ТРАНЗИТИВНАЯ МОДЕЛЬ ПРОЦЕССОВ И ЕЁ ИСПОЛЬЗОВАНИЕ ДЛЯ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ДИНАМИЧЕСКИХ СИСТЕМ

В. М. Шпаков (Санкт-Петербург)

Функционирование динамической системы определяется совокупностью протекающих в ней процессов. Различают процессы непрерывные, дискретно-событийные и гибридные. Все процессы могут быть представлены с помощью переменных состояния. В случае непрерывного процесса изменения носят количественный характер и в качестве переменных состояния используются вещественные переменные, которые за бесконечно малый промежуток времени изменяются на бесконечно малые величины. В дискретно-событийных процессах изменения происходят мгновенно в дискретные моменты времени и носят 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. Событийные изменения в гибридных процессах называются режимами. При изменении режима может происходить изменение динамики непрерывной составляющей процесса, или скачкообразное изменение величин некоторых непрерывных состояний, или и то и другое. С другой стороны, выход некоторого непрерывного состояния за границу заданного диапазона изменений может представлять собой событие и вызывать изменение режима данного гибридного процесса.

В последнее время для решения задач анализа дискретно-событийных и гибридных динамических систем было предложено использовать транзитивные модели протекающих в системах процессов [1]. Эти модели определяют транзитивное замыкание отношения следования между состояниями процесса. Они для любого возможного состояния процесса однозначно определяют переход (transition) его в следующее состояние и, тем самым, позволяют специфицировать процесс непосредственно в виде последовательности или траектории состояний.

Математической моделью дискретно-событийного процесса является конечный автомат, или машина конечных состояний (Finite State Machine (FSM)), базовый формализм которой представляет собой следующий кортеж:

$$FSM = (Q, \Sigma, \sigma, q_0),$$

где Q – множество символов, представляющих дискретные состояния;

Σ – множество входных символов;

$\sigma: Q \times \Sigma \to Q$ – функция перехода;

$q_0 \in Q$ – начальное состояние.

В качестве вычислительной модели FSM может быть использована совокупность продукционных правил вида *Условие* $\to$ *Действие*. Условием являются значения текущего состояния и входного символа, а действием – переход в заданное состояние. При совокупности взаимодействующих процессов функция перехода одного процесса может зависеть от состояния любого другого процесса и от значения любого входного символа. В этом случае входной алфавит и алфавит состояний FSM будут представлять собой прямые произведения соответствующих множеств задания отдельных переменных. Элементами этих произведений будут соответственно входные слова и слова-состояния. В [2] показано, что любое символьное состояние можно представить совокупностью логических состояний, что в ряде случаев улучшает выразительные возможности спецификаций процессов с помощью правил и позволяет существенно повысить эффективность исполнительной процедуры, сканирующей базу правил. В этом

случае условие правила представляет собой элементарную конъюнкцию логических переменных состояния и входных команд.

Под состоянием гибридного процесса понимают пару (v, x), которая состоит из значения дискретной переменной режима $v \in V$ и точки $x \in R$, являющейся состоянием непрерывной составляющей процесса. В случае совокупности взаимодействующих процессов различной динамики режим одного гибридного процесса может также зависеть от входных команд Σ и состояний дискретно-событийных процессов $q \in Q$, а также от режимов других гибридных процессов $v \in V$. Кроме того, режим гибридного процесса может зависеть от предикатов $g \in G$ непрерывных состояний любого непрерывного процесса или непрерывной составляющей любого гибридного процесса $x \in X$, включая входные процессы $X_i \subseteq X$. Предикаты от непрерывных состояний могут иметь различный вид. Хорошей выразительной возможностью обладают предикаты диапазонов изменения непрерывных состояний, использованные в [2]:

$$((x_{j_1} \geq a_k + x_{j_2}) \wedge (x_{j_3} \leq b_k + x_{j_4})) \rightarrow g'_k, \qquad (1)$$

где $x_{j_1}, x_{j_2}, x_{j_3}, x_{j_4} \in X, g'_k \in G$, a_k и b_k – константы, соответствующие некоторому диапазону. В общем случае состояния дискретно-событийных процессов могут также зависеть от режимов гибридных процессов и от предикатов непрерывных процессов. Таким образом, можно говорить о глобальной динамической ситуации, определяемой всеми логическими переменными, а также о локальных ситуациях, вызывающих определённые изменения дискретных состояний и режимов. С учётом введённых обозначений произвольная локальная ситуация может быть представлена в следующем виде:

$S_j = s_{j_1}, ... s_{j_i}, ..., s_{j_n}$, где $s_{j_i} = w_{j_i}$ или $s_{j_i} = \neg w_{j_i}$,

где $w_{j_i} \in \Sigma \cup Q \cup V \cup G$, $i \leq n$, $n = 1...N$, $N = |\Sigma \cup Q \cup V \cup G|$.

