

**Тема 2** «Компьютерное моделирование информационных систем  
с использованием типовых схем»

**Лекция 1** «Иерархические модели функционирования систем»

1. Блочная конструкция модели
2. Моделирование функционирования систем на базе Q-схем.

## Введение.

Объекты информационных систем характеризуются сложностью структуры, алгоритмов поведения, многопараметричностью, что, естественно, приводит к сложности их машинных моделей; это требует при их разработке построения иерархических модульных конструкций, а также использования формального описания внутрисистемных процессов. Типовые математические схемы являются связующим звеном в цепочке «концептуальная модель — машинная модель», позволяя эффективно решать при моделировании проблемы взаимодействия заказчика и исполнителя.

### 1. 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  $M_m$  при использовании аналитического, имитационного или комбинированного (аналитико-имитационного) подходов.

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

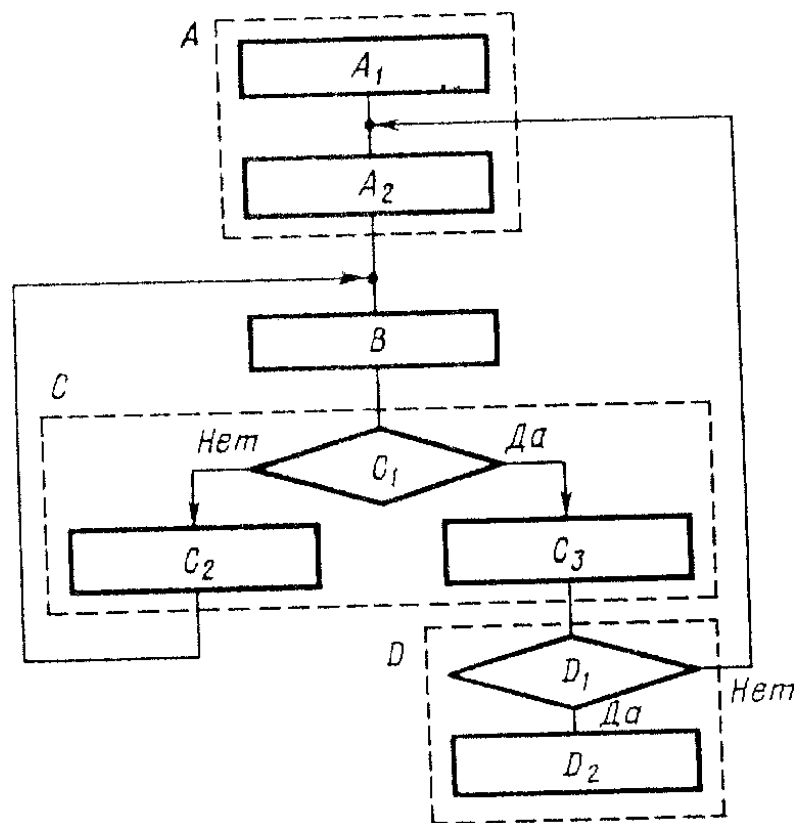


Рис. 1.

Эта схема содержит следующие укрупненные модули:  $A$  — модуль задания начальных значений состояний, содержащий два подмодуля ( $A_1$  — для задания начальных состояний моделируемого варианта и  $A_2$ , — для задания начальных состояний для одного прогона модели);  $B$  — 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  $\min t_i^{(j)}$ ;  $C$  — модуль логического переключения, содержащий три подмодуля ( $C_1$  — для логического перехода по номеру блока модели  $i$  или по времени  $T$ , 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  $M_m$  по заданному числу прогонов или по точности результатов моделирования;  $D_2$  — для окончательной обработки информации, полученной на модели  $M_m$ , 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  $M_m$  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.

## 2. 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:

$I$  — источники;

$H$  — накопители;

$K$  — каналы обслуживания заявок.

