

СПб ГУТ)))

Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича



Моделирование информационных систем

Санкт-Петербург

Моделирование систем

Лектор

Козлова Ольга Александровна

Литература

- Советов Б.Я., Яковлев А. М.
Моделирование систем, 3-е издание,
переработанное и дополненное. - М.,
2001 -374с.

Тема 1

Основные понятия моделирования систем

Целью создания моделей является изучение, описание, проектирование или оптимизация некоторого объекта или процесса. Например, в общественной деятельности модели используют для проведения анализа при принятии решений.

Модель - это такой материальный или мысленно представляемый объект, который в процессе исследования замещает объект оригинал так, что его непосредственное изучение дает новые знания об объекте-оригинале.

Моделирование - это замещение одного объекта другим с целью изучения важнейших свойств замещаемого объекта. Замещаемый объект называется *оригиналом*, замещающий - *моделью*.

В математической теории модель определяется как результат отображения одной абстрактной математической структуры на другую, также абстрактную, либо как результат интерпретации первой модели в терминах и образах второй.

Главная особенность моделирования в том, что это метод опосредованного познания с помощью объектов-заместителей. Модель выступает как своеобразный инструмент для познания, который исследователь ставит между собой и объектом и с помощью которого изучает интересующий его объект.

Модели могут быть качественно различными, они образуют иерархию, в которой модель более высокого уровня (например, теория) содержит модели нижних уровней (гипотезы) как свои части, элементы.

Моделирование представляет собой процесс построения, изучения и применения моделей. Этот процесс включает три элемента: субъект (исследователь), объект исследования, модель и состоит из следующих основных этапов.

1. Конструирование или поиск в реальном мире подходящей модели исследуемого объекта.
2. Замена объекта исследования моделью и проведение «модельных» экспериментов.
3. «Перенос знаний с модели на оригинал». При этом знания о модели должны быть скорректированы с учетом тех свойств объекта-оригинала, которые не нашли отражения или были изменены при построении модели.
4. Практическая проверка получаемых с помощью моделей знаний о реальном объекте.

Основные требования к модели:

1. наглядность построения;
2. обозримость основных свойств и отношений;
3. доступность ее для исследования или воспроизведения;
4. простота исследования, воспроизведения;
5. сохранение информации, содержащиеся в оригинале (с

точностью рассматриваемых при построении модели гипотез) и получение новой информации.

Проблема моделирования состоит из трех задач:

1. построение модели (эта задача менее формализуема и конструктивна, в том смысле, что нет алгоритма для построения моделей);
2. исследование модели (эта задача более формализуема, имеются методы исследования различных классов моделей);
3. использование модели (конструктивная и конкретизируемая задача).

Свойства модели:

1. конечность: модель отображает оригинал лишь в конечном числе его отношений и, кроме того, ресурсы моделирования конечны;
2. упрощенность: модель отображает только существенные стороны объекта;
3. приближенность: действительность отображается моделью грубо или приблизительно;
4. адекватность: модель успешно описывает моделируемую систему;
5. информативность: модель должна содержать достаточную информацию о системе - в рамках гипотез, принятых при построении модели.

Моделирование - циклический процесс. Это означает, что за первым четырехэтапным циклом может последовать второй, третий и т.д. При этом знания об исследуемом объекте расширяются и уточняются, а исходная модель постепенно совершенствуется.

Моделирование тесно связано с такими гносеологическими категориями, как: *абстракция, формализация, идеализация, аналогия, гипотеза* и др. Процесс моделирования неизбежно включает и построение абстракций, и умозаключения по аналогии, и конструирование научных гипотез.

Методология моделирования долгое время развивалась независимо в недрах отдельных наук. Отсутствовала единая система понятий, единая терминология. Лишь постепенно стала осознаваться роль моделирования как универсального метода научного познания, как важной гносеологической категории.

Однако моделирование не единственный источник знаний об объекте. Процесс моделирования «погружен» в общий процесс познания. Это обстоятельство особенно важно учитывать на завершающей стадии, когда происходит объединение и обобщение результатов исследования, получаемых на основе многообразных средств познания.

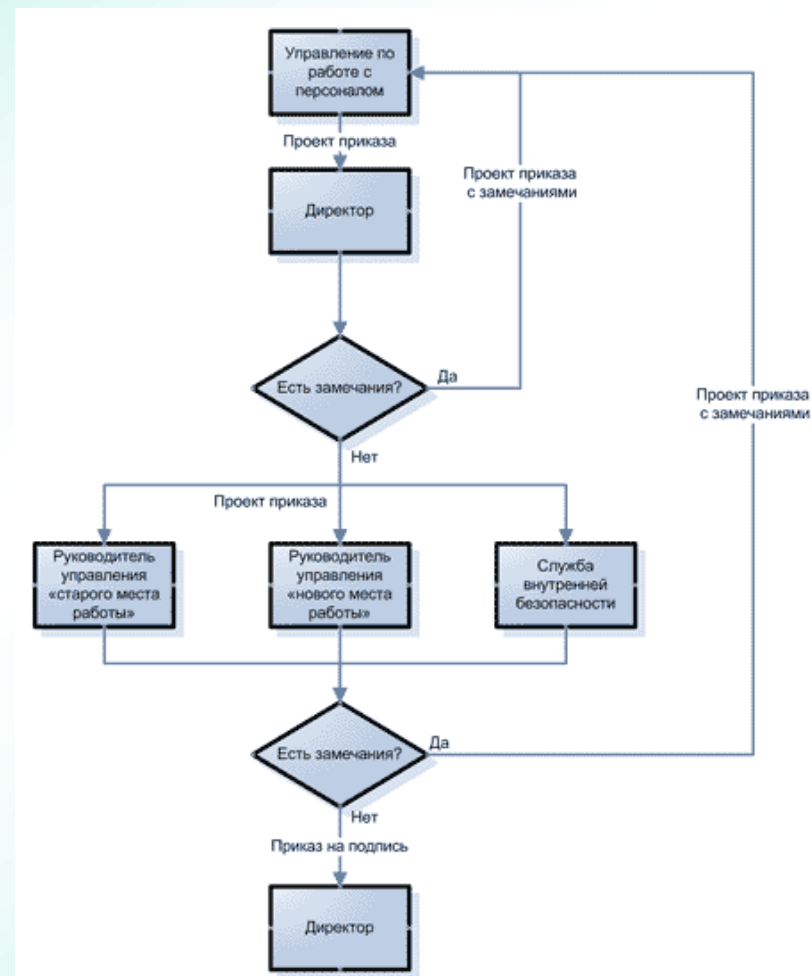
Моделирование тесно связано с такими гносеологическими категориями, как: ***абстракция, формализация, идеализация, аналогия, гипотеза*** и др. Процесс моделирования неизбежно включает и построение абстракций, и умозаключения по аналогии, и конструирование научных гипотез.

Абстракция служит для представления существенных свойств объекта без детального учета его внутренней организации и конкретной реализации.

Модель отражает только те стороны явлений, которые существенны в данном исследовании, и позволяет абстрагироваться от второстепенных факторов. Элементарная форма абстракции - типизация объектов.

Формализация - отображение

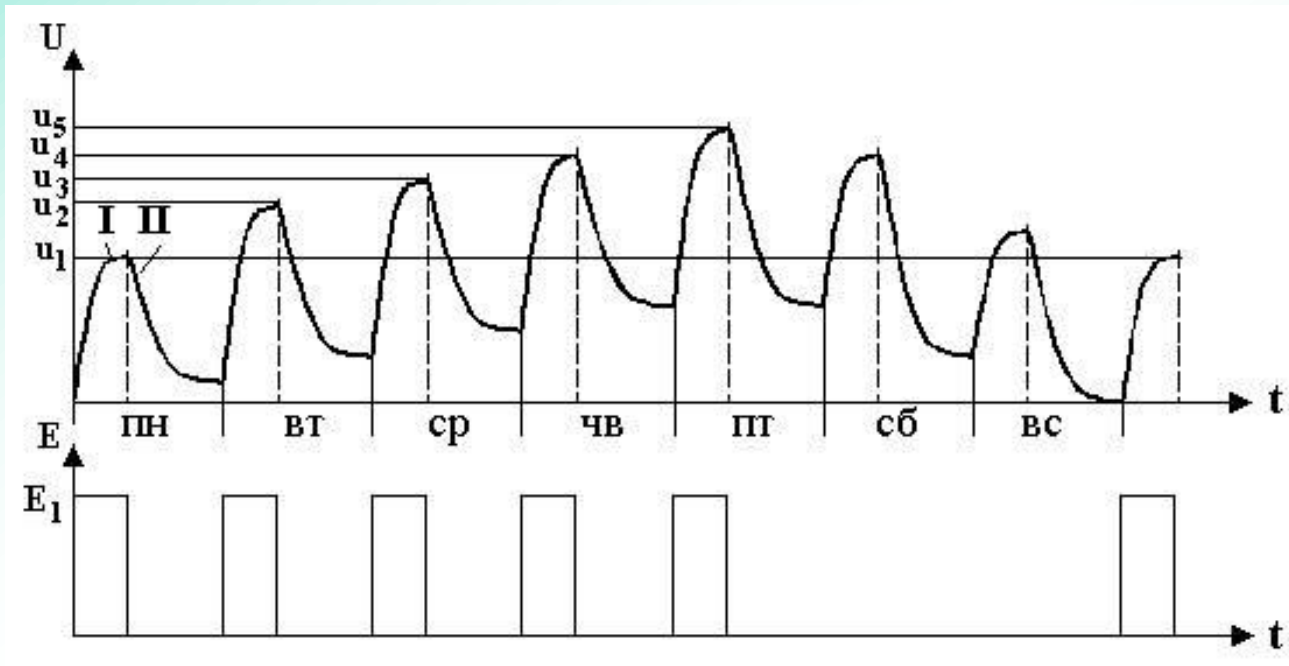
объекта или явления в знаковой форме
какого-либо искусственного языка
(например, математики) с целью
исследования объектов и свойств через
формальное исследование
соответствующих знаков. Термин
«формальный» означает, что объекты и
операции над ними рассматриваются
чисто формально, без какой бы то ни
было содержательной интерпретации
СИМВОЛОВ.



