоречивых правил.

Спецификации процессов с помощью правил 1-го типа в сильной степени подвержены возникновению ошибок. Их использование для реализации взаимосвязанных процессов возможно только при наличии эффективных средств верификации баз таких правил. Разработка спецификаций процессов, основанных на использовании правил 2-го типа, является, как правило, задачей более трудоемкой, чем с использованием правил 3-го типа, так как необходимо определять оба возможных значения переменных.

Рассмотрим особенности использование правил 2-го и 3-го типов на примере спецификации простейшего гибридного процесса управления клапанами системы релейной стабилизации уровня. Пусть имеются: непрерывная переменная `Level` для представления уровня, две логические переменные для представления состояний впускного `Vin Open` и выпускного `Vout Open` клапанов и три предиката от уровня `(L > 200)`, `(L < 200)` и `(L < 20)`. Открытое состояние клапана соответствует значению переменной *True*. Пусть задачей управления является поддержание переменной `Level` в диапазоне значений между 20 и 200. Для этого при снижении уровня до нижнего значения необходимо открыть впускной клапан и закрыть выпускной. В этой ситуации происходит повышение уровня с постоянной скоростью. При превышении уровнем верхнего значения необходимо закрыть впускной клапан и открыть выпускной. После этого происходит понижение уровня со скоростью, пропорциональной значению уровня. На рис. 1 приведен график изменения уровня описанного процесса.

Труды СПИИРАН. 2007. Вып. 4. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2007. Issue 4. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

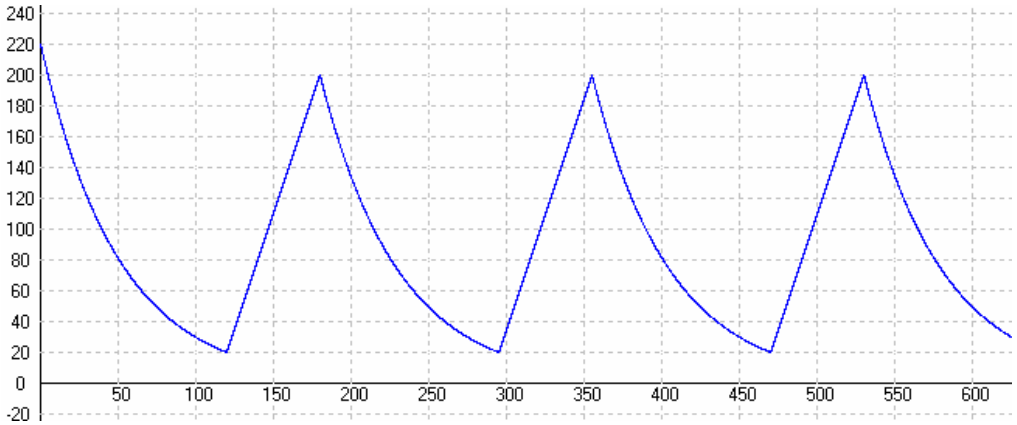

Рис. 1. Процесс релейной стабилизации уровня.

На рис. 2 приведена часть экрана редактора трансформационных правил среды EnviCon, на которой представлены правила трансформации ситуаций 2-го типа, специфицирующие процесс управления клапанами. На рисунке имена переменных, которым задано значение *True*, представлены на светлом фоне, а имеющие противоположное значение — на темном.

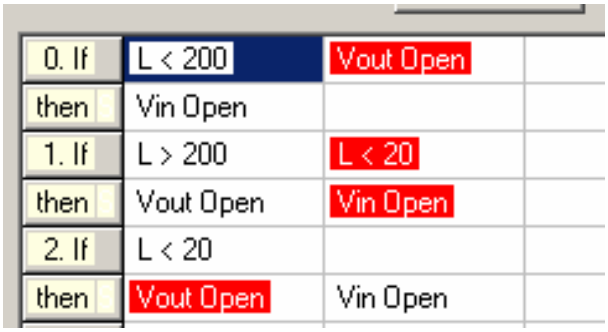

Рис. 2 Управление клапанами с помощью правил 2-го типа.

