

СПЕЦИФИКАЦИЯ ДИСКРЕТНОГО ПОВЕДЕНИЯ ГИБРИДНЫХ СИСТЕМ

И. Н. Томилов, Д. Н. Достовалов, М. С. Денисов (Новосибирск)

Матрица переходов

Удобной и наглядной формой описания поведения гибридных систем (ГС), использующейся в большинстве современных систем моделирования, является ориентированный граф. Вершины графа соответствуют непрерывным режимам, а дуги – условиям смены режима и выполняемым при этом мгновенным действиям. Эта форма является аналогом диаграммы состояний, принятой в «унифицированном языке моделирования» UML [2].

В системе моделирования ИСМА [3] разработан и реализован альтернативный вариант спецификации с использованием символьного представления модели на предметно-ориентированном языке LISMA [4]. На этапе семантического анализа на основе текстового представления модели системой генерируется *матрица переходов*, однозначно управляющая дискретным поведением и реализующая все возможности языка графической спецификации ГС в виде карты состояний.

Базовым языковым объектом, на котором основывается спецификация дискретного поведения ГС на языке LISMA, является *условно-адресная пара* (УАП). УАП состоит из адресной составляющей – названия режима *st*, и условной составляющей – логического предиката *pr* (рис. 1). УАП определяет непрерывный режим ГС и условия, при выполнении которых происходит переход в этот режим.

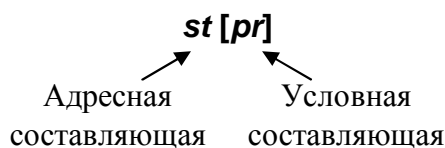


Рис. 1. Условно-адресная пара

По сути, задание дискретного перехода является заданием дуги орграфа, описывающего карту состояний ГС. Для задания дуги необходимо, во-первых, указать начальную и конечную вершины дуги, во-вторых, указать условие, при выполнении которого осуществляется переход.

Язык LISMA дает возможность формировать комбинированное условие, позволяющее определять следующий режим системы не только на основе предиката перехода в него, но и в совокупности с предикатом выхода из предыдущего режима. Переход происходит только тогда, когда оба предиката принимают истинное значение. Этот механизм позволяет учитывать предысторию, т.е. выбор следующего режима также зависит от того, в каком режиме система находилась в предыдущий момент времени.

Дуга карты состояний в ИСМА генерируется на основе двух УАП. Их адресные составляющие определяют начальную и конечную вершины, конъюнкция условных составляющих – предикат перехода.

Формат записи УАП входа в состояние и выхода из состояния отличаются. УАП выхода из состояния может не содержать условной составляющей. В этом случае действует система умолчаний, и семантический анализатор подставляет условную составляющую с предикатом *pr: TRUE*.

Пусть $V = \{v_1, \dots, v_n\}$ – непустое множество локальных непрерывных режимов ГС, $A \subseteq V^2$ – множество дискретных переходов системы (множество дуг карты состояний), где $V^2 = V \times V = \{(v_i, v_j), i, j = \overline{1, n}\}$ – множество всех упорядоченных пар элементов

множества V . Тогда пару $\langle V, A \rangle$ будем называть ориентированным графом $G(V, A)$, в котором (v_i, v_j) – дуга (v_i – начальная, v_j – конечная вершины).

Матрица смежности графа $G(V, A)$ определяется как квадратная матрица $M = [m_{ij}]$ порядка n , в которой

$$m_{ij} = \begin{cases} 1, & \text{если } (v_i, v_j) \in A, \\ 0, & \text{если } (v_i, v_j) \notin A. \end{cases} \quad (1)$$

Таким образом, матрица смежности содержит информацию о возможных дискретных переходах без учета логических предикатов.

Целесообразным является в качестве элемента матрицы использовать условие, при выполнении которого осуществляется переход.