Пример формализации
документооборота

Идеализация - это мысленное конструирование объектов, которые

в реальности не существуют (например, идеальный газ, идеальный импульс, неограниченная память и др.). В результате идеализации реальные объекты лишаются некоторых присущих им свойств и наделяются гипотетическими свойствами.



Пример идеализации среднесуточной температуры

Аналогия - прием, используемый в методах познания состоящий в

том, что знания о предметах и явлениях выявляются на основании изучения сходных с ними в каком-либо отношении (по структуре, составу, принципу действия, используемым средствам, характеристикам) объектов.

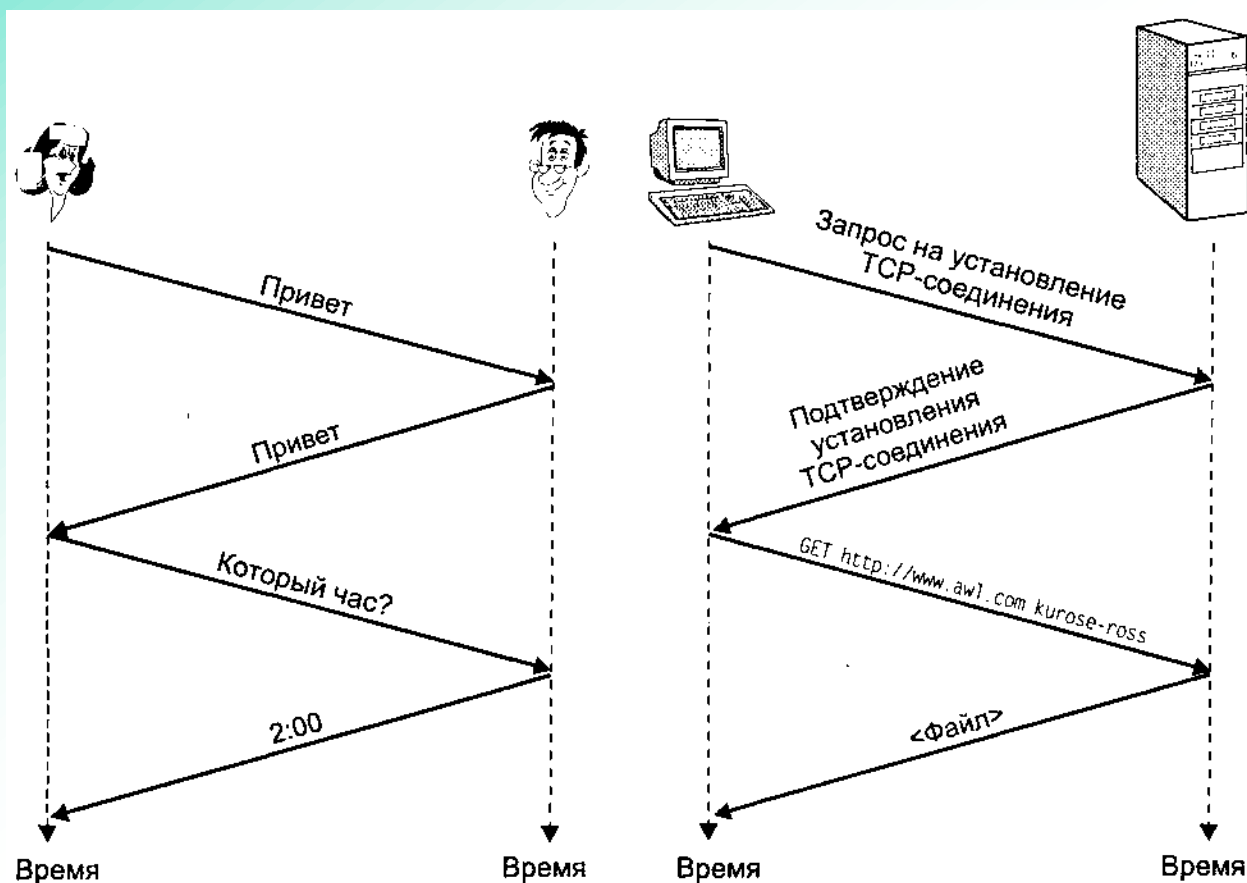
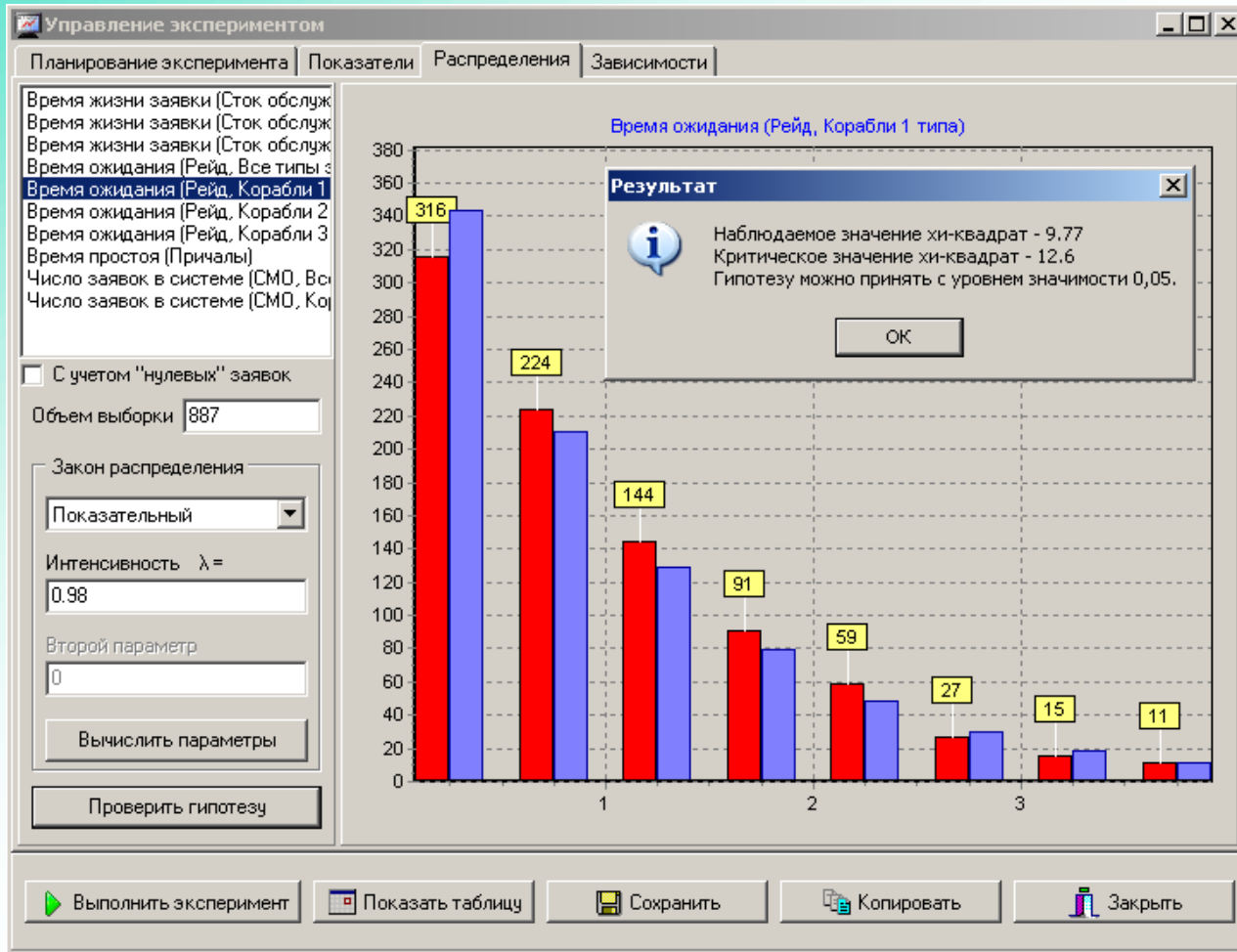


Рис. 1.2. Протоколы общения между людьми и между компьютерами

Однако, делая выводы на основании аналогии нужно знать, насколько эти выводы достоверны. Модель никогда полностью не тождественна оригиналу, между ними всегда есть определенные различия. Поэтому выводы на основании аналогии всегда имеют не абсолютно достоверный, а предположительный характер.

Гипотеза - предположение, основывающееся на догадке,

наблюдении, опыте и требующее подтверждения или доказательства. В процессе познания каждая гипотеза подвергается проверке. С накоплением новых фактов одна гипотеза может быть заменена другой.



При этом часто старая гипотеза не отбрасывается полностью, а только исправляется и уточняется. В результате исправленная и уточненная гипотеза превращается в теорию или закон.

Краеугольным камнем метода моделирования является теория подобия, устанавливающая те условия, при которых модель действительно отражает (в том или ином смысле) оригинал.

Подобие явлений, характеризующееся соответствием (например, пропорциональностью) величин, участвующих в изучаемых явлениях, происходящих в оригинале и в моделях, по степени соответствия параметров модели и оригинала может быть трех видов:

- 1. полное подобие,*
- 2. неполное подобие,*
- 3. приближенное подобие.*

Неполное подобие, в отличие от **полного подобия**, связано с изучением процессов только во времени или только в пространстве.

Приближенное подобие реализуется при некоторых упрощающих допущениях, что, как правило, приводит к искажениям, которые должны заранее оцениваться количественно.

Все виды подобия подчиняются общим закономерностям, описываемым теоремами о подобии и некоторыми дополнительными положениями.