Пусть в исходном состоянии уровень больше 200 и оба клапана закрыты. В этой ситуации срабатывает только правило № 1, которое открывает выпускной клапан. Уровень начинает понижаться. Когда он станет меньше 200, то не будет срабатывать ни одно правило. Клапаны не изменят своего состояния, и уровень будет понижаться до значения 20. Когда уровень достигнет значения 20, сработает правило № 2, которое откроет впускной клапан и закроет выпускной. Правило № 0 нужно только для того, чтобы процесс начинался из исходной ситуации, когда оба клапана закрыты и уровень больше 20, но меньше 200.

Особенностью трансформационного правила 3-го типа является то, что в его исполнительную часть могут помещаться только переменные, которые при срабатывании правила принимают значения *True*. Исполняющая процедура, обрабатывающая такие правила, перед началом цикла сканирования правил всем логическим переменным вспомогательного вектора присваивает значения *False*. Затем в ходе сканирования срабатывающие правила меняют значения некоторых переменных на *True*. В конце цикла значения вспомогательных переменных присваиваются соответствующим переменным вектора состояния. Таким образом, если на данном шаге алгоритма обновления состояния нет ни одного правила, присваивающего некоторой переменной значение *True*, то она

будет иметь значение *False*. То есть отсутствие в данном случае трактуется как отрицание. Очевидно, что при этом невозможно появление конфликтующих правил, что безусловно повышает надежность спецификаций. Невозможность с помощью правил этого типа явного присваивания переменным значения *False* несколько снижает выразительные возможности баз таких правил и несколько усложняет спецификации процессов, основанные на их использовании.

На рис. 3 представлена часть экрана редактора трансформационных правил среды EnviCon, на которой приведена спецификация того же процесса (рис. 1) с помощью правил 3-го типа. Особенностью этих правил является невозможность использования в их исполнительных частях отрицаний переменных. Рассмотрим опять то же самое исходное состояние, то есть уровень больше 200 и оба клапана закрыты. В этой ситуации срабатывает только правило № 1, которое открывает выпускной клапан, после чего уровень начинает понижаться. При достижении уровнем значения 200 правило № 1 перестанет держать выпускной клапан в открытом положении. Чтобы он не закрылся, а был открыт до достижения уровнем значения 20, используется правило № 2, которое блокирует выпускной клапан в открытом положении. При достижении уровнем значения 20 не будет ни одного правила, открывающего выпускной клапан, поэтому он будет закрыт. После его закрытия начинает срабатывать правило № 0, которое открывает впускной клапан и поддерживает его открытым до тех пор, пока уровень ни станет больше 200. После этого условная часть правила № 0 примет значение *False*, что приведет к закрытию впускного клапана, а условная часть правила № 1 — *True*, что вызовет открытие выпускного клапана. Это в свою очередь вызовет срабатывание правила № 2, которое заблокирует выпускной клапан в открытом состоянии до тех пор, пока уровень ни достигнет значения 20. Процесс, таким образом, циклически повторяется.

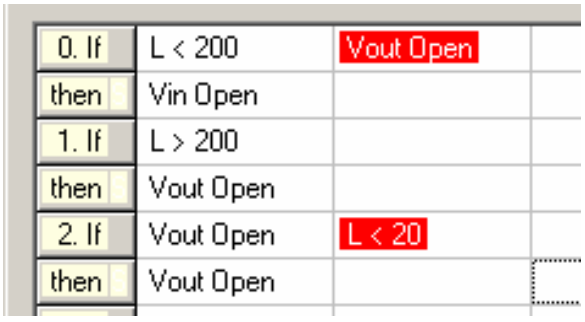

Рис. 3. Управление клапанами с помощью правил 3-го типа.

