Моделирование с использованием типовых схем

Блочная конструкция модели

При машинной реализации любой из типовых математических схем *(D, F, P, Q, N, A-схем)* необходимо решить вопрос о взаимодействии блоков модели $M_м$ при использовании аналитического, имитационного или комбинированного (аналитико-имитационного) подходов.

Машинная модель $M_м$ системы *S* представляется как совокупность блоков $\{m_i\}$, $i=\overline{1,n}$. Каждый блок модели можно охарактеризовать конечным набором возможных состояний $\{z_0\}$, в которых он может находиться. Пусть в течение рассматриваемого интервала времени *(0, T),* т. е. времени прогона модели, блок изменяет состояния в моменты времени $t_i^{(j)} \le T$, где *j* — номер момента времени.

Моменты времени смены состояний блока m_i, можно условно разделить на три группы:

1) случайные моменты, связанные с внутренними свойствами части системы *S,* соответствующей данному блоку;

2) случайные моменты, связанные с изменением состояний других блоков (включая блоки, имитирующие воздействия внешней среды *E*);

3) детерминированные моменты, связанные с заданным расписанием функционирования блоков модели.

При моделировании для каждого блока модели m_i, $i=\overline{1,n}$, необходимо фиксировать момент очередного перехода блока в новое состояние $t_i^{(j)}$ и номер этого состояния s_i-, образуя при этом массив состояний. Этот массив отражает динамику функционирования модели системы, так как в нем фиксируются все изменения в процессе функционирования моделируемой системы *S* по времени. В начале моделирования в массив состояний должны быть занесены исходные состояния, заданные начальными условиями.

При машинной реализации модели $M_м$ ее блоки, имеющие аналогичные функции, могут быть представлены в виде отдельных программных модулей. Работа каждого такого модуля имитирует работу всех однотипных блоков.

Типовая укрупненная схема моделирующего алгоритма, построенного по блочному принципу, для систем с дискретными событиями рис. 1.

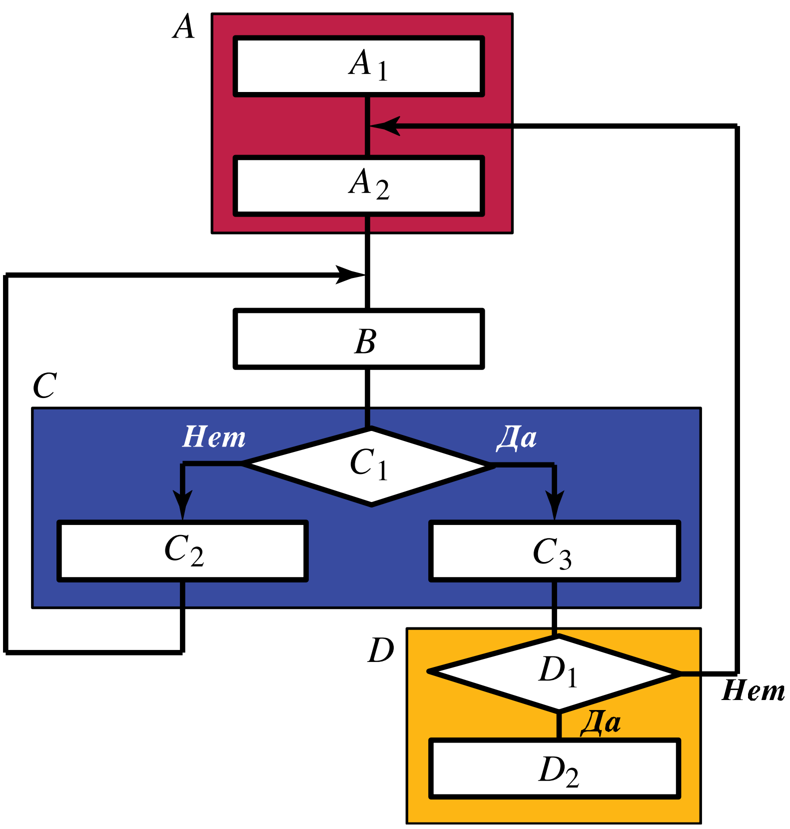

Рис. 1.