Пусть pr_i и pr_j – условные составляющие УАП выхода из текущего и входа в следующее состояние соответственно. Тогда определим матрицу переходов как квадратную матрицу $M' = [m'_{ij}]$ порядка n , в которой

$$m'_{ij} = \begin{cases} (pr_i \wedge pr_j), & \text{если } (v_i, v_j) \in A, \\ false, & \text{если } (v_i, v_j) \notin A. \end{cases} \quad (2)$$

Матрица переходов является полным функциональным аналогом карты состояний, а особенности языка спецификации ГС LISMA дают дополнительные возможности управления дискретным поведением.

Для иллюстрации однозначности соответствия между картой поведения и матрицей переходов приведем описание дискретного поведения модели двух резервуаров [5] в виде карты состояний (рис. 2) и матрицы переходов (таблица).

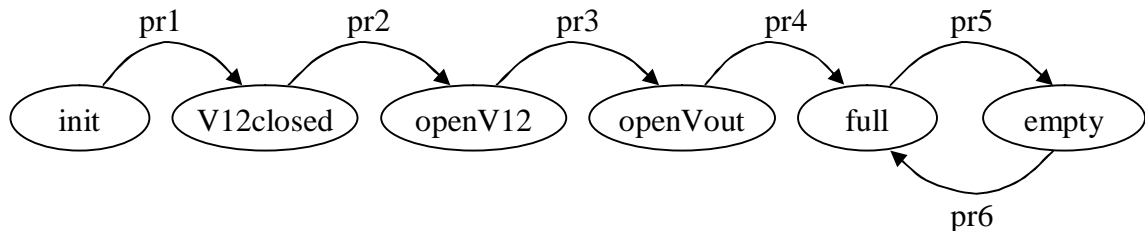


Рис. 2. Карта состояний модели двух резервуаров

Матрица переходов

	init	V12closed	openV12	openVout	full	empty
init	false	false	false	false	false	false
V12closed	pr1	false	false	false	false	false
openV12	false	pr2	false	false	false	false
openVout	false	false	pr3	false	false	false
full	false	false	false	pr4	false	pr6
empty	false	false	false	false	pr5	false

Представленное в таблице состояние *init* является стартовым системным состоянием. Оно задается неявным образом в виде операторов, не имеющих отношения к описанным пользователем состояниям. В рамках этого состояния выполняются следующие действия:

- задание глобальных начальных условий (инициализация фазовых переменных);
- выполнение динамики, соответствующей начальному поведению системы.

Состояние *init* генерируется семантическим анализатором и имеет следующий формат:

```
init [TRUE] is
<операторы_задания_глобальных_начальных_условий>
<операторы_задания_начального_поведения_системы>
from NULL;
```

Ключевое слово *NULL* в секции *from* означает, что переход в стартовое состояние из других состояний невозможен.

Событийное управление

В рамках принятой терминологии под *событием* понимается некоторое действие (последовательность действий) нулевой длительности, осуществляемое при выполнении определенного условия, и не приводящее к смене текущего непрерывного состояния. Т.е. событие можно считать состоянием нулевой длительности. Графическая интерпретация события изображена на рис. 3.

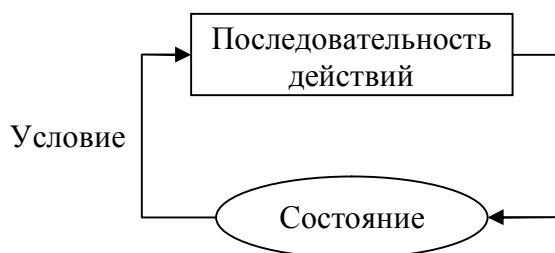


Рис. 3. Графическая интерпретация события

При помощи механизма событийного управления реализуются внешние воздействия на моделируемую систему, т.е. действия, имеющие другую природу происхождения и не зависящие от законов функционирования системы. Но событие может косвенно привести к смене локального состояния, так как оно может изменить значения фазовых переменных. Использование событийного управления особенно оправдано в случае изменения начальных условий при возникновении события.