Сравнение приведенных спецификаций процесса (рис. 1), выполненных с помощью двух различных типов правил, позволяет сделать следующие выводы. Правила 2-го типа обладают большими выразительными возможностями. Спецификации на их основе легче интерпретируются пользователями, но получаются несколько длиннее, из-за того что необходимо специфицировать оба значения большинства переменных. В исполнительных частях этих правил допускается использование как самих переменных, так и их отрицаний. Поэтому в базе таких правил возможно формирование противоречивых правил, которые в одно и то же время будут специфицировать по отношению к одному и тому же состоянию процесса противоположные действия. Основным достоинством правил 3-го типа является то, что они не допускают возможности формирования таких противоречивых правил.

Труды СПИИРАН. 2007. Вып. 4. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2007. Issue 4. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

4. Использование событий для спецификации процессов

Динамическое событие происходит при изменении динамической ситуации. С каждой ситуацией могут быть связаны два события: одно с возникновением ситуации, другое — с ее исчезновением. Определение динамического события состоит в определении ситуации и указания на возникновение или исчезновение данной ситуации. На рис. 4 представлено отношение между ситуацией и связанными с ней событиями. Событие постоянно имеет значение *False* и только при изменении ситуации может принимать значение *True*. В теории событие имеет бесконечно малую длительность. В системах, основанных на правилах, длительность события равна длительности одного шага алгоритма обновления состояния.

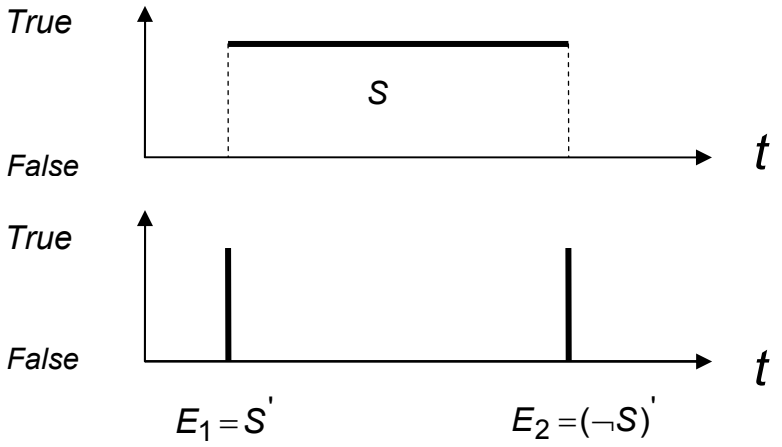

Рис. 4. Отношения между ситуацией и связанными с ней событиями.

Для вычисления события необходимо сравнивать текущее значение указанной ситуации с ее предыдущим значением. Если эти значения не совпадают, то формируется событие, связанное с возникновением или с исчезновением данной ситуации. Вычисление событий может выполняться исполняющей процедурой. Можно сформировать событие с помощью правил трансформации ситуаций. Например, правила 3-го типа, специфицирующие события E_1 и E_2, связанные с ситуацией S, будут иметь следующий вид:

$$S \to S_1;\ \neg S_1, S \to E_1;\ \neg S, S_1 \to E_2.$$

Введена вспомогательная ситуация S_1, значение которой определяется первым правилом и повторяет значение ситуации S с отставанием на один шаг. Переменная события E_1 определяется вторым правилом. Она имеет значение *True* в течение одного шага, когда S_1 еще ложна, а S уже истинна, то есть при возникновении ситуации S. Третье правило специфицирует событие E_2, связанное с исчезновением ситуации S.

Формирование событий необходимо при спецификации и реализации дискретных (событийных) изменений непрерывных состояний. Такие изменения могут происходить, например, при включении и выключении различных устройств в технических системах. При спецификации процесса падения мячика

Труды СПИИРАН. 2007. Вып. 4. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2007. Issue 4. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

событием является касание мячиком пола. С этим событием связано скачкообразное изменение знака и величины скорости. В этом качестве идентификаторы событий используются в правилах, определяющих транзитивные процедуры изменения непрерывных состояний. Эти правила здесь не рассматриваются.