Эта схема содержит следующие укрупненные модули: *А* — модуль задания начальных значений состояний, содержащий два подмодуля *(А1* — для задания начальных состояний моделируемого варианта и *А2*, — для задания начальных состояний для одного прогона модели); *В* — модуль определения очередного момента смены состояния, осуществляющий просмотр массива состояний и выбирающий блок модели m_i, $i=\overline{1,n}$, с минимальным временем смены состояния min $t_i^{(j)}$; *С* — модуль логического переключения, содержащий три подмодуля (*C_1* — для логического перехода по номеру блока модели *i* или по времени *Т,* т. е. для решения вопроса о завершении прогона; *C_2* — для фиксации информации о состояниях, меняющихся при просмотре блока, а также для определения момента следующей смены состояния блока *m_i* и номера следующего особого состояния *s_0;* *C_3* — для завершения прогона в случае, когда $t_i^{(j)} \geq T$, фиксации и предварительной обработки результатов моделирования); *D* — модуль управления и обработки, содержащий два подмодуля *(D_1* — для проверки окончания исследования варианта модели *$M_м$* по заданному числу прогонов или по точности результатов моделирования; *D_2* — для окончательной обработки информации, полученной на модели *$M_м$,* и выдачи результатов моделирования).

Построение моделирующего алгоритма по блочному принципу позволяет за счет организации программных модулей уменьшить затраты времени на моделирование системы *S,* так как машинное время в этом случае не тратится на просмотр

повторяющихся ситуаций. Кроме того, данная схема алгоритма получается проще.

Если говорить о перспективах, то блочный подход создает хорошую основу для автоматизации имитационных экспериментов с моделями систем, которая может полностью или частично охватывать этапы формализации процесса функционирования системы *S,* подготовки исходных данных для моделирования, анализа свойств машинной модели $M_м$ системы, планирования и проведения машинных экспериментов, обработки и интерпретации результатов моделирования системы.

Автоматизация процесса моделирования создаст перспективы использования моделирования в качестве инструмента повседневной работы системного специалиста.

Моделирование функционирования систем на базе Q-схем.

Характерная ситуация в работе таких систем — появление заявок (требований) на обслуживание и завершение обслуживания в случайные моменты времени, т.е. стохастический характер процесса их функционирования. В общем случае моменты поступления заявок в систему *S* из внешней среды *E* образуют входящий поток, а моменты окончания обслуживания образуют выходящий поток обслуженных заявок.

Формализуя какую-либо реальную систему с помощью *Q-схемы,* необходимо построить структуру такой системы. В качестве элементов структуры *Q-схем* рассматриваются элементы трех типов:

И — источники;

Н — накопители;

К — каналы обслуживания заявок.

Q-схему можно считать заданной, если определены:

1. потоки событий (входящие потоки заявок и потоки обслуживании для каждого *Н* и *К)*;
2. структура системы *S* (число фаз $L^{Ф}$, число каналов обслуживания L^{K}, число накопителей L^{H} каждой из $L^{Ф}$ фаз обслуживания заявок и связи *И*, *Н* и *К*);
3. алгоритмы функционирования системы (дисциплины ожидания заявок в *Н* и выбора на обслуживание *К*, правила ухода заявок из *Н* и *К*).

При моделировании систем, формализуемых в виде *Q-схем,* часто возникают задачи имитации потоков заявок с некоторыми ограничениями, позволяющими упростить как математическое описание, так и программную реализацию генераторов потоков заявок.

Так, для ординарных потоков с ограниченным последействием интервалы между моментами поступления заявок являются независимыми и совместная плотность распределения может быть представлена в виде произведения частных законов распределении

$$f(y_1, y_2, y_3, \ldots, y_n) = f(y_1) f(y_2) \ldots f(y_n)$$

Где $f_i(y_i), i = \overline{1, k}$, при *i>1* являются условными функциями плотности величин y_i при условии, что в момент начала *i*-го интервала поступит заявка. Относительно начального момента времени t_0 никаких предположений не делается, поэтому функция $f_1(y_1)$ — безусловная.

