

НЕКОТОРЫЕ АСПЕКТЫ ПОСТРОЕНИЯ ИМИТАЦИОННЫХ МОДЕЛЕЙ СЛОЖНЫХ СИСТЕМ

А.Ф. Лановой

(Харьковский национальный университет радиоэлектроники)

Рассматривается проблема разработки и построения имитационной модели сложной системы с использованием объектно-ориентированного подхода. На начальном этапе исследования был выполнен анализ структуры модели на основе механизма управления элементами модели.

имитационная модель, сложная система, объектно-ориентированный подход

Введение и постановка задачи. Объектно-ориентированное моделирование предоставляет огромное количество инструментальных средств для построения и исследования моделей сложных динамических систем. В большинстве случаев принято считать, что динамическая система относится к сложным, если она включает в себя множество компонент, которые обладают различными динамическими характеристиками. Однако во многих случаях при моделировании сложных систем нет необходимости в использовании сложных пакетов прикладных программ. Использование объектно-ориентированного подхода при создании имитационных моделей таких систем позволяет создавать многоуровневые системы, состоящие из набора элементарных моделей, работающих одновременно и во многих случаях независимо друг от друга. В этом случае основной задачей становится решение проблемы одновременного запуска всего связанного набора элементарных моделей и координация их функционирования [1, 2].

Формализованное описание универсальной модели сложной динамической системы. Формализованное описание имитационной модели M сложной системы дискретного типа в понятиях DEVS (Discrete Event Specification)[11] будет иметь следующий вид:

$$M = \langle X, S, Y, \delta_{\text{int}}, \delta_{\text{ext}}, \lambda, \tau \rangle, \quad (1)$$

где X – входное множество управляющих воздействий; S – множество состояний системы; Y – множество результирующих воздействий системы; $\delta_{\text{int}}: S \rightarrow S$ – внутренняя функция изменения состояния системы;

$\delta_{ext}: Q \times S \rightarrow S$ – внешняя функция изменения состояния системы; Q – это «общее состояние», включающее в себя весь ряд предыдущих и текущих состояний модели; $\lambda: Q \rightarrow Y$ – выходная функция; τ – временная функция преобразования. Этот метод формализованного описания имитационной модели широко освещен во многих публикациях, посвященных вопросам имитационного моделирования [1 – 7].

Вместе следует заметить, что ни один из элементов модели (1) не обладает ярко выраженными дискретными характеристиками. Поэтому вполне правомочной будет попытка использования аппарата DEVS при описании динамической системы.

Одним из наиболее важных применений DEVS-формализации при описании модели является точное и полное описание с ее помощью составных элементов модели – объектов. Использование этого метода позволяет нам объединять в единое целое сложные модели, основанные на модульном принципе. Задача в этом случае заключается в том, чтобы однозначно установить внутренние связи между модулями модели таким образом, чтобы внешнее входное воздействие вызывало соответствующую реакцию этих элементов. Внешние выходные воздействия также должны порождаться на основе внутренних связей, которые задают внутреннюю структуру модели. Тогда формализованное описание модели сложной системы CM с учетом вышеизложенного будет иметь следующий вид:

$$CM = \langle D, \{M_i\}, \{I_i\}, \{Z_{ij}\}, SELECT \rangle, \quad (2)$$

где D – имя модуля; M_i – множество моделей, составляющих модуль, в описаниях DEVS; I_i – множество элементов, влияющих на поведение модели; Z_i – функция перехода модели из состояния i в состояние j . Функция SELECT обрабатывает одновременно происходящие внутренние события при смене состояния модели. В некоторых версиях DEVS (Extended DEVS) функция SELECT заменена функцией ORDER, которая преобразует в последовательную форму выполнение одновременно полученных внешних воздействий. Использование расширенной спецификации DEVS приобретает особое значение при использовании распараллеливания вычислений, но этот вопрос заслуживает отдельного обсуждения.