$Q$ -схему можно считать заданной, если определены:

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

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 последствием интервалы между моментами поступления заявок являются независимыми и совместная плотность распределения может быть представлена в виде произведения частных законов распределения

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

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  $f_i(y_i)$  — безусловная.

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

$$f_i(y_i) = \lambda \left( 1 - \int_0^{y_i} f(y) dy \right) \quad (14.1)$$

где  $\lambda$  - 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

$$t_2 = t_1 + y_2, \dots, t_k = t_{k-1} + y_k \quad (14.2)$$

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

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

$$f_i(y_i) = a'(t_0, y_i) e^{-a(t_0, y_i)} \quad (14.3)$$

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  $\Delta f$ » и «принцип  $\delta z$ ». При построении моделирующего алгоритма  $Q$ -схемы по «принципу  $\Delta f$ », т. е. алгоритма с детерминированным шагом, необходимо для построения адекватной модели  $M_m$  определить минимальный интервал времени между соседними событиями  $\Delta t' = \min\{u_{ij}\}$  (во входящих потоках и потоках обслуживания) и принять, что шаг моделирования равен  $\Delta t'$ .

В моделирующих алгоритмах, построенных по «принципу  $\delta z$ », т. е. в алгоритмах со случайным шагом, элементы  $Q$ -схемы просматриваются при моделировании только в моменты особых состояний (в моменты появления заявок из  $I$  или изменения состояний  $K$ ). При этом длительность шага  $\Delta f = var$  зависит как от особенностей самой системы  $S$ , так и от воздействий внешней среды  $E$ .

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

При синхронном способе один из элементов  $Q$ -схемы ( $I$ ,  $H$  или  $K$ ) 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 моделиро-

вания (просмотру элементов *Q*-схемы) может соответствовать любое особое состояние всего множества элементов *I*, *H* и *K*. 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.

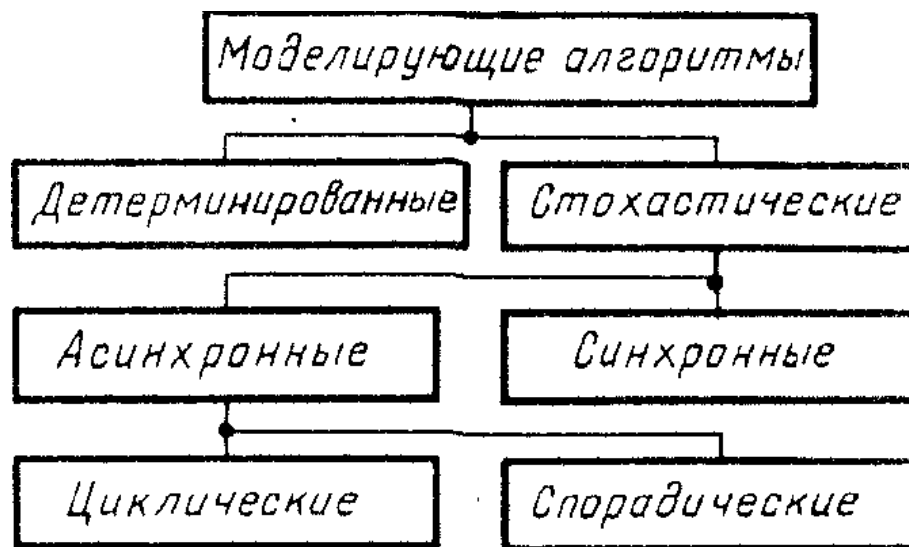


Рис. 2.

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 по-



строения, 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  $\Delta t$ » или по «принципу  $\delta z$ »). 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  $S$ .

## **Лекция 2 «Моделирование функционирования систем N-схем и A-схем»**

1. Структурный подход на базе N-схем
2. Формализация на базе A-схем

## Введение.

Характерной особенностью  $N$ -схем является то, что с их помощью можно моделировать процессы в системах  $S$ , в которых происходит последовательная смена дискретных состояний, в том числе если эта смена происходит при выполнении разнообразных условий. Таким образом, с использованием  $N$ -схем могут быть описаны системы  $S$ , относящиеся к разным классам: аппаратные, физические, программные, экономические и т. д. С помощью  $A$ -схем можно моделировать системы обобщенного агрегативного подхода.