События могут также использоваться для формирования правил трансформации ситуаций. В этом случае в условной части правила указывается событие, а в исполнительной части — идентификаторы и значения логических переменных состояния, которые они принимают при возникновении данного события. Эти значения переменных сохраняются неизменными до тех пор, пока они ни будут изменены другими правилами. Возможно совместное использование в условных частях правил ситуаций и событий. Поскольку исполняющая процедура обрабатывает условную часть правила как конъюнкцию указанных там переменных, то совместное использование ситуаций и событий специфицирует новое событие. Естественно, что одни и те же события в разных ситуациях могут вызывать различные действия.

На рис. 5 приведены правила, использующие события для спецификации управления клапанами и реализации процесса, приведенного на рис 1. Условные части правил содержат символ O, который указывает исполняющей процедуре, что правило срабатывает только при возникновении (Originate) указанной в условной части ситуации. В исполнительных частях правил содержится символ S, который сообщает исполняющей процедуре, что переменные в этой части правила следует рассматривать как ситуации, то есть они должны сохранять свои значения после изменения.

Рис. 5. Управление клапанами с помощью правил, использующих события.

Как видно из рис. 5, спецификация управления в данном случае содержит меньше правил и является достаточно наглядной. Однако спецификация с помощью событий больших совокупностей взаимодействующих процессов является сложной и трудоемкой для пользователя задачей. Это объясняется тем, что в каждый текущий момент состояния процессов определяются не текущей ситуацией, как например при использовании трансформационных правил 3-го типа, а совокупностью предыдущих событий. Для эффективного использования событий в правилах спецификации процессов необходимо разработать процедуры верификации и валидации баз таких правил.

5. Заключение

Сравнительный анализ рассмотренных правил и приведенных примеров спецификаций процесса, основанных на их использовании, позволяют сделать вывод о том, что наиболее пригодными для разработки исполняемых спецификаций процессов в настоящее время являются правила трансформации ситуа-

Труды СПИИРАН. 2007. Вып. 4. ISSN 2078-9181 (печ.), ISSN 2078-9599 (онлайн)
SPIIRAS Proceedings. 2007. Issue 4. ISSN 2078-9181 (print), ISSN 2078-9599 (online)
www.proceedings.spiiras.nw.ru

ций 3-го типа. Это объясняется тем, что структура этих правил делает невозможным формирование противоречивых правил. При этом они обладают достаточными выразительными возможностями. Для эффективного использования правил других типов необходимо разработать процедуры, обеспечивающие выявление противоречивых правил.

Широкое использование трансформационных правил для реализации дискретно-событийных и гибридных процессов, на наш взгляд, сдерживается недостаточным развитием средств верификации и валидации баз таких правил. Помимо процедур, обеспечивающих поиск противоречивых правил, необходимо также разработать процедуры, позволяющие находить дублирующие и перекрывающие друг друга правила, а также замкнутые цепочки правил, приводящих к возникновению нежелательных циклических изменений ситуаций.

Литература

1. *Alur R., Henzinger T. A., Lafferriere G., Pappas G. J.* Discrete Abstractions of Hybrid Systems // Proceedings of the IEEE. 2000. No. 88. P. 971–984.
2. *Шпаков В. М.* Ситуационные спецификации имитационных моделей гибридных реактивных систем // Труды СПИИРАН. Вып. 1, т. 2. СПб.: СПИИРАН, 2002. 212–222 с.
3. *Шпаков В. М.* Спецификация знаний динамики на основе транзитивной модели непрерывных процессов // Труды СПИИРАН. Вып. 3, т. 1. СПб.: Наука, 2006. 191–197 с.