

# Моделирование функционирования систем N-схем и А-схем

## 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.).

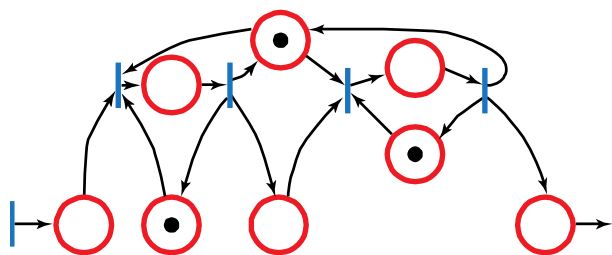


Рис. 1.

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 невзаимодействующих 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 неодновременны).

**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 разрешенных 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.

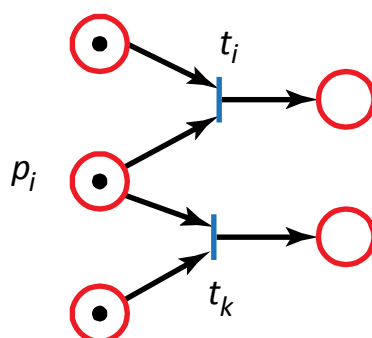


Рис. 2

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 применения  $N$ -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  $N_E$ -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  $N_E$ -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 ( $N_E$ -схем) более сложна по сравнению с программированием моделей на основе обычных сетей Петри. Для упрощения перехода к моделирующей программе рационально использовать языки имитационного моделирования.

### 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  $A$  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. *A-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 *A-схем*.

В основу моделирующего алгоритма *A-схемы* положен принцип просмотра состояний модели в моменты скачков, т. е.

«принцип  $\delta z$ » («принцип особых состояний»). 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 *А-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.**