На языке LISMA событийное управление реализуется стандартным условным оператором *if-then-endif*, имеющим следующий формат:

```
if <условие> then <последовательность_действий> endif;
```

где <условие> – логический предикат; <последовательность_действий> – непустая последовательность операторов присваивания.

Общим вариантом условного оператора, используемого в языках программирования высокого уровня, является конструкция *if-then-else*, описываемая неоднозначной грамматикой. Но использование этого варианта осложняется необходимостью перехода к однозначной грамматике [1, 6], в то время как вариант *if-then-endif* является однозначным.

Эффективность использования событийного управления при спецификации моделей ГС можно показать на примере модели системы охлаждения ДВС [7] (рис. 4).

<pre> // малый контур little [(t<75)] is t'=k1*t; from init; // большой контур с выключенным // вентилятором big_off [(t>=75) and (t<85) and (on>0)] is on=0; t'=(k2-k3)*t; from little, big_on; // большой контур с включенным // вентилятором big_on [(t>=110) and (on<1)] is on=1; t'=(k2-k3-k4)*t; from big_off; </pre>	<pre> // события управления вентилятором if (t>=110) then on~=1; endif; if (t<85) then on~=0; endif; // малый контур little [t<75] is t'=k1*t; from init; // большой контур big [t>=75] is t'=(k2-k3-k4*on)*t; from little; </pre>
---	--

Рис. 4. Модель системы охлаждения ДВС без событийного управления (слева) и с использованием событийного управления (справа)

Использование событийного управления при построении модели ГС позволяет:

- упростить логику дискретного поведения модели;
- сократить число непрерывных режимов модели;
- описывать дискретные процессы в форме, отражающей их физический смысл;
- реализовывать неоднозначные характеристики (типа гистерезиса), не прибегая к использованию вспомогательных переменных.

Заключение

Формализмы символьной и структурно-символьной спецификации, реализованные в рамках пакета моделирования ИСМА, дополняют существующие формализмы спецификации ГС и имеют при этом свои функциональные преимущества.

Карты поведения, являясь простым и наглядным способом представления системы дискретных переходов, для многих задач могут оказаться избыточной спецификацией, особенно в случае классических динамических систем.

Символьная модель во всех случаях сопровождается декларативностью, что для задач повышенной размерности может оказаться неприемлемым. Поэтому при разработке языка спецификации LISMA декларативные конструкции были исключены.

На примере технической системы описаны механизмы символьной спецификации дискретной составляющей моделей ГС в инструментальной среде. Имитационные

эксперименты с рядом моделей показывают достаточную адекватность полученных результатов, что подтверждает состоятельность изложенных формализмов спецификации.

Литература

1. **Ахо А. В., Равви С., Ульман Дж.** Компиляторы: принципы, технологии, инструменты: Пер. с англ. М.: Вильямс, 2001. 768 с.
2. **Бенькович Е. С., Колесов Ю. Б., Сениченков Ю. Б.** Практическое моделирование динамических систем. СПб.: БХВ-Петербург, 2002. 464 с.
3. **Шорников Ю. В. и др.** Инструментальные средства машинного анализа. Свидетельство об официальной регистрации программы для ЭВМ № 2005610126. М.: Роспатент, 2005.
4. **Шорников Ю. В., Томилов И. Н.** Программа языкового процессора с языка LISMA (Language of ISMA). Свидетельство об официальной регистрации программы для ЭВМ № 2007611024. М.: Роспатент, 2007.
5. **Шорников Ю. В. и др.** Спецификация и численный анализ гибридных систем в ИСМА // Труды науч.-техн. конф. “Научное программное обеспечение в образовании и научных исследованиях”. СПб.: Изд-во Политехн. ун-та, 2008. С. 138–144.
6. **Шорников Ю. В.** Теория и практика языковых процессоров: Учебное пособие. Новосибирск: Изд-во НГТУ, 2004. 203 с.
7. **Хачиян А. С.** Двигатели внутреннего сгорания. М.: Высш. шк., 1985.