Помимо соблюдения условий подобия **эффективность моделирования** зависит от того, в какой степени модель согласована с культурной средой, в которой ей предстоит функционировать. Она должна органически вписываться в эту среду, а не представлять собой чуждый ей элемент. Еще одним аспектом согласованности является обеспеченность модели ресурсами.

Основные различия между моделью и действительностью

связаны с конечностью, упрощенностью и приближенностью моделей.

Основное противоречие познавательных моделей состоит в

необходимости использования для познания бесконечного мира ограниченных средств. Поэтому модель подобна оригиналу лишь в конечном числе отношений.

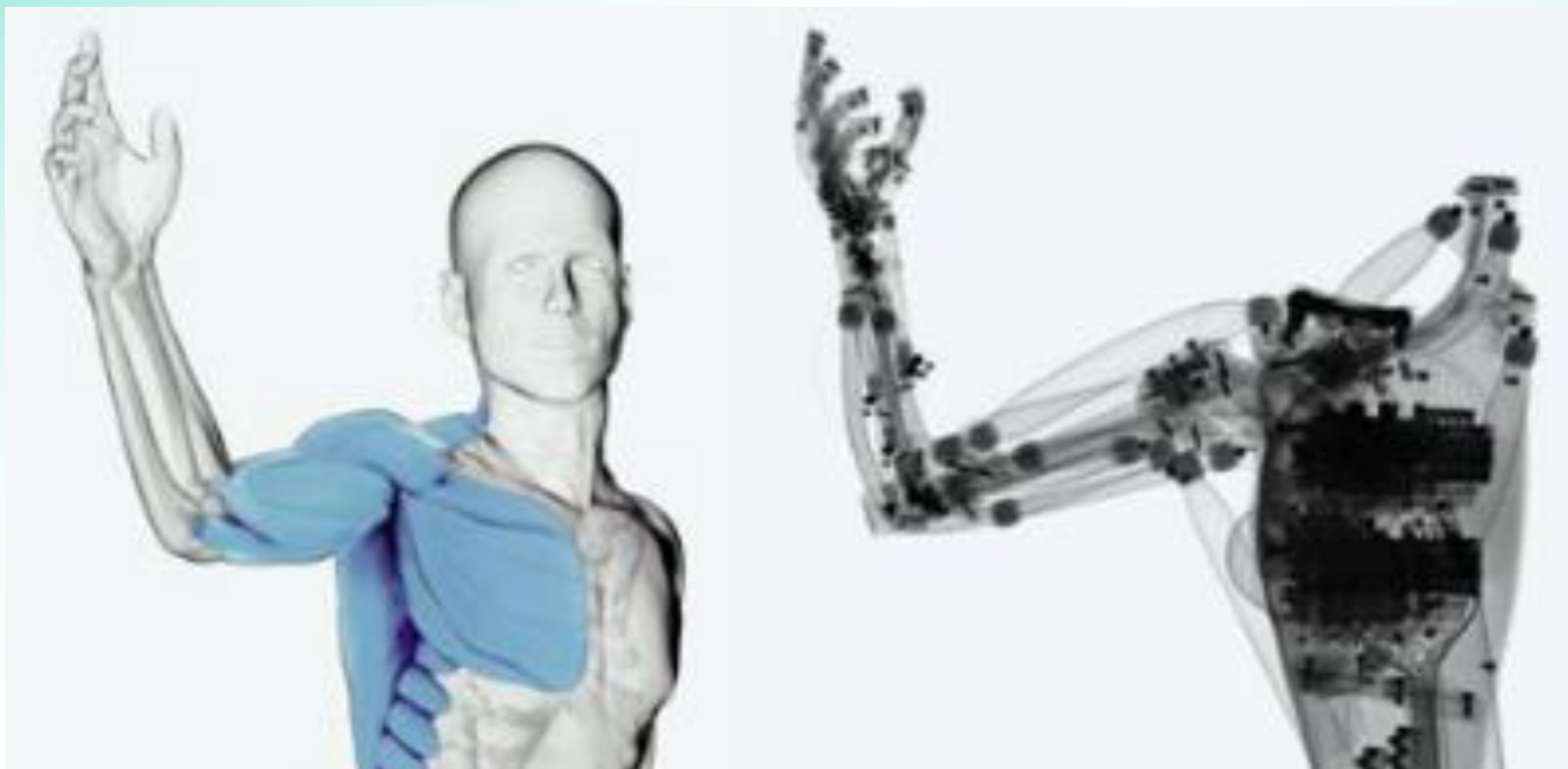
Познавательные возможности модели обуславливаются тем, что

модель отображает лишь наиболее существенные черты объекта-оригинала. Вопрос о необходимой и достаточной мере сходства оригинала и модели и об адекватности модели требует особого анализа.

Модель, с помощью которой успешно достигается поставленная цель, **называется адекватной этой цели.**

Сходство модели и действительности проверяется на практике.

При моделировании, как и в других методах исследования, решающим критерием истинности является практика. Помимо безусловно истинного, в модели есть, как правило, и верное лишь при определенных условиях, и даже неправильное, т.е. не имеющее отношение к оригиналу.



Виды моделей

По сфере применения модели разделяются на: *экономические, социальные, биологические, производственно-технологические, научно-исследовательские* и др.

В каждой области знаний существует своя классификация моделей. Ввиду обширности этого вопроса рассмотрим лишь некоторые общие признаки, по которым эту классификацию можно производить.

С точки зрения целей создания модели делятся на *познавательные и прагматические*.

Познавательные модели создаются на основании изучения и для изучения уже существующих объектов. Они являются формой организации и представления знаний, средством соединения новых знаний с имеющимися. В процессе их создания познавательные модели приближаются к реальности, которую они отображают.

Прагматические модели служат для организации практических действий, на основе сформулированной ранее «идеальной» модели. В этом случае выявляемые расхождения между реальностью и моделью устраняются путем «подгонки» реальности под модель.

Примерами таких процессов могут служить: техническое проектирование, экономическое реформирование. При проведении экономических реформ реальная жизнь подстраивается под некие схемы (модели), внедрение которых, по мнению их авторов, должно улучшить социальные и экономические показатели.

Таким образом, имеется **два основных отличия** рассмотренных выше типов моделей.

Во-первых, познавательные модели строятся на основании отображаемых объектов, в то время как построение прагматических моделей предшествует созданию реальных объектов.

Во-вторых, познавательные модели отображают существующее, а прагматические - не существующее (а лишь желаемое).

В зависимости от используемых средств модели делятся на абстрактные или идеальные и материальные, т.е. реальные или вещественные.



Абстрактные модели являются идеальными конструкциями, построенными средствами мышления, сознания.

Они бывают двух типов: ***мысленные, умозрительные или интуитивные*** модели, с одной стороны, и ***логико-математические***, с другой.

Класс абстрактных моделей образуют ***вербальные*** (словесные), ***знаковые*** и ***математические*** модели.

Вербальные модели используют **естественные** и **специальные** языки.

Примерами моделей, основанных на использовании **естественных языков**, являются различные инструкции, планы действий и т.д.

Для построения таких моделей не требуется специальной подготовки, но точность таких моделей не велика. На практике их приблизительность, расплывчатость преодолевается с помощью «правильного» понимания (интерпретации).

Дифференциация наук потребовала создания **специализированных языков**, более четких и точных, чем естественный язык.

Применение специальных языков требует дополнительных затрат на их разработку и изучение, но они многократно окупаются за счет преимуществ использования компактных, ясных и точных моделей.

Знаковые модели изучаются в семиотике (науке о знаках).

В семиотике выделены три основные группы или аспекта отношений:

синтаксис (греч. «построение», «порядок») - отношение между знаками;

семантика (греч. «обозначение») - отношение между знаками и тем, что они обозначают;

прагматика (греч. «дело», «действие») - отношение между знаками и результатом их использования.

Знаковые модели особенно широко применяются в системах искусственного интеллекта (СИИ).

Математическая модель есть описание какого-либо класса явлений внешнего мира, выраженное с помощью математической символики.

Класс математических моделей очень широк, для их изучения необходимо знание соответствующих разделов математики, поэтому ограничимся лишь перечислением некоторых из них.

К математическим моделям относятся: *алгебраические, логико-алгебраические, графовые.*

При построении математических моделей процессов функционирования систем существуют следующие основные подходы:

1. непрерывно-детерминированный (например, дифференциальные уравнения, уравнения состояния);
2. дискретно-детерминированный (конечные автоматы);
3. дискретно-стохастический (вероятностные автоматы);
4. непрерывно-стохастический (системы массового обслуживания);
5. обобщенный или универсальный (агрегативные системы).

Классификация моделей и видов моделирования объектов и систем в соответствии с теорией подобия должна выделить в них наиболее общие признаки и свойства реальных систем.

Признаки классификации	Виды математических моделей
1. Принадлежность к иерархическому уровню	<ul style="list-style-type: none"> ▪ <u>микроуровня</u> Модели ▪ <u>макроуровня</u> Модели ▪ Модели <u>метауровня</u>
2. Характер взаимоотношений со средой	<ul style="list-style-type: none"> ▪ <u>Открытые (непрерывный обмен)</u> ▪ <u>Закрытые (слабая связь)</u>
3. Характер отображаемых свойств объекта	<ul style="list-style-type: none"> ▪ Структурные ▪ Функциональные
4. Способ представления свойств объекта	<ul style="list-style-type: none"> ▪ Аналитические ▪ Алгоритмические ▪ Имитационные

5. Способ получения модели	<ul style="list-style-type: none">▪▪	Теоретические Эмпирические
6. Причинная обусловленность	<ul style="list-style-type: none">▪▪	Детерминированные Вероятностные
7. По отношению к времени	<ul style="list-style-type: none">▪▪	Динамические Статические

8. По типу уравнений	<ul style="list-style-type: none">▪ Линейные▪ Нелинейные
9. По множеству значений переменных	<ul style="list-style-type: none">▪ Непрерывные▪ Дискретные▪ Дискретно- непрерывные
10. По назначению	<ul style="list-style-type: none">▪ Технические▪ Экономические▪ Социальные и т.д.