Иерархия модели и ее оптимизация. Пусть $N = \{d_1, \dots, d_n\}$ – все множество элементов *библиотеки модулей* (БМ), из которых могут быть непосредственно сформированы модели на этапе моделирования. Через K обозначим конечное множество *менеджеров*, управляющих взаимодействием элементов библиотеки в модели. Менеджеры необходимы

того, чтобы элементы библиотеки взаимодействовали друг с другом в соответствии с некоторой технологией. Менеджеры будут обозначаться через $k, k', k'', k_1, k_2, \dots \in K$.

У каждого менеджера модели имеются некоторые правила, в соответствии с которыми он управляет поведением своих элементов БМ и взаимодействием своей модели с другими менеджерами. Определим подчиненность формально. Пусть $CM = N \cup K$ – все множество элементов модели сложной системы. Тогда определим множество ребер подчиненности как $E \subseteq CM \times K$. Ребро подчиненности $(d, k) \in E$ означает, что элемент БМ $d \in CM$ непосредственно подчинен менеджеру $k \in K$, т.е. ребро направлено от элемента БМ к его менеджеру, осуществляющему непосредственное управление его функционированием. Тогда модель $M_i \in CM$ управляется менеджером $k_i \in K$, если существует цепочка ребер управляющих воздействий из d в k .

Будем также говорить, что менеджер управляет элементами БМ, или элемент БМ управляется менеджером.

Дадим строгое определение иерархии.

Определение 1. Ориентированный граф $H = (N \cup K, E)$ с множеством ребер управления $E \subseteq (N \cup K) \times K$ назовем иерархией, управляющей множеством элементов БМ N , если H ациклический, любой менеджер имеет подчиненные элементы БМ и найдется менеджер, которому подчинены все менеджеры. Через $\Omega(N)$ обозначим множество всевозможных иерархий.

Ациклическость означает, что образованный граф управления не содержит замкнутые контуры, реализующие неконтролируемые обратные связи в модели. Определение также исключает ситуации, в которых имеются менеджеры без подчиненных элементов БМ, так как это противоречит роли менеджера. Определение требует, чтобы в иерархии был менеджер, который управляет (непосредственно или с помощью других менеджеров) всеми элементами БМ. Такой менеджер способен разрешать конфликтные ситуации и исключить конкурентную борьбу за ресурсы между любыми элементами БМ сложной модели.

Моделью или группой элементов БМ модели $s \in N$ назовем любое непустое подмножество множества элементов БМ, выполняющее определенные действия по моделированию объекта. По определению 1 каждый менеджер имеет по крайней мере один подчиненный элемент БМ. Начав с любого менеджера k , мы можем двигаться «сверху вниз» к подчиненным элементам БМ менеджера k . В силу ациклическости в конечном итоге приходим к подчиненной группе БМ. То есть каждому менеджеру k в

любой иерархии H подчинена некоторая группа модулей модели $s_H(k) \subseteq N$. Будем также говорить, что менеджер k управляет группой модулей из элементов БМ $s_H(k)$.

Можно предположить, что простейшая модель $d \in N$ состоит из двух зависимых компонент: простейшая группа $s_H(d) = \{d\}$, состоящая из одного элемента БМ. Также будем говорить, что менеджер k модели $d \in N$ управляет простейшей группой $s_H(d) = \{d\}$.

Очевидно, что для любой иерархии H и любого менеджера $k \in K$ выполняется условие:

$$s_H(k) = s_H(d_1) \cup s_H(d_2) \cup \dots \cup s_H(d_n),$$

где d_1, \dots, d_n – все непосредственно подчиненные менеджеру k элементы модели. Для любой подчиненной модели d менеджера k выполняется $s_H(d) \subseteq s_H(k)$.