Порядок моделирования моментов появления заявок в стационарном потоке с ограниченным последействием следующий. Из последовательности случайных чисел, равномерно распределенных на интервале (0, 1), выбирается случайная величина и формируется первый интервал y_1 в соответствии с

$$f_1(y_1) = \lambda\left(1 - \int_0^{y_1} f(y)dy\right)$$

где λ - интенсивность потока событий, любым из способов формирования случайной величины. Момент наступления первого события $t_1 = t_0 + y_1$ следующие моменты появления событий определяются как

$$t_2 = t_1 + y_2, ..., t_k = t_{k-1} + y_k$$

где y_k — случайная величина с плотностью $f(y)$.

При формирования потока событий, описываемого нестационарным распределением Пуассона с мгновенной плотностью потока $\lambda(t)$, плотность распределения длины первого интервала

$$f_1(y_1) = a'(t_0, y_1)e^{-a(t_0, y_1)}$$

где $a = \int_{t_0}^{t_0+\Delta t} \lambda(t)dt$ — математическое ожидание числа событий на интервале $(t_0, t_0+\Delta t)$. Дальнейшая методика моделирования случайной величины y_i при $i>1$ аналогична формированию y_i с использованием условной функции распределения.

Моделирующий алгоритм должен адекватно отражать процесс функционирования системы S и в то же время не

создавать трудностей при машинной реализации модели M_M. При этом моделирующий алгоритм должен отвечать следующим основным требованиям:

- обладать универсальностью относительно структуры, алгоритмов функционирования и параметров системы S;
- обеспечивать одновременную (в один и тот же момент системного времени) и независимую работу необходимого числа элементов системы S;
- укладываться в приемлемые затраты ресурсов ЭВМ (машинного времени и памяти) для реализации машинного эксперимента;
- проводить разбиение на достаточно автономные логические части, т. е. возможность построения блочной структуры алгоритма;
- гарантировать выполнение рекуррентного правила — событие, происходящее в момент времени t_k, может моделироваться только после того, как промоделированы все события, произошедшие в момент времени $t_{k-1}<t_k$.

Существует два основных принципа построения моделирующих алгоритмов: «принцип Δf» и «принцип δz». При построении моделирующего алгоритма *Q-схемы* по «принципу Δf», т. е. алгоритма с детерминированным шагом, необходимо для построения адекватной модели M_M определить минимальный интервал времени между соседними событиями $\Delta t' = min\{u_i\}$ (во входящих потоках и потоках обслуживании) и принять, что шаг моделирования равен $\Delta t'$.

В моделирующих алгоритмах, построенных по «принципу ***δz»,*** **т. е. в алгоритмах со случайным шагом, элементы** ***Q-схемы*** **просматриваются при моделировании только в моменты особых состояний (в моменты появления заявок из** ***И*** **или изменения состояний** ***К).*** **При этом длительность шага** ***Δf = var*** **зависит как от особенностей самой системы** ***S*****, так и от воздействий внешней среды** ***Е.***

Моделирующие алгоритмы со случайным шагом могут быть реализованы синхронным и асинхронным способами.

При синхронном способе один из элементов ***Q-схемы*** **(*И*, *Н* или *К*) выбирается в качестве ведущего и по нему «синхронизируется» весь процесс моделирования.**

При асинхронном способе построения моделирующего алгоритма ведущий (синхронизирующий) элемент не используется, а очередному шагу моделирования (просмотру элементов ***Q-схемы)*** **может соответствовать любое особое состояние всего множества элементов** ***И*, *Н*** **и** ***К*****. При этом просмотр элементов** ***Q-схемы*** **организован так, что при каждом особом состоянии либо циклически просматриваются все элементы, либо спорадически — только те, которые могут в этом случае изменить свое состояние (просмотр с прогнозированием).**

Классификация возможных способов построения моделирующих алгоритмов ***Q-схем*** **приведена на рис. 2.**