Математический подход к моделированию имеет ряд

недостатков:

1. низкая адекватность математической модели реальному объекту;
2. проблемы, связанные с решаемостью математических моделей из-за наличия в них разрывных функций;
3. непригодность математических моделей для большинства объектов с переменной структурой;
4. приближенные методы реализаций моделей с переменными коэффициентами требуют значительных затрат и не обладают достаточной точностью решения.

Операции над моделями

1. *Линеаризация*

Пусть $M = M(X, Y, A)$, где X - множество входов, Y - выходов, A - состояний системы. Схематически можно это изобразить: $X \Rightarrow A \Rightarrow Y$

Если X, Y, A - линейные пространства (множества), а f, y - линейные операторы, то система (модель) называется линейной. Другие системы (модели) - нелинейные. Нелинейные системы трудно поддаются исследованию, поэтому их часто линеаризуют - сводят к линейным каким-то образом.

2. Идентификация

Пусть $M=M(X,Y,A)$, $A=\{a_i\}$, $a_i=(a_{i1},a_{i2},\dots,a_{ik})$ - вектор состояния объекта (системы). Если вектор a_i зависит от некоторых неизвестных параметров, то задача идентификации (модели, параметров модели) состоит в определении по некоторым дополнительным условиям, например, экспериментальным данным, характеризующим состояние системы в некоторых случаях. Идентификация - решение задачи построения по результатам наблюдений математических моделей, описывающих адекватно поведение реальной системы.

3. Агрегирование

Операция состоит в преобразовании (сведении) модели к модели (моделям) меньшей размерности (X, Y, A) .

4. Декомпозиция

Операция состоит в разделении системы (модели) на подсистемы (подмодели) с сохранением структур и принадлежности одних элементов и подсистем другим.

5. Сборка

Операция состоит в преобразовании системы, модели, реализующей поставленную цель из заданных или определяемых подмоделей (структурно связанных и устойчивых).

6. Макетирование

Эта операция состоит в апробации, исследовании структурной связности, сложности, устойчивости с помощью макетов или подмоделей упрощенного вида, у которых функциональная часть упрощена (хотя вход и выход подмоделей сохранены).

7. Экспертиза, экспертное оценивание

Операция или процедура использования опыта, знаний, интуиции, интеллекта экспертов для исследования или моделирования плохо структурируемых, плохо формализуемых подсистем исследуемой системы.

8. Вычислительный эксперимент

Это эксперимент, осуществляемый с помощью модели на ЭВМ с целью распределения, прогноза тех или иных состояний системы, реакции на те или иные входные сигналы. Прибором эксперимента здесь является компьютер (и модель!).

Материальные модели представляют собой некоторые материальные объекты, отражающие в той или иной мере свойства моделируемых объектов моделирования.

По способу отражения этих свойств различают *физические* и *предметно-математические* модели.

Физические модели имеют такую же физическую природу, как оригинал и отличаются в основном размерами (модель дамбы, модель самолета).

Необходимо учитывать, что для получения адекватной физической модели не достаточно лишь пропорционально изменить размеры объекта. Необходимо доказать, что натура и модель подобны, подчиняются одним и тем же физическим законам и описываются одинаковыми математическими зависимостями.

Частным случаем физических моделей являются *геометрические* модели, которые отличаются от оригинальных объектов масштабом исполнения (макет здания).

К недостаткам физических моделей относится относительно высокая трудоемкость их изготовления, сложность изменения параметров моделей при переходе от одного варианта к другому.

Аналогии между лабораторным и вычислительным экспериментами

Лабораторный эксперимент	Вычислительный эксперимент
Образец	Модель
Физический прибор	Программа для компьютера
Калибровка прибора	Тестирование программы
Измерение	Расчет
Анализ данных	Анализ данных

Предметно-математические модели основаны на совпадении математических выражений, которыми описываются процессы в оригинале и в модели.

Для построения этих моделей могут использоваться **методы прямой аналогии** (например, гидравлическая модель для изучения межотраслевого баланса, электрическая модель для изучения циркуляции денежных потоков) и **машинное моделирование**.

Метод прямой аналогии (МПА) основан на одинаковости математического описания многих электрических, механических, тепловых, гидродинамических и других явлений (например, закон Ома для электрической цепи и закон Фурье для теплового потока).

МПА имитирует физическую систему по ее элементам таким образом, что каждому физическому элементу природы соответствует определенный эквивалент в модели. В МПА, как и в физических моделях, происходит расчленение исследуемой системы на составляющие ее физические элементы.

Метод машинного моделирования использует другой подход. Его применение связано с расчленением математического описания исследуемой системы на отдельные математические операции. При этом могут использоваться различные классы вычислительных машин: *аналоговые (АВМ), цифровые (ЦВМ) и аналого-цифровые (АЦВМ).*

Метод полунатурного моделирования (ПНМ) состоит в том, что на определенном этапе исследования одна часть системы (например, объект управления) замещается моделью, а другая часть (устройство управления) остается натурной. В процессе разработки системы модели отдельных устройств могут заменяться реальными блоками.

Кибернетические модели

Известно, что первые успехи кибернетики связаны с развитием систем автоматического регулирования (САР). В это время сформировались многие фундаментальные понятия, такие, как: *обратная связь, структурно-графовые модели систем, частотные методы расчета и проектирования систем обработки информации.*

Большое влияние на кибернетику оказала *статистическая физика*, вероятностные методы которой были перенесены и развиты в *теории массового обслуживания и надежности, теории игр* и в *теории распознавания образов*. Эти модели образуют **класс вероятностных моделей**.

Модели дискретной математики содержат модели в виде *графов, автоматов, лингвистических и логических структур.*

По степени универсальности кибернетические модели условно делятся на три типа:

- 1. модели, не зависящие от предметной области (ПО);*
- 2. модели ориентированные на ПО;*
- 3. модели, настраиваемые на ПО.*

Модели не зависящие от ПО так называемые, автономные или синтаксические модели, являются наиболее универсальными. Примерами таких моделей могут служить модели массового обслуживания.

Модели систем массового обслуживания (СМО) являются одной из разновидностей вероятностных моделей и характеризуются случайным законом изменения входных воздействий и параметров системы. Они зависят от типа этого закона и структуры системы, но никак не связаны с природой моделируемого объекта (например, управление пассажиропотоками, обслуживание клиентов, передача пакетов по каналам связи).

Модели ориентированные на ПО применяются в тех случаях, когда реальные системы обладают только им присущей индивидуальностью. В этом случае необходимо, чтобы в модели присутствовала ПО. Кроме того, это связано еще с требованием семантического (логического или смыслового) и прагматического (целевого) управления. Задачи семантической и прагматической обработки информации не могут быть решены без ориентации на предметную область.

Примером такой модели может служить **ситуационная модель** управления сложной системой (типа аэропорт Пулково). При ситуационном управлении в результате сбора, накопления и формализации знаний об объекте управления и способах управления им вырабатываются правила типа «ситуация - решение по управлению».

Настраиваемые модели занимают промежуточное положение между независимыми и ориентированными на ПО моделями.

Примерами таких моделей могут служить *оболочки экспертных систем*. С помощью таких оболочек можно решать определенные задачи (диагностики, мониторинга, проектирования и т.д.) в различных предметных областях. Перед их применением с помощью специалистов-экспертов остается заполнить базу знаний.

База знаний - семантическая модель, предназначенная для представления в ЭВМ знаний накопленных людьми в определенной предметной области.

Имитационное моделирование используется в тех случаях, когда аналитические методы оказываются малопригодными из-за большой размерности и сложности решаемых задач. Проблема заключается в том, что многочисленные допущения, которые вынужден делать исследователь в этом случае для упрощения задачи, делают аналитическую модель неадекватной реальной системе. Имитационная модель позволяет более полно учитывать особенности реальной системы и в то же время обладает достаточной точностью.

Процесс создания имитационной модели, как правило, включает три стадии:

1. составление вербальной модели системы с помощью естественного и/или специальных языков;
2. составление формализованной модели на основании словесного описания;
3. реализация модели на ЭВМ с помощью специальных программных средств и/или языков моделирования

Модели систем искусственного интеллекта (СИИ) возникли в связи с желанием и необходимостью повысить «интеллектуальность» разрабатываемых систем управления, наделить машины способностью принимать решения аналогично тому, как это делает человек.

В настоящее время в психологии мышления существуют *несколько моделей творческой деятельности.*

В лабиринтной модели переход от постановки задачи к решению проходит через лабиринт возможных альтернативных путей. Но не всякая творческая задача сводится к такой схеме, кроме того, возможностей может быть слишком много и их не в состоянии за приемлемое время перебрать даже самая быстрая ЭВМ. В этом случае кроме (или вместо) процедуры целенаправленного поиска требуется, использовать другие процедуры.

Ассоциативная модель мышления в качестве основной процедуры использует ассоциативный поиск и ассоциативное рассуждение. Ассоциативное рассуждение позволяет переносить приемы, уже использованные ранее, на текущую ситуацию. Ассоциативные модели в целом не получили широкого распространения, но успешно используются в задачах распознавания образов, в классификационных задачах, для самообучения ЭВМ.

Модельная гипотеза опирается на идею внутреннего представления человеком проблемной области, на его знания о ее особенностях, закономерностях и процедурах действия в ней. В модельной гипотезе основными процедурами являются: *представление знаний, моделирование рассуждения, поиск релевантной информации* и некоторые другие. Эти процедуры в совокупности с процедурами целенаправленного и ассоциативного поиска составляют основу СИИ.