### 1. Структурный подход на базе $N$ -схем

Применение аппарата  $N$ -схем позволяет осуществить структурный подход к построению имитационной модели системы  $S$ , при котором обеспечиваются наглядность модели, модульный принцип ее разработки (сборки), возможность перехода к автоматизированной интерактивной процедуре.

Построение  $N$ -схемы происходит формально: состояниям системы соответствуют позиции  $N$ -схемы, событиям — переходы. Нанесем маркировку, соответствующую такому состоянию системы, при котором каналы свободны, операторы не заняты, в системе нет заказов (рис. 1.).

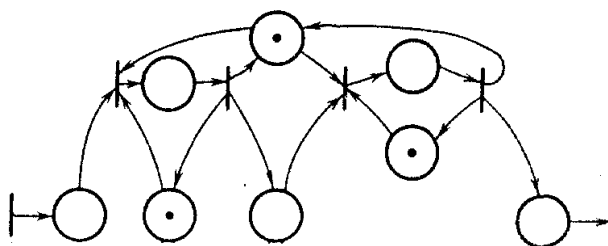


Рис. 1.

Видно, что для выполнения каждого события (перехода) необходимо выполнение определенных условий. Эти условия в  $N$ -схемах (сетях Петри) называются *предусловиями*. Выполнение события может вызвать нарушение предусловий и привести к выполнению условий для совершения других событий — *постусловий*.

Процесс моделирования заключается в последовательном вычислении маркировок, получающихся в результате выполнения событий (переходов). События, по которым нет предусловий, являются входами *N-схемы*. Каждый вход должен быть присоединен к модели, генерирующей запуск события в соответствии с условиями, определяемыми моделируемой реальностью. В частности, это может быть другая *N-схема*, моделирующая процесс появления этих событий.

В *N-схемах* два или несколько разрушенных не взаимодействующих событий могут происходить независимо друг от друга, т. е. *N-схемам* и их моделям свойствен параллелизм, или одновременность. Синхронизировать события, пока этого не требует моделируемая система, нет нужды. Таким образом, *N-схемы* удобны для моделирования системы с распределенным управлением, в которых несколько процессов выполняются одновременно.

Другая важная особенность *N-схем* — это их асинхронная природа. Внутри *N-схемы* отсутствует измерение времени. Для простоты обычно вводят следующее ограничение. Запуск перехода (и соответствующего события) рассматривается как мгновенное событие, занимающее нулевое время, а возникновение двух событий одновременно невозможно. Моделируемое таким образом событие называется *примитивным* (примитивные события мгновенны и не одновременны).

*Непримитивными* называются такие события, длительность которых отлична от нуля. Любое непримитивное событие может быть представлено в виде двух примитивных событий: «начало непримитивного события», «конец непримитивного события» — и состояния (условия) «непримитивное событие происходит».

Ранее упоминалось, что в *N-схемах* все разрешенные переходы срабатывают одновременно и независимо. Однако с помощью *N-схем* можно моделировать и такие системы *S*, в которых порядок запуска в разрешенных переходах имеет существенное значение. Ситуация, в которой невозможно одновременное выполнение двух разрешенных переходов, изображена на рис. 2., где два разре-

шенных перехода  $d_j$  и  $d_k$  находятся в конфликте. Может быть запущен только один из них, так как при запуске он удаляет метку из общего входа и запрещает другой переход.