При этом продукционное правило, определяющее изменения дискретных состояний и режимов, а следовательно, трансформацию ситуаций, будет иметь вид

$$S_j \rightarrow w'_{j_1}, ..., w'_{j_i}, ..., w'_{j_m}, \text{ где } w'_{j_i} \in Q \cup V. \qquad (2)$$

Штрихи в обозначении переменных состояния означают непосредственное следование во времени, т. е. сначала ситуация в левой условной части правила принимает значение True, а затем эти же значения принимают переменные правой части правила. Механизм продвижения времени может быть различным, он определяется исполняющей процедурой.

Каждый режим гибридного процесса обладает своей динамикой изменения непрерывных переменных, определяемой зависящей от режима функцией $f : V \times R \rightarrow R$. Транзитивный подход может быть использован и для моделирования непрерывных составляющих гибридных процессов [1]. При этом непрерывный процесс представляется последовательностью дискретных значений вектора непрерывных состояний. Естественно, погрешность представления непрерывного процесса дискретным по времени процессом будет определяться величиной дискрета по времени Δt, т.е. длительностью одного шага цикла обновления непрерывного состояния. Высокое быстродействие современных компьютеров позволяет обеспечить приемлемую точность при реализации данного подхода для моделирования многих реальных процессов. С точки зрения эффективности реализации исполняющей процедуры трансформацию непрерывных состояний целесообразно специфицировать с помощью правил трансформации отдельно

для каждого режима работы. Очевидно, при изменении режима будет происходить изменение динамики, которая будет определяться другими правилами. Трансформационные правила должны определять последующие значения непрерывных состояний на основании текущих состояний и времени. Это можно сделать двумя способами: задавая текущую (мгновенную) скорость изменения координаты вектора непрерывного состояния на следующем шаге или непосредственно задавая значение этой координаты. Правило спецификации скоростей может иметь следующий вид:

$$S_j \to (\dot{x}_k = (c_{jk} + a_{jk} x_m)),\ j,k,m \in N, \qquad (3)$$

где c_{jk}, a_{jk} – коэффициенты, соответствующие изменению k-й переменной при данном режиме, определяемом ситуацией S_j; $\dot{x}_k$ – скорость изменения (производная) специфицируемой данным правилом переменной на следующем шаге. Совокупность правил такого типа эквивалентна системе линейных разностных уравнений первого порядка и позволяет описывать линейные динамические структуры произвольного вида. Для реализации этого правила исполняющая процедура должна на каждом шаге вычислять координаты через скорость и приращение времени.

При втором способе на каждом шаге вычисляется текущее значение ситуации, задаваемое левой частью правила, и в случае его истинности вычисляется значение специфицируемой непрерывной переменной из правой части в соответствии с заданной функциональной зависимостью и текущими значениями аргументов. Вычисленные значения присваиваются переменной на следующем шаге. В этом случае правило будет иметь вид

$$S_j \to (x'_k = f_{jk}(x_m, t)),\ j,k,m \in N, \qquad (4)$$

где x'_k – значение x_k на следующем шаге; f_{jk} – функция, задающая изменение k-й непрерывной переменной в ситуации S_j; t – время. С помощью правил этого типа могут быть специфицированы интегро-дифференциальные и произвольные нелинейные функциональные зависимости. В правилах (3) и (4) символ = обозначает операцию присваивания, а не операцию отношения.

Естественно, что в одной и той же ситуации изменение одной непрерывной переменной может задаваться лишь одним из правил (3), (4). Для удобства реализации целесообразно разбить множество непрерывных состояний на два подмножества по типу используемого для спецификации их элементов правила [X_v для правил (3) и X_f для правил (4)]. В результате для множества непрерывных переменных имеем $X = X_i \cup X_v \cup X_f$.