Модели представления знаний



Первый подход, называемый **эмпирическим**, основан на изучении принципов организации человеческой памяти и моделировании механизмов решения задач человеком. На основе этого подхода в настоящее время разработаны и получили наибольшую известность следующие модели:

1. **продукционные модели** – модель основанная на правилах, позволяет представить знание в виде предложений типа: «ЕСЛИ условие, ТО действие». Продукционная модель обладает тем недостатком, что при накоплении достаточно большого числа (порядка нескольких сотен) продукций они начинают противоречить друг другу;

2. **сетевые модели (или семантические сети)** – в инженерии знаний под ней подразумевается граф, отображающий смысл целостного образа. Узлы графа соответствуют понятиям и объектам, а дуги – отношениям между объектами. Обладает тем недостатком, что однозначного определения семантической сети в настоящее время отсутствует;

3. фреймовая модель – основывается на таком понятии как фрейм (англ. frame – рамка, каркас). Фрейм – структура данных для представления некоторого концептуального объекта. Информация, относящаяся к фрейму, содержится в составляющих его слотах. Слоты могут быть терминальными либо являться сами фреймами, т.о. образуя целую иерархическую сеть.

Условно в группу эмпирического подхода можно включить **нейронные сети и генетические алгоритмы**, относящиеся к бионическому (основано на предположении о том, что если в искусственной системе воспроизвести структуры и процессы человеческого мозга, то и результаты решения задач такой системой будут подобны результатам, получаемым человеком) направлению искусственного интеллекта. Особенностью моделей этого типа является широкое использование эвристик, что в каждом случае требует доказательства правильности получаемых решений.

Второй подход можно определить как **теоретически обоснованный**, гарантирующий правильность решений. Он в основном представлен моделями, основанными на формальной логике (исчисление высказываний, исчисление предикатов), формальных грамматиках, комбинаторными моделями, в частности моделями конечных проективных геометрий, теории графов, тензорными и алгебраическими моделями. В рамках этого подхода до настоящего времени удавалось решать только сравнительно простые задачи из узкой предметной области.

Процедурные знания описывают последовательность действий при решении задач (алгоритмы, программы, инструкции и т.д.).

Декларативные знания имеют в основном повествовательный характер (например, «знание информатики - необходимое условие успешного изучения специальных дисциплин»).

На первых этапах автоматизации обработки данных главную роль играли процедурные знания, доминировали языки программирования процедурного типа. В настоящее время по мере повышения уровня интеллектуальности ЭВМ все большую роль при обработке информации играют декларативные знания.

К языкам процедурного типа относятся такие, ставшие традиционными языки, как: Basic, Pascal, Си++.

Языки декларативного типа пока еще не получили столь широкого распространения. В зависимости от используемого математического аппарата они подразделяются на: *функциональные, логические, реляционные*. Рассмотрим некоторые примеры.

Функциональный язык обработки символьной информации основывается на алгебре списочных структур, лямбда-исчислении, теории рекурсивных функций.

Первым, спроектированным функциональным языком стал [Лисп](#). Он был предложен Джоном Мак-Карти в качестве средства исследования границ применимости компьютеров, в частности, методом решения задач искусственного интеллекта. Лисп послужил эффективным инструментом экспериментальной поддержки теории программирования и развития сферы его применения. Вариант данного языка широко используется в системе автоматизированного проектирования [AutoCAD](#) и называется [AutoLISP](#).

В качестве основных свойств функциональных языков программирования обычно рассматриваются следующие:
краткость и простота;

Программы на функциональных языках обычно намного короче и проще, чем те же самые программы на императивных языках.

Язык аналогичного типа - РЕФАЛ (Рекурсивных Функций Алгоритмический язык) ориентирован на решение задач, использующих нормальные алгоритмы Маркова.

Язык логического программирования Пролог и язык реляционного типа ФРЛ рассмотрены далее при описании соответствующих моделей представления знаний.

Модели формального представления знаний в ЭВМ относятся к одному из трех типов: *логические, сетевые, продукционные*.

Каждая модель представления знаний имеет свои преимущества и недостатки, поэтому разработаны **комбинированные языки представления знаний** (в них чаще всего используются фреймовые и продукционные модели).

Компьютерное моделирование



После прохождения этих этапов наиболее полно могут быть выполнены требования, предъявляемые к моделям:

Универсальность — характеризует полноту отображения моделью изучаемых свойств реального объекта;

Адекватность — способность отражать нужные свойства объекта с погрешностью не выше допустимой;

Точность — оценивается степенью совпадения значений характеристик реального объекта со значениями этих характеристик, полученных с помощью моделей;

Экономичность — определяется затратами ресурсов ЭВМ (памяти и времени на ее реализацию и эксплуатацию).

Качество моделирования может быть оценено характеристикой

его потребительских свойств:

- эффективность использования его по назначению (цели);
- ресурсоемкость;
- стоимость.

КАЧЕСТВО МОДЕЛИРОВАНИЯ

ЭФФЕКТИВНОСТЬ

Корректность и
точность

Понятность
алгоритма

Сложность
алгоритма

Уровень сложности
алгоритмического
языка

Устойчивость
результатов

СТОИМОСТЬ

Затраты на
разработку

Затраты на
приобретение
ресурсов

Затраты на
сопровождение и
эксплуатацию

Другие
виды
затрат

РЕСУРСОЕМКОСТЬ

Требуемый объем памяти

Количество оборудования

Объем требуемого
информационного
обеспечения

Количество личного
состава

Объем выходных
данных

Сложные системы и декомпозиция

Системный анализ родился как метод исследования и проектирования сложных систем.

Что же такое "сложная" система?

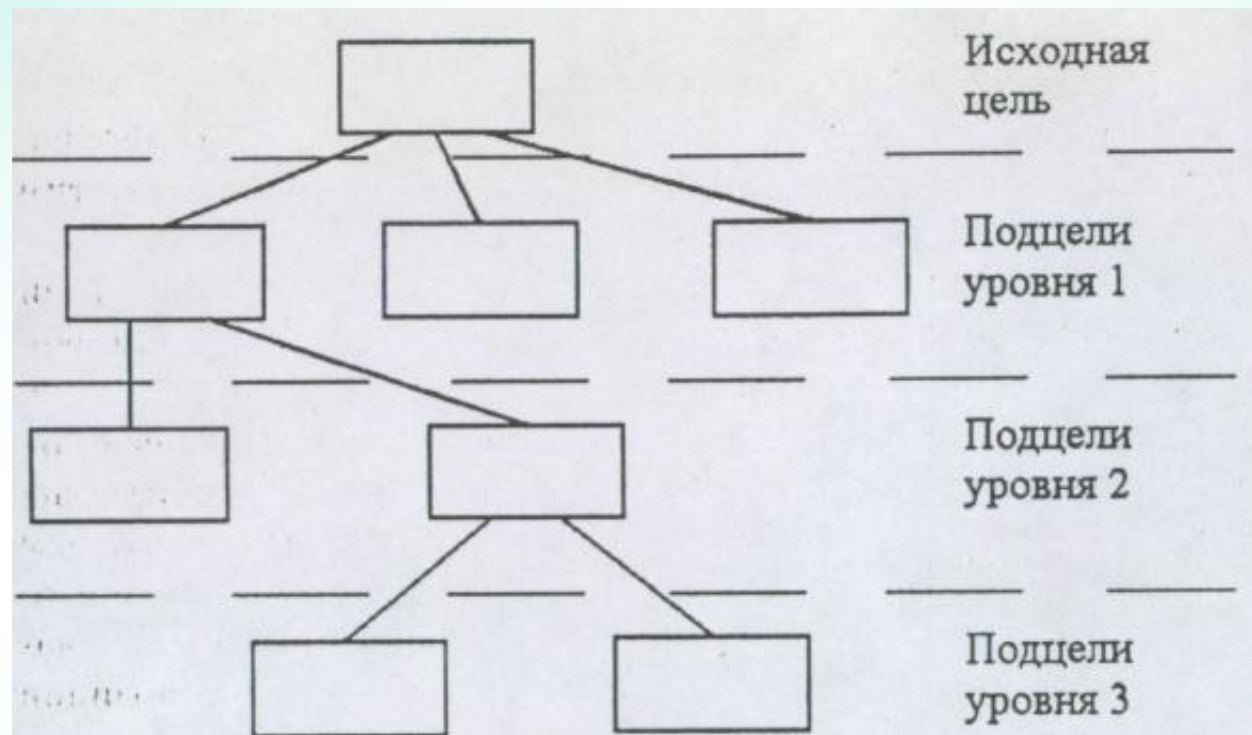
Понятие это неформальное, и обычно, говоря о сложных системах, перечисляют их основные особенности:

1. наличие большого числа разнородных элементов (подсистем);
2. сложный характер, неоднородность связей между подсистемами;
3. сложность функций, выполняемых системой;
4. наличие неопределенности в описании системы;
5. сложность определения (организации) требуемого управляющего воздействия на систему и т.д.