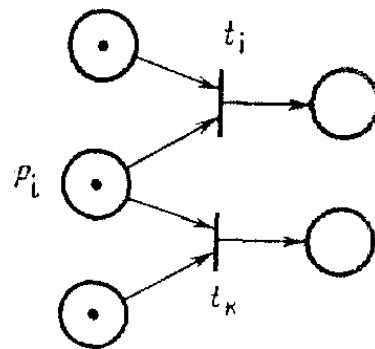


Рис. 2

Возможность моделирования параллелизма и довольно простые процедуры объединения подсистем, представленных  $N$ -схемами, делают их весьма полезным инструментом моделирования сложных аппаратно-программных информационно-вычислительных комплексов и сетей, состоящих из большого количества одинаковых компонент.

$N$ -схема представляет собой формализованное описание процесса функционирования системы  $S$ , причем структура  $N$ -схем отражает причинно-следственные связи в системе  $S$ , а совместно с начальной маркировкой — процессы, которые в этой системе происходят. Таким образом, переход от  $N$ -схем к моделирующей программе может производиться формальным путем, т. е. автоматически, с использованием специального языка и транслятора.

Для моделирования конкретных систем  $S$  в узкоспециализированной области количество возможных событий-переходов ограничено. Можно составить список стандартных событий и соответствующий ему набор программных модулей. Вместе со средствами описания структуры  $N$ -схемы, поясняющей взаимодействия, такой набор составит пакет моделирования систем в специализированной области. Известны системы, построенные по такому принципу для моделирования аппаратных и программных средств вычислительных систем, для моделирования протоколов связи. Известны также примеры успешного примене-

ния *N-схем* (сетей Петри) для исследования социальных, экономических систем, сложных физических и химических процессов.

С использованием *N-схем* осуществляется структурный подход к построению имитационной модели, при котором обеспечиваются наглядность модели, модульный принцип ее разработки (сборки), возможность перехода к автоматизированной интерактивной процедуре проектирования.

Еще большие возможности для моделирования сложных систем дают такие расширения *N-схем*, как *E-сети*, которые обозначим как *N<sub>E</sub>-схемы*. В отличие от временных сетей в *E-сетях* определено дополнительно четыре типа переходов:

- разветвление,
- объединение,
- управляемое разветвление,
- приоритетное объединение.

Одним из основных вопросов, который надо решить разработчику имитационной модели процесса, формализуемого на базе *N-схем*, является выбор языка программирования. Реализация модулей *N<sub>E</sub>-схем* на машинно-ориентированном языке или же языках общего назначения позволяет снизить затраты машинного времени и оперативной памяти при моделировании систем, но при этом следует учитывать высокую трудоемкость разработки библиотеки моделирующих подпрограмм. Этот недостаток устраняется при использовании для моделирования системы *S*, формализованной на базе *N-схем*, языков имитационного моделирования.

Программная реализация моделей систем *S* на базе расширенных *N-схем* (*N<sub>E</sub>-схем*) более сложна по сравнению с программированием моделей на основе обычных сетей Петри. Для упрощения перехода к моделирующей программе рационально использовать языки имитационного моделирования.

## 2. Формализация на базе А-схем

А-схемы представляются в виде кусочно-линейных агрегатов (КЛА) и позволяют описать достаточно широкий класс процессов дающих возможность построения на их основе не только имитационных, но и аналитических моделей. А-схемы рассматривается как объект, который в каждый момент времени характеризуется внутренними состояниями  $z(t) \in Z$ ; в изолированные моменты времени на вход агрегата А могут поступать входные сигналы  $x(t) \in X$ , а с его выхода могут сниматься выходные сигналы  $y(t) \in Y$ . Класс КЛА выделяется с помощью конкретизации структуры множеств  $Z, X, Y$ , т. е. пространств состояний, входных и выходных сигналов соответственно, а также операторов переходов  $V, U, W$  и выходов  $G$ .

Если мы хотим описать процесс функционирования прибора обслуживания как КЛА, то основное состояние будет соответствовать числу заявок в приборе (П), в накопителе (Н) и канале (К), а вектор дополнительных координат будет содержать информацию о длительности пребывания заявки, ее приоритетности и др., т. е. ту информацию, значение которой необходимо для описания процесса  $z(t)$ .