Определение 2. Функцию затрат менеджера $k \in K$ в иерархии $H = (N \cup K, E) \in \Omega(N)$ назовем секционной, если она имеет вид:

$$c(s_H(d_1), \dots, s_H(d_n)), \quad (3)$$

где d_1, \dots, d_n – все непосредственно подчиненные менеджеру k элементарные модели; $s_H(d_1), \dots, s_H(d_n)$ – группы модулей, формирующие модель системы; $c(\cdot)$ – функция, ставящая в соответствие любому набору групп неотрицательное действительное число. Затраты иерархии равны сумме затрат на управление моделью с использованием всех ее менеджеров:

$$c(H) = \sum c(s_H(d_1), \dots, s_H(d_n)). \quad (4)$$

Функцию затрат иерархии (4) также назовем секционной. Иерархию назовем оптимальной, если она минимизирует затраты на множестве $\Omega(N)$.

Секционная функция затрат иерархии складывается из затрат всех менеджеров, причем затраты каждого менеджера определяются только группами модулей, которыми управляют непосредственные подчиненные менеджера.

Поясним определение секционной функции на примере фрагмента иерархии, изображенного на рис. 1. Менеджер модели k управляет группой модулей $\{d_1, d_2, d_3, d_4\}$. При этом k осуществляет управление с помощью двух непосредственно подчиненных менеджеров k_1 и k_2 . Подчиненный менеджер k_1 управляет группой модулей $\{d_1, d_2\}$, формирующих модель m_1 , а подчиненный менеджер k_2 управляет группой модулей $\{d_3, d_4\}$, формирующих модель m_2 . Предполагается, что модели m_1 и m_2 полностью адекватны моделируемым элементам объекта. В этом случае затраты менеджера k на управление всей моделью в целом не зависят от того, как именно k_1 и k_2 управляют подчиненными модулями модели.

Согласно определению 2 затраты менеджера зависят только от того, каким образом подчиненная группа модулей модели распределена между своими менеджерами. В рассмотренном примере группа $\{d_1, d_2, d_3, d_4\}$ разделена на две непересекающиеся подгруппы:

$$\{d_1, d_2, d_3, d_4\} = \{d_1, d_2\} \cup \{d_3, d_4\},$$

поэтому затраты менеджера k составят $c(\{d_1, d_2\}, \{d_3, d_4\})$.

То есть предполагается, что затраты менеджера зависят только от той *секции* («модуля», «звена» и т.п.), которой он управляет непосредственно. На рис. 1 такая секция состоит из самого менеджера k и его непосредственных подчиненных менеджеров k_1 и k_2 . От остальной части иерархии затраты менеджера не зависят.

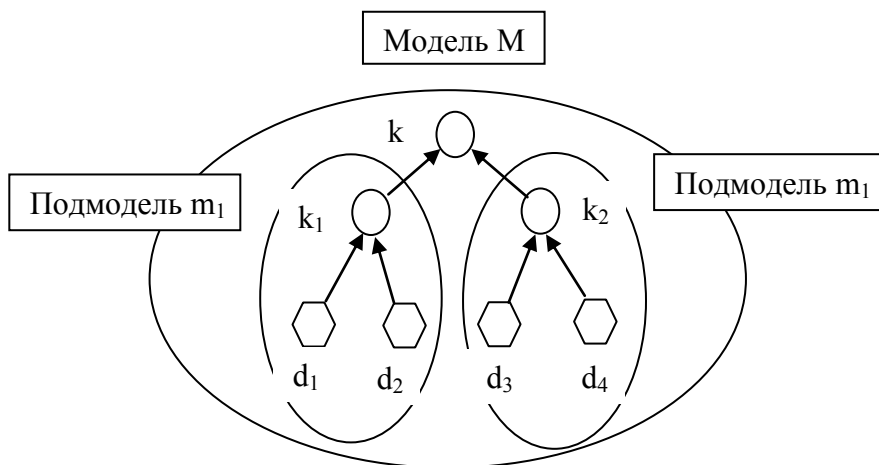


Рис. 1. Иерархия модели

Кроме того предполагается, что затраты зависят только от числа одновременно управляемых объектов, входящих в состав модуля, и не зависят от назначения отдельных модулей, т.е. секционная функция не изменяется при перестановке менеджеров. В частности это означает, что затраты $c(\{d_1, d_2\}, \{d_3, d_4\})$ менеджера k зависят только от набора групп $(\{d_1, d_2\}, \{d_3, d_4\})$, а не от того, в каком порядке эти группы записаны (в каком порядке пронумерованы непосредственно сами библиотечные модули), то есть $c(\{d_1, d_2\}, \{d_3, d_4\}) = c(\{d_3, d_4\}, \{d_1, d_2\})$.