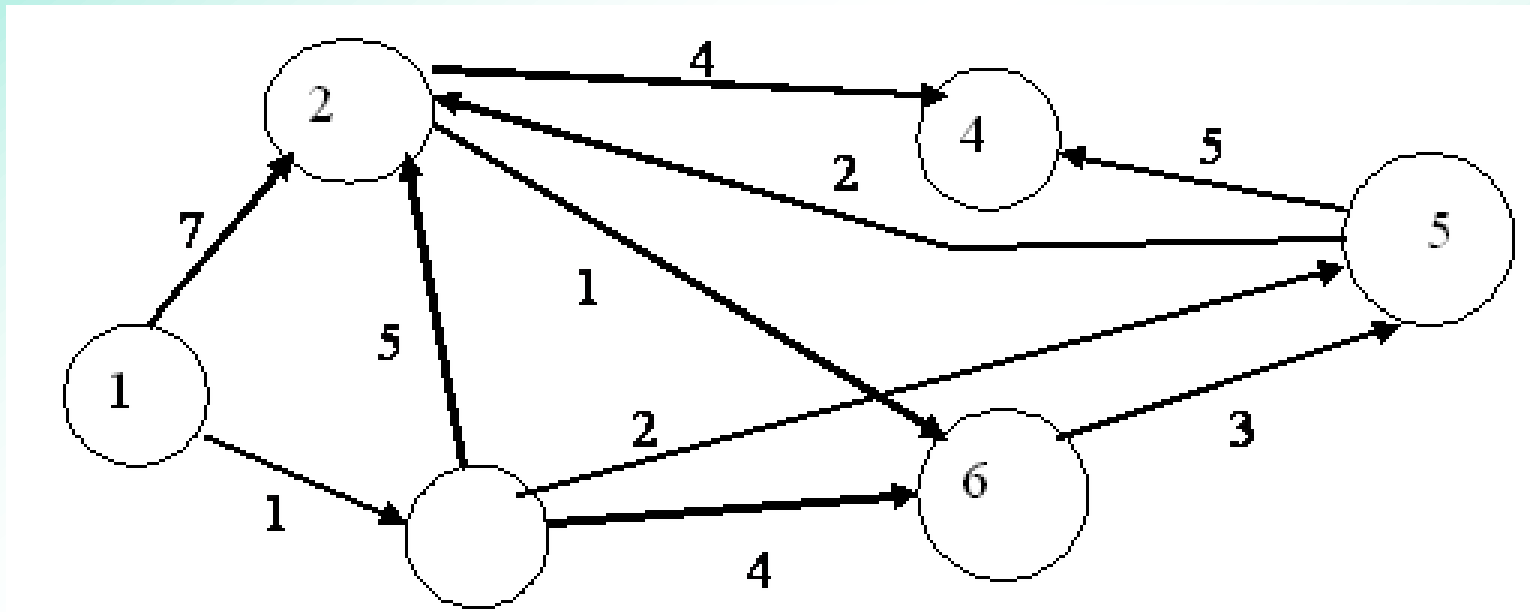
Каждая из этих особенностей может оказаться существенной или несущественной: все зависит от конкретной ситуации и целей исследования. Поэтому более универсальный способ выделения класса сложных систем связан со сложностью самого процесса исследования системы. Если методика математического моделирования приводит к успеху сразу же, "за один проход", то нет оснований называть систему "сложной".

Введение этого термина оправдано, если решить задачу в исходном виде не удастся. В этом случае она разбивается на несколько вспомогательных подзадач, решаемых по отдельности. Такой прием называется **декомпозицией** и является основным методом исследования сложных систем.

При декомпозиции исходная система делится на подсистемы, а цель - на подцели. Далее для решения каждой подзадачи пользуются той же методикой, что и для всей системы. Если в ходе решения (а возможно, и до того) какие-то из подзадач окажутся слишком сложными, то снова проводится декомпозиция: возникают подзадачи следующего уровня и т.д. Результатом этого процесса является структуризация: исходная система приобретает иерархическую (многоуровневую) структуру.

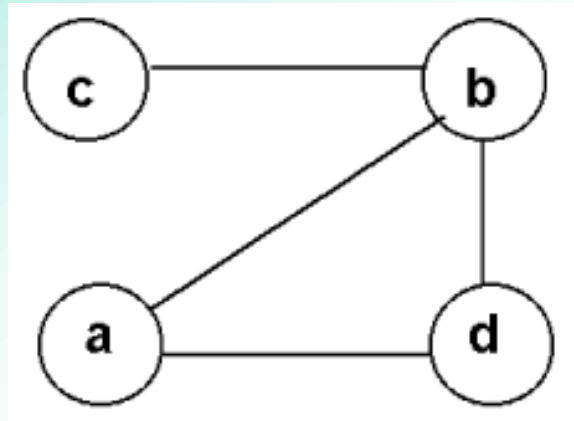


Каждой сложной системе ставится в соответствие граф (*структурный*), вершинами которого являются подсистемы, а дугами - имеющиеся между ними связи. Если связи направленные, то граф системы является **ориентированным** (*направленным*). К этому классу относятся, например, структурные схемы (*граф-схемы*) систем автоматического управления.



Модели на основе направленных графов (модели программ VisSim, Simulink, MBTY).

У других систем влияние связанных подсистем обоюдно и они описываются неориентированными графами (например, сложные электрические и электронные схемы).



Модели на основе ненаправленных графов (модели программы Electronics Workbench).

По виду направленного графа:

1. Модели с последовательным графом (ПФ разложена на множители).
2. Модели с параллельным графом (ПФ разложена на элементарные дроби).
3. Модели на основе одного из двух универсальных графов, которые соответствуют стандартной форме записи передаточной функции.
4. Модели с графами, специфика которых учитывает эффект квантования параметров.
5. Модели с матричными графами (ABCD-граф или граф для решения уравнений в форме Коши).

В сложных системах часто приходится проводить несколько вариантов декомпозиции и соответственно строить несколько деревьев целей. Это - обычно связано с наличием нескольких критериев функционирования системы. Возникающие при этом задачи многокритериального выбора изучаются в теории принятия решений. Успех декомпозиции часто определяется интуицией и опытом исследователя. Человек, по данным психологов, может мысленным взором охватить структуру декомпозированной системы, если на каждом уровне возникает не более чем 5 ± 2 подзадачи.

Классификация видов моделирования систем



Тема 2

Этапы моделирования систем

Требования пользователя к модели. Сформулируем

основные требования, предъявляемые к модели M процесса функционирования системы S .

1. Полнота модели должна предоставлять пользователю возможность получения необходимого набора оценок характеристик системы с требуемой точностью и достоверностью.

2. Гибкость модели должна давать возможность воспроизведения различных ситуаций при варьировании структуры, алгоритмов и параметров системы.

3. Длительность разработки и реализации модели большой системы должна быть по возможности минимальной при учете ограничений на имеющиеся ресурсы.

4. Структура модели должна быть блочной, т. е. допускать возможность замены, добавления и исключения некоторых частей без переделки всей модели.

5. Информационное обеспечение должно предоставлять возможность эффективной работы модели с базой данных систем определенного класса.

6. Программные и технические средства должны обеспечивать эффективную (по быстродействию и памяти) машинную реализацию модели и удобное общение с ней пользователя.

7. Должно быть реализовано проведение целенаправленных (планируемых) машинных экспериментов с моделью системы с использованием аналитико-имитационного подхода при наличии ограниченных вычислительных ресурсов.

С учетом этих требований рассмотрим основные положения, которые справедливы при моделировании на ЭВМ систем **S**, а также их подсистем и элементов. При машинном моделировании системы **S** характеристики процесса ее функционирования определяются на основе модели **M**, построенной исходя из имеющейся исходной информации об объекте моделирования.

При получении новой информации об объекте его модель пересматривается и уточняется с учетом новой информации, т. е. процесс моделирования, включая разработку и машинную реализацию модели, является итерационным. Этот итерационный процесс продолжается до тех пор, пока не будет получена модель ***M***, которую можно считать адекватной в рамках решения поставленной задачи исследования и проектирования системы ***S***.

Моделирование систем с помощью ЭВМ можно использовать в следующих случаях: а) для исследования системы **S** до того, как она спроектирована, с целью определения чувствительности характеристики к изменениям структуры, алгоритмов и параметров объекта моделирования и внешней среды; б) на этапе проектирования системы **S** для анализа и синтеза различных вариантов системы и выбора среди конкурирующих такого варианта, который удовлетворял бы заданному критерию оценки эффективности системы при принятых ограничениях; в) после завершения проектирования и внедрения системы, т. е. при ее эксплуатации, для получения информации, дополняющей результаты натуральных испытаний (эксплуатации) реальной системы, и для получения прогнозов эволюции (развития) системы во времени.

Существуют общие положения, применяемые ко всем перечисленным случаям машинного моделирования. Даже в тех случаях, когда конкретные способы моделирования отличаются друг от друга и имеются различные модификации моделей, например в области машинной реализации моделирующих алгоритмов с использованием конкретных программно-технических средств, в практике моделирования систем можно сформулировать общие принципы, которые могут быть положены в основу методологии машинного моделирования.

Этапы моделирования систем. Рассмотрим основные этапы моделирования системы S , к числу которых относятся: построение концептуальной модели системы и ее формализация; алгоритмизация модели системы и ее машинная реализация; получение и интерпретация результатов моделирования системы.

Взаимосвязь перечисленных этапов моделирования систем и их составляющих (подэтапов) может быть представлена в виде сетевого графика, показанного на рис. 1. Перечислим эти подэтапы: 1.1 - постановка задачи машинного моделирования системы; 1.2 - анализ задачи моделирования системы; 1.3 - определение требований к исходной информации об объекте моделирования и организация ее сбора; 1.4 - выдвижение гипотез и принятие предположений; 1.5 - определение параметров и переменных модели; 1.6 - установление основного содержания модели; 1.7 - обоснование критериев оценки эффективности системы; 1.8 - определение процедур аппроксимации; 1.9 - описание концептуальной модели системы; 1.10 - проверка достоверности концептуальной модели; 1.11 - составление технической документации по первому этапу; 2.1 - построение логической схемы модели; 2.2 - получение математических соотношений; 2.3 - проверка достоверности модели системы; 2.4 - выбор инструментальных средств для моделирования; 2.5 - составление плана выполнения работ по программированию; 2.6 - спецификация и построение схемы программы; 2.7 - верификация и проверка достоверности схемы программы; 2.8 - проведение программирования модели; 2.9 - проверка достоверности программы; 2.10 - составление технической документации по второму этапу;