Основные преимущества агрегативного подхода состоят в том, что в руки разработчиков моделей и пользователей дается одна и та же формальная схема, т. е. *А-схема*. Это позволяет использовать результаты математических исследований процессов, описывающих функционирование агрегативных систем, при создании моделирующих алгоритмов и их программной реализации на ЭВМ. В настоящее время имеются разработки математического обеспечения, в основу которого положен агрегативный подход. Но при этом у пользователя всегда должна оставаться свобода в переходе от концептуальной к формальной модели. Таким образом, имеется возможность многовариантного представления процесса функционирования некоторой системы  $S$  в виде модели  $M$ , построенной на основе *А-схем*.

В основу моделирующего алгоритма *А-схемы* положен принцип просмотра состояний модели в моменты скачков, т. е. «принцип  $\delta z$ » («принцип особых со-

стояний»). Работа такого блока сводится к выбору типа агрегата ( $A^E$ ,  $A^K$ ,  $A^H$ ,  $L^P$  и  $A^C$ ), для которого реализуется дальнейшее «продвижение» при моделировании.

В схеме моделирующего алгоритма, имитирующего воздействие на систему  $S$  внешней среды  $E$ , определяется, какое событие имело место, поступление или выдача сигнала из внешней среды, т. е. заявки входного потока в  $A$ -схему. При наступлении времени выдачи заявки она выдается в  $A$ -схему и генерируется интервал времени между моментом поступления новой заявки, при этом учитываются схемы моделирующих алгоритмов, имитирующих работу всех агрегатов. Работа этих схем полностью соответствует описанию процесса функционирования агрегатов. А также учитывается работа схем агрегатов выполняющих вспомогательные функции сопряжения всех агрегатов. Они реализуют взаимодействие основных агрегатов, разрешая или запрещая передачу сигналов между ними в зависимости от ситуации с учетом правил обмена сигналами в  $A$ -схеме. При этом в схемах предусмотрено тестирование ошибок, связанных с нарушением при задании исходных данных этих правил обмена сигналами в  $A$ -схеме.

Применение агрегативного подхода при моделировании систем дает ряд преимуществ по сравнению с другими, менее универсальными подходами.

Так, агрегативный подход в силу модульной структуры модели и дискретного характера обмена сигналами дает возможность использовать внешнюю память ЭВМ для хранения сведений о моделируемых объектах, что в значительной степени снижает ограничения по сложности, возникающие при попытке представить процесс функционирования моделируемой системы  $S$  в целом как последовательность взаимосвязанных системных событий для записи его в виде моделирующего алгоритма или на языке имитационного моделирования.

Объем программ имитации мало зависит от сложности моделируемого объекта, которая определяет лишь число операций, требуемых для реализации машинной модели  $M_m$  и объем памяти, необходимый для хранения сведений об агрегатах и их связях.



При агрегативном подходе возникают и некоторые трудности, связанные с организацией диалога пользователя с имитационной системой, так как представление моделируемой системы в виде *A-схемы* предполагает и структуризацию в соответствующем виде входных данных. Следовательно, пользователь, как и разработчик модели  $M_m$  должен владеть языком агрегативных систем для решения своих задач.

В перспективе агрегативный подход создает основу для автоматизации машинных экспериментов. Такая автоматизация может полностью или частично охватывать этапы формализации процесса функционирования системы  $S$ , подготовки исходных данных для моделирования, планирования и проведения машинных экспериментов, обработки и интерпретации результатов моделирования. Процесс автоматизации моделирования будет постепенным и поэтапным. Решение задачи автоматизации создает перспективы применения моделирования в качестве инструмента для повседневной работы инженера системотехника в сфере проектирования и эксплуатации информационных систем, систем сбора и обработки информации, систем автоматизации проектирования, систем автоматизации научных исследований и комплексных испытаний и т. д.