Таким образом, класс секционных функций достаточно широк. В рамках этого класса построены различные модели оптимизации иерархии управления [4, 10], и есть основания полагать, что многие важные эффекты, связанные с иерархиями, могут быть промоделированы с по-

мощью секционных функций. Поэтому важно их изучение и решение задачи об оптимальной иерархии. В данной работе рассмотрено достаточное условие оптимальности иерархии частного вида – «конвейерной» (последовательной) иерархии.

Условие оптимальности «конвейерной» иерархии.

Определение 3. Иерархию назовем последовательной («конвейерной»), если в ней каждый менеджер управляет только двумя объектами, хотя бы один из которых является элементом БМ.

Рассмотрим последовательную иерархию H . Высший менеджер k в H управляет всеми составляющими модели: $s_H(k) = N$. Менеджеру k непосредственно подчинен некоторый элемент БМ d' и некоторый менеджер k' , то есть $s_H(k) = N = s(k') \cup \{d'\}$. Аналогично, менеджеру k' непосредственно подчинен некоторый элемент БМ d'' и некоторый менеджер k'' , причем $s(k'') = N \setminus \{d', d''\}$ и т.д.

На рис. 2 изображена последовательная иерархия общего вида.

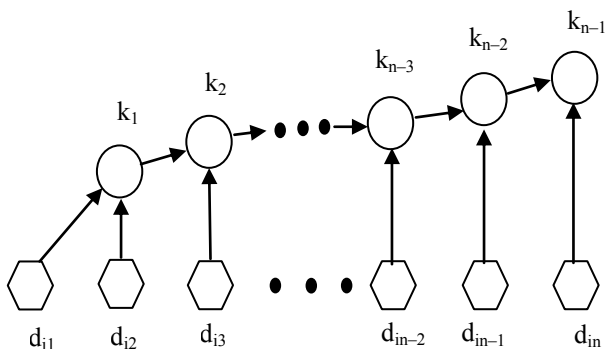


Рис. 2. Общий вид последовательной иерархии

В ней элементы БМ d_1, \dots, d_n расположены в порядке возрастания иерархии: сначала элемент с номером i_1 , затем элемент с номером i_2 , и так далее. Здесь (i_1, \dots, i_n) – некоторая перестановка чисел $(1, \dots, n)$. Следует заметить, что в последовательной иерархии имеется $n-1$ менеджер: $K = \{k_1, \dots, k_{n-1}\}$ (рис. 2).

Первый менеджер непосредственно управляет элементами d_{i_1} и d_{i_2} . Второй менеджер непосредственно управляет первым менеджером и элементом БМ d_{i_3} . Третий менеджер непосредственно управляет вторым менеджером и исполнителем d_{i_4} , и т.д.

Высший менеджер k_{n-1} непосредственно управляет исполнителем d_{i_n} и предыдущим менеджером k_{n-2} .

При различных условиях оптимальными будут различные иерархии. В частности, в работе [10] оптимальная иерархия сочетает в себе последовательную иерархию («conveyer belt») и иерархию классического вида, в котором менеджерам следующего уровня подчинены только менеджеры предыдущего уровня.

Таким образом, может быть интересна интерпретация последовательной иерархии как графа обработки информации. В последовательной иерархии (рис. 2) сначала свои операции производит первый менеджер, затем второй, и так далее. То есть в каждый момент времени в обработке задействован только один менеджер. Остальные менеджеры в это время простаивают. Таким образом, последовательная иерархия может соответствовать последовательной обработке информации, поступающей от элементов модели. Подобная иерархия требует большого времени на обработку, однако последовательная иерархия может справиться с обработкой интенсивного потока информации [12].