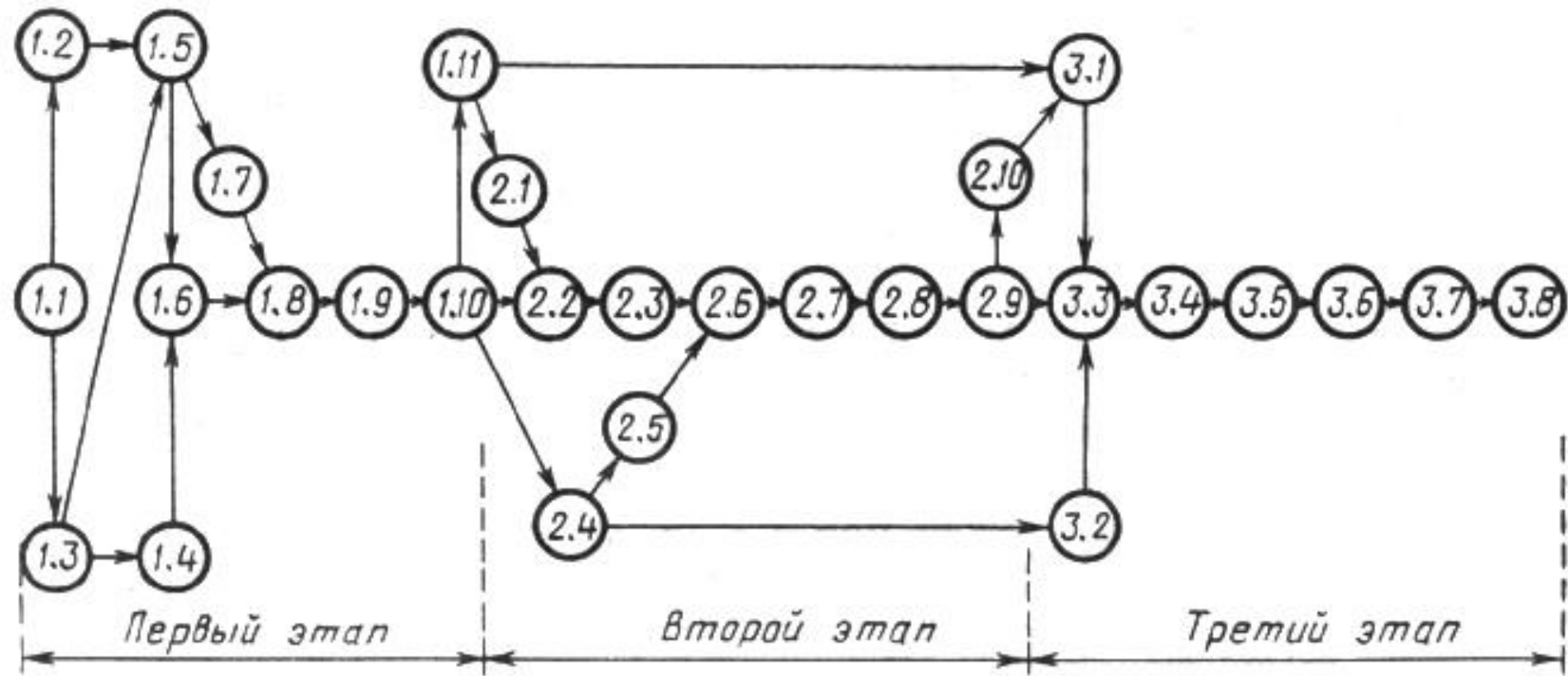


Рис. 1. Взаимосвязь этапов моделирования систем

3.1 - планирование машинного эксперимента с моделью системы; 3.2 - определение требований к вычислительным средствам; 3.3 - проведение рабочих расчетов; 3.4 - анализ результатов моделирования системы; 3.5 - представление результатов моделирования; 3.6 - интерпретация результатов моделирования; 3.7 - подведение итогов моделирования и выдача рекомендаций; 3.8 - составление технической документации по третьему этапу.

Таким образом, процесс моделирования системы S сводится к выполнению перечисленных подэтапов, сгруппированных в виде трех этапов. На этапе построения концептуальной модели M_k и ее формализации проводится исследование моделируемого объекта с точки зрения выделения основных составляющих процесса его функционирования, определяются необходимые аппроксимации и получается обобщенная схема модели системы S , которая преобразуется в машинную модель M_k на втором этапе моделирования путем последовательной алгоритмизации и программирования модели. Последний третий этап моделирования системы сводится к проведению согласно полученному плану рабочих расчетов на ЭВМ с использованием выбранных программно-технических средств, получению и интерпретации результатов моделирования системы S с учетом воздействия внешней среды E . Очевидно, что при построении модели и ее машинной реализации при получении новой информации возможен пересмотр ранее принятых решений, т. е. процесс моделирования является итерационным. Рассмотрим содержание каждого из этапов более подробно.

На первом этапе машинного моделирования - построения *концептуальной модели* M_k системы S и ее формализации - формулируется модель и строится ее формальная схема, т. е. основным назначением этого этапа является переход от содержательного описания объекта к его математической модели, другими словами, процесс формализации. Моделирование систем на ЭВМ в настоящее время - наиболее универсальный и эффективный метод оценки характеристик больших систем. Наиболее ответственными и наименее формализованными моментами в этой работе являются проведение границы между системой S и внешней средой E , упрощение описания системы и построение сначала концептуальной, а затем формальной модели системы. Модель должна быть адекватной, иначе невозможно получить положительные результаты моделирования, т. е. исследование процесса функционирования системы на неадекватной модели вообще теряет смысл. Под адекватной моделью будем понимать модель, которая с определенной степенью приближения на уровне понимания моделируемой системы S разработчиком модели отражает процесс ее функционирования во внешней среде E .

Переход от описания к блочной модели. Наиболее рационально строить модель функционирования системы по блочному принципу. При этом могут быть выделены три автономные группы блоков такой модели. Блоки первой группы представляют собой имитатор воздействий внешней среды **E** на систему **S**; блоки второй группы являются собственно моделью процесса функционирования исследуемой системы **S**; блоки третьей группы - вспомогательными и служат для машинной реализации блоков двух первых групп, а также для фиксации и обработки результатов моделирования.

Рассмотрим механизм перехода от описания процесса функционирования некоторой гипотетической системы к модели этого процесса. Для наглядности введем представление об описании свойств процесса функционирования системы S , т. е. об ее концептуальной модели M_k как совокупности некоторых элементов, условно изображенных квадратами так, как показано на рис. 2., а. Эти квадраты представляют собой описание некоторых подпроцессов исследуемого процесса функционирования системы S , воздействия внешней среды E и т. д. Переход от описания системы к ее модели в этой интерпретации сводится к исключению из рассмотрения некоторых второстепенных элементов описания (элементы 5 - 8, 39 - 41, 43 - 47). Предполагается, что они не оказывают существенного влияния на ход процессов, исследуемых с помощью модели. Часть элементов (14, 15, 28, 29, 42) заменяется пассивными связями $h1$, отражающими внутренние свойства системы (рис. 2, б). Некоторая часть элементов (1 - 4, 10, 11, 24, 25) - заменяется входными факторами x и воздействиями внешней среды $u1$. Возможны и комбинированные замены: элементы 9, 18, 19, 32, 33 заменены пассивной связью $h2$ и воздействием внешней среды E . Элементы 22, 23, 36, 37 отражают воздействие системы на внешнюю среду y .

Оставшиеся элементы системы S группируются в блоки **SI**, **SII**, **SIII**, отражающие процесс функционирования исследуемой системы. Каждый из этих блоков достаточно автономен, что выражается в минимальном количестве связей между ними. Поведение этих блоков должно быть хорошо изучено и для каждого из них построена математическая модель, которая в свою очередь может содержать ряд подблоков.

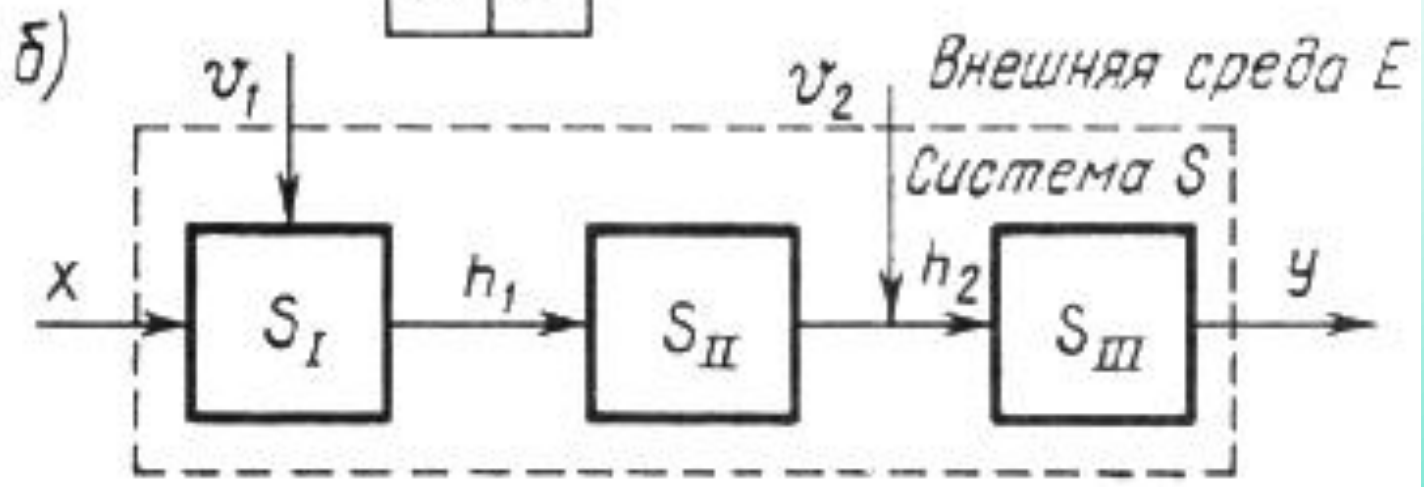
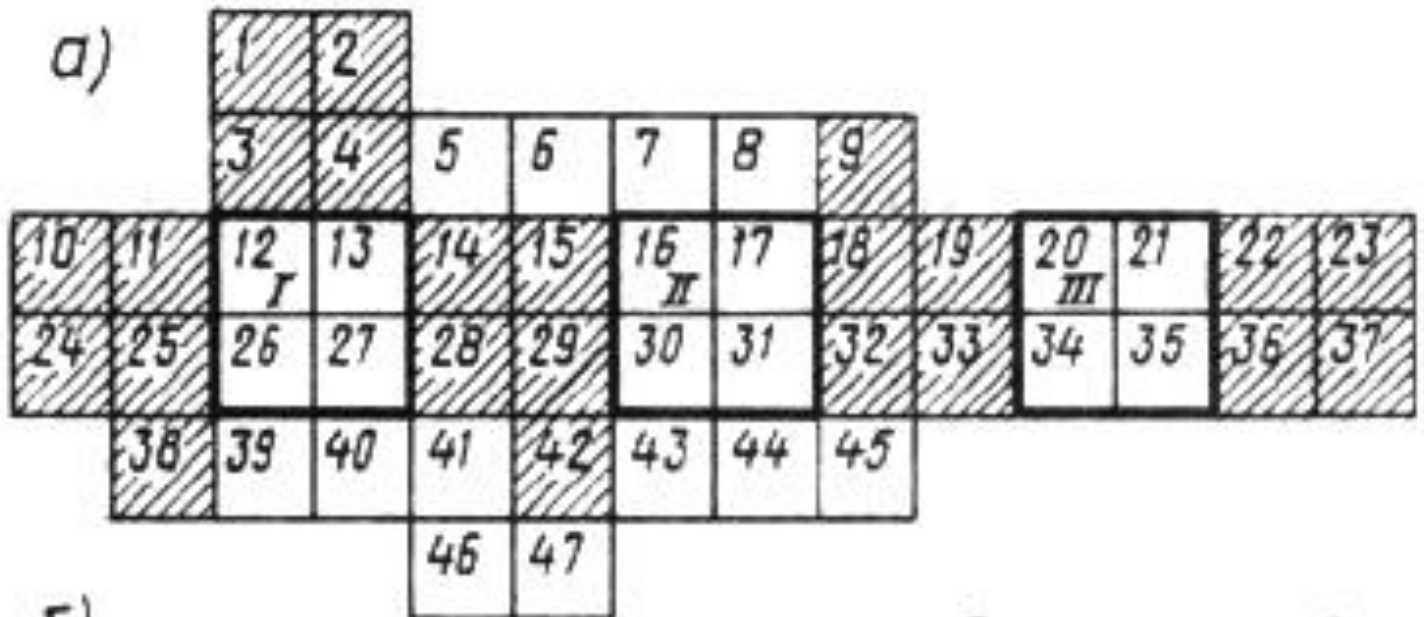