В СПИИРАН разработан прототип среды моделирования динамических систем, основанный на реализации описанной транзитивной модели процессов. Программирование совокупности процессов в среде осуществляется путём формирования правил (1)–(4). Среда представляет собой набор взаимодействующих инструментальных средств. Она позволяет создавать иерархически структурированные совокупности моделей и осуществлять связи между ними. Состояние каждой модели задаётся двумя векторами. Координатами одного из них являются логические переменные, которые используются для представления входных воздействий, состояний дискретно-событийных процессов и режимов гибридных процессов. Координатами второго векто-

ра являются вещественные числа, используемые для представления непрерывных входных процессов и непрерывных переменных состояния гибридных и непрерывных процессов. Редакторы векторов обеспечивают редактирование имен координат и задание координатам начальных значений. Значения логических координат отображаются цветом фона: красный для значения *False*, зелёный для *True*. Значение логической координаты изменяется путем двойного щелчка мышью на имени соответствующей координаты.

Среда содержит четыре редактора правил, каждый из которых ориентирован на структуру векторов и структуру одного из правил (1)–(4). Они позволяют формировать правила путём множественного выбора необходимых координат из векторов данной модели и векторов состояния модели более высокого уровня. На рис. 1 приведена часть экранной формы редактора правил трансформации ситуаций (продукционных правил). Условная часть правила обозначена идентификатором “If”,

Рис. 1. Редактор правил трансформации ситуаций

а исполнительная часть – идентификатором “then”. Обе части правила содержат имена логических переменных, значения которых представлены цветом. Если текущие значения переменных таковы, что конъюнкция переменных условной части правила имеет значение *True*, то переменным исполнительной части правила присваиваются значения, специфицированные в этой части правила. На рис. 2 приведена часть экранной формы редактора функциональных правил. Исполняющая процедура при обработке этих правил вначале проверяет значение переменной из графы “Ситуация”. Если это значение соответствует указанному, то по текущим значениям аргументов вычисляется значение функции и это значение присваивается переменной из графы “Функция”.

Рис. 2. Редактор правил трансформации функций

Среда также содержит редакторы выходных логического и вещественного векторов модели, и редактор связей данной модели с моделями одного с ней уровня и с моделью более высокого уровня.

Механизм продвижения времени обеспечивает режимы модельного и реального масштаба времени. В первом случае пользователь непосредственно задает длительность шага цикла обновления состояния. В режиме реального времени длительность шага пропорциональна длительности исполняющей процедуры, коэффициент пропорциональности задается пользователем. Предусмотрено три способа приостановки процесса моделирования: по нажатию клавиши, при возникновении помеченной ситуации или при каждом изменении состояния.

Предусмотрены три способа визуализации процесса имитации: первый – в виде таблиц, содержащих имена и значения координат векторов состояния; второй – в виде графиков изменения выбранных непрерывных переменных, третий – анимационная картинка. В последнем случае сама картинка и связь ее элементов с координатами векторов должны разрабатываться отдельно для каждой конкретной задачи.

Эффективность и выразительность реализованного подхода экспериментально исследовались путём разработки имитационных моделей ряда тестовых систем, а также создания модели автоматизированной промышленной установки для ожижения гелия КГУ-150. Экспериментальные исследования показали, что описанная транзитивная модель процессов допускает эффективную компьютерную реализацию и обладает выразительностью, достаточной для спецификации многих видов процессов и динамических систем.

Как вектора состояний процессов и базы правил, так и исполняющие процедуры обработки правил могут быть эффективно реализованы с помощью универсальных алгоритмических языков программирования. Полученные программные модули могут служить основой создания компьютерных сред или компонентов моделирования процессов, встраиваемых в различные программные продукты. Полученные таким путём приложения предоставляют пользователю возможность самостоятельно программировать модели процессов на языке, близком к предметной области, что в ряде случаев позволяет расширить области применения программных средств и длительности их жизненных циклов.

Литература

1. **Alur R., Henzinger T. A., Lafferriere G. and Pappas G. J.** Discrete Abstractions of Hybrid Systems//Proceedings of the IEEE 88, 2000. – P. 971–984.
2. **Шпаков В. М.** Исполняемые спецификации транзитивных моделей технологических процессов//Мехатроника, автоматизация, управление. – 2004. – № 3.