Поскольку под обработкой понимается вычисление ассоциативной функции, которую можно настраивать на любую последовательную иерархию, постольку для любого порядка элементов моделей (i_1, i_2, \dots, i_n) результат вычислений будет одним и тем же. Однако от функционально разных элементов БМ может поступать информация, требующая различного времени обработки или затрат на обработку. То есть, последовательная иерархия с минимальными затратами может соответствовать эффективному последовательному имитационному моделированию, когда временные характеристики полученных результатов не играют особой роли.

Выводы.

1. Выполнено формализованное описание модели, которое используется при имитационном моделировании основных параметров сложных системы в процессе ее функционирования.

2. Дано определение иерархии управления элементами модели, построенной на базе элементов библиотечных модулей, каждый из которых реализует простейшие функции системы. Функции управления в этом случае возлагаются на основной элемент модели – менеджер модели.

3. Предложен метод оценки оптимальности модели на основе минимизации затрат на управление объектами, входящими в состав модели.

4. Рассмотрены некоторые особенности построения имитационной модели при использовании последовательной иерархии в контуре управления элементами модели. Сделаны выводы о невозможности использования этого метода управления моделью при имитационном моделировании короткоживущих динамических объектов.

5. Направлением дальнейших исследований является разработка алгоритмической модели построения имитационной модели сложной системы с использованием иерархии управления элементами модели с целью ее применения в системах моделирования реальных сложно-организованных систем.

ЛИТЕРАТУРА

1. Томашевский В.Н. *Имитационное моделирование систем и процессов*. – К.: ИСДО, "ВПОЛ", 1994. – 124 с.
2. Кузьменко В.М. *Специальные языки программирования. Программные и инструментальные средства моделирования сложных систем*. – Х.: ХНУРЭ, 2001 – 488 с.
3. Воронин А.А., Мишин С.П. *Оптимальные иерархическиеструктуры*. – М.: ИПУ РАН, 2003. – 214 с.
4. *Интеллектуальные системы принятия проектных решений / А.В. Алексеев, А.Н. Борисов, Э.Р. Вилюмс, Н.Н. Слядзь, С.А. Фомин*. – Рига: Зинатне, 1997. – 364 с.
5. Советов Б. Я. *Моделирование систем*. – М.: Высшая школа, 2001. – 343 с.
6. Геловани В.А., Юрченко А.В. *Проблемы компьютерного моделирования*. – М.: МННИПУ, 1990. – 171 с.
7. *Постановка одного класу транспортних завдань великої розмірності / В.М. Кузьменко, М.І. Варсак, Ю.І. Ларіонов, О.Ф. Лановий; Харк.техн.ун-т радіоелектроніки*. – Х., 1996. – 8 с. – *Бібліогр. 3 назв. Деп. В ДНТБ України 22.04.96, № 961-Ук96*.
8. Лановий О.Ф. *Про методи імітаційного моделювання дискретних технологічних процесів // 2-а Міжнародна конференція «Теорія і техніка прийому, передачі та обробки інформації. Ч. 2. (м. Туапсе)». Тези доповідей*. – Х.: ХИРЭ, 1996. – С. 180-181.
9. Кузьменко В.М., Лановий О.Ф., Попов В.Ю., Пижова І.І. *Імітаційне моделювання у рішенні задач оптимізації проектування технологічного обладнання // 3-я Всесоюзна школа «Проектування автоматизованих систем контролю та управління складними об'єктами (м. Туапсе)». Тези доповідей*. – Х.: ХИРЭ, 1988. – 32 с.
10. Chow A.C., Zeigler B., 1994, *parallel DEVS: A Parallel, Hierarchical, Modular Modeling Formalism, Proceedings of the 1994 Winter Simulation Conference*.
11. Raczynski S., 1986, *PASION – Pascal-related Simulation Language for Small Systems, SIMULATION 46(6)*.
12. Moody D., Shanks G. *Improving the quality of data models: empirical validation of a quality management framework // ER 1998*. – P. 255-276.

Поступила 18.01.2006

Рецензент: доктор технических наук Н.М. Зацеркляный,
Харьковский национальный университет радиоэлектроники.