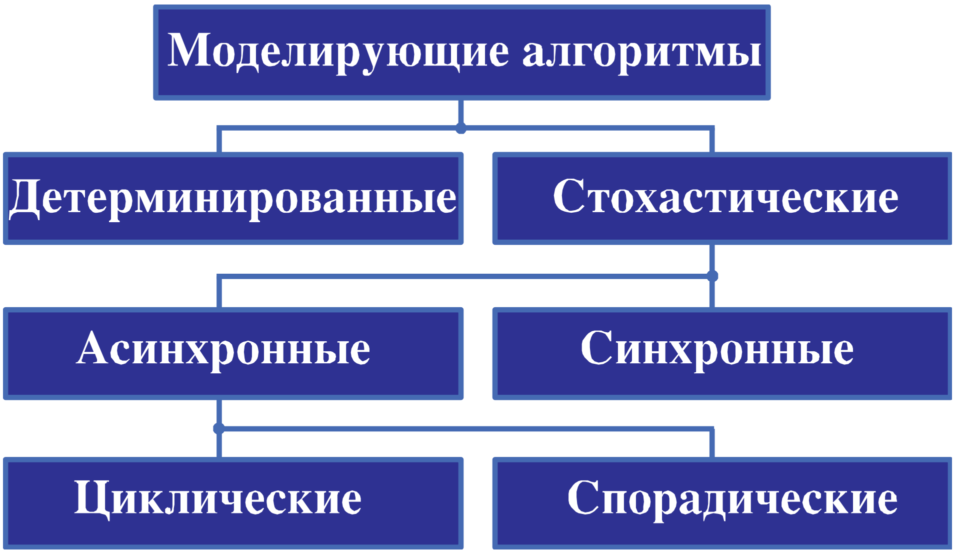

Рис. 2.

Математическое обеспечение и ресурсные возможности современных ЭВМ позволяют достаточно эффективно провести моделирование различных систем, формализуемых в виде *Q-схем,* используя либо пакеты прикладных программ, созданные на базе алгоритмических языков общего назначения, либо специализированные языки имитационного моделирования. При имитации процесса функционирования *Q-схемы* на ЭВМ требуется организовать массив состояний.

Моделирующие алгоритмы и способы их модификации могут быть использованы для моделирования широкого класса систем. Однако эти алгоритмы будут отличаться по сложности реализации, затратам машинного времени и необходимого объема памяти ЭВМ.

Сложность различных моделирующих алгоритмов *Q-схем,* в основу построения которых положены детерминированный и асинхронный циклический принцип построения алгоритмов наиболее просты с точки зрения логики их построения, так как при этом используется перебор всех элементов *Q-схемы* на каждом шаге. Трудности возникают с машинной реализацией этих

алгоритмов вследствие увеличения затрат машинного времени на моделирование, так как просматриваются все состояния элементов *Q-схемы* (по «принципу Δt» или по «принципу δz»). Затраты машинного времени на моделирование существенно увеличиваются при построении детерминированных моделирующих алгоритмов *Q-схем,* элементы которых функционируют в различных масштабах времени, например когда длительности обслуживания заявок каналами многоканальной *Q-схемы* значительно отличаются друг от друга.

В стохастическом синхронном алгоритме рассматриваются прошлые изменения состояний элементов *Q-схемы,* которые произошли с момента предыдущего просмотра состояний, что несколько усложняет логику этих алгоритмов.

Асинхронный спорадический алгоритм позволяет просматривать при моделировании только те элементы *Q-схемы,* изменения состояний которых могли иметь место на данном интервале системного времени, что приводит к некоторому упрощению этих моделирующих алгоритмов по сравнению с синхронными алгоритмами и существенному уменьшению затрат машинного времени по сравнению с детерминированными и циклическими алгоритмами.

К настоящему времени накоплен значительный опыт моделирования *Q-схем* (при их классическом рассмотрении или в различных приложениях). Рассмотренные моделирующие алгоритмы позволяют практически отразить все возможные варианты многофазных и многоканальных *Q-схем,* а также провести исследование всего спектра их вероятностно-

временных характеристик, различных выходных характеристик, интересующих исследователя или разработчика системы *S*.