Рис. 2. Модель системы: а - концептуальная; б - блочная

Построенная *блочная модель* процесса функционирования исследуемой системы **S** предназначена для анализа характеристик этого процесса, который может быть проведен при машинной реализации полученной модели.

Математические модели процессов. После перехода от описания моделируемой системы **S** к ее модели **M**, построенной по блочному принципу, необходимо построить математические модели процессов, происходящих в различных блоках. Математическая модель представляет собой совокупность соотношений (например, уравнений, логических условий, операторов), определяющих характеристики процесса функционирования системы **S** в зависимости от структуры системы, алгоритмов поведения, параметров системы, воздействий внешней среды **E**, начальных условий и времени.

Математическая модель является результатом формализации процесса функционирования исследуемой системы, т. е. построения формального (математического) описания процесса с необходимой в рамках проводимого исследования степенью приближения к действительности.

Для иллюстрации возможностей формализации рассмотрим процесс функционирования некоторой гипотетической системы **S**, которую можно разбить на m подсистем с характеристиками $y_1(t), y_2(t), \dots, y_{h_Y}(t)$ с параметрами h_1, h_2, \dots, h_{n_H} при наличии входных воздействия x_1, x_2, \dots, x_{n_H} и воздействий внешней среды u_1, u_2, \dots, u_{n_H} . Тогда математической моделью процесса может служить система соотношений вида

Таким образом, формализации процесса функционирования любой системы **S** должно предшествовать изучение составляющих его явлений. В результате появляется содержательное описание процесса, которое представляет собой первую попытку четко изложить закономерности, характерные для исследуемого процесса, и постановку прикладной задачи. Содержательное описание является исходным материалом для последующих этапов формализации: построения формализованной схемы процесса функционирования системы и математической модели этого процесса. Для моделирования процесса функционирования системы на ЭВМ необходимо преобразовать математическую модель процесса в соответствующий моделирующий алгоритм и машинную программу.

Подэтапы первого этапа моделирования. Рассмотрим более подробно основные подэтапы построения концептуальной модели M_k , системы и ее формализации (см. рис. 1).

1.1. Постановка задачи машинного моделирования системы. Дается четкая формулировка задачи исследования конкретной системы **S** и основное внимание уделяется таким вопросам, как: а) признание существования задачи и необходимости машинного моделирования; б) выбор методики решения задачи с учетом имеющихся ресурсов; в) определение масштаба задачи и возможности разбиения ее на подзадачи.

Необходимо также ответить на вопрос о приоритетности решения различных подзадач, оценить эффективность возможных математических методов и программно-технических средств их решения. Тщательная проработка этих вопросов позволяет сформулировать задачу исследования и приступить к ее реализации. При этом возможен пересмотр начальной постановки задачи в процессе моделирования.

1.2. Анализ задачи моделирования системы. Проведение анализа задачи способствует преодолению возникающих в дальнейшем трудностей при ее решении методом моделирования. На рассматриваемом втором этапе основная работа сводится именно к проведению анализа, включая: а) выбор критериев оценки эффективности процесса функционирования системы **S**; б) определение эндогенных и экзогенных переменных модели **M**; в) выбор возможных методов идентификации; г) выполнение предварительного анализа содержания второго этапа алгоритмизации модели системы и ее машинной реализации; д) выполнение предварительного анализа содержания третьего этапа получения и интерпретации результатов моделирования системы.

1.3. Определение требований к исходной информации об объекте моделирования и организация ее сбора. После постановки задачи моделирования системы **S** определяются требования к информации, из которой получают качественные и количественные исходные данные, необходимые для решения этой задачи. Эти данные помогают глубоко разобраться в сущности задачи, методах ее решения.

Таким образом, на этом подэтапе проводится: а) выбор необходимой информации о системе **S** и внешней среде **E**; б) подготовка априорных данных; в) анализ имеющихся экспериментальных данных; г) выбор методов и средств предварительной обработки информации о системе.

При этом необходимо помнить, что именно от качества исходной информации об объекте моделирования существенно зависят как адекватность модели, так и достоверность результатов моделирования.

1.4. Выдвижение гипотез и принятие предположений. Гипотезы при построении модели системы **S** служат для заполнения "пробелов" в понимании задачи исследователем. Выдвигаются также гипотезы относительно возможных результатов моделирования системы **S**, справедливость которых проверяется при проведении машинного эксперимента. Предположения предусматривают, что некоторые данные неизвестны или их нельзя получить. Предположения могут выдвигаться относительно известных данных, которые не отвечают требованиям решения поставленной задачи. Предположения дают возможность провести упрощения модели в соответствии с выбранным уровнем моделирования. При выдвижении гипотез и принятии предположений учитываются следующие факторы: а) объем имеющейся информации для решения задач; б) подзадачи, для которых информация недостаточна; в) ограничения на ресурсы времени для решения задачи; г) ожидаемые результаты моделирования.

Таким образом, в процессе работы с моделью системы **S** возможно многократное возвращение к этому подэтапу в зависимости от полученных результатов моделирования и новой информации об объекте.

1.5. Определение параметров и переменных модели. Прежде чем перейти к описанию математической модели, необходимо определить параметры системы h_k , $k = 1, n_H$, входные и выходные переменные x_i , $i = 1, n_X$, y_j , $j = 1, n_Y$, воздействия внешней среды u_l , $l = 1, n_V$. Конечной целью этого подэтапа является подготовка к построению математической модели системы **S**, функционирующей во внешней среде **E**, для чего необходимо рассмотрение всех параметров и переменных модели и оценка степени их влияния на процесс функционирования системы в целом. Описание каждого параметра и переменной должно даваться в следующей форме: а) определение и краткая характеристика; б) символ обозначения и единица измерения; в) диапазон изменения; г) место применения в модели.

1.6. Установление основного содержания модели. На этом подэтапе определяется основное содержание модели и выбирается метод построения модели системы, которые разрабатываются на основе принятых гипотез и предположений. При этом учитываются следующие особенности: а) формулировка задачи моделирования системы; б) структура системы **S** и алгоритмы ее поведения, воздействия внешней среды **E**; в) возможные методы и средства решения задачи моделирования.

1.7. Обоснование критериев оценки эффективности системы. Для оценки качества процесса функционирования моделируемой системы **S** необходимо выбрать некоторую совокупность критериев оценки эффективности, т. е. в математической постановке задача сводится к получению соотношения для оценки эффективности как функции параметров и переменных системы.

Эта функция представляет собой поверхность отклика в исследуемой области изменения параметров и переменных и позволяет определить реакцию системы. Эффективность системы S можно оценить с помощью интегральных или частных критериев, выбор которых зависит от рассматриваемой задачи.

1.8. Определение процедур аппроксимации. Для аппроксимации реальных процессов, протекающих в системе S , обычно используются три вида процедур: а) детерминированную; б) вероятностную; в) определения средних значений. При детерминированной процедуре результаты моделирования однозначно определяются по данной совокупности входных воздействий, параметров и переменных системы S . В этом случае отсутствуют случайные элементы, влияющие на результаты моделирования. Вероятностная (рандомизированная) процедура применяется в том случае, когда случайные элементы, включая воздействия внешней среды E , влияют на характеристики процесса функционирования системы S и когда необходимо получить информацию о законах распределения выходных переменных. Процедура определения средних значений используется тогда, когда при моделировании системы интерес представляют средние значения выходных переменных при наличии случайных элементов.

1.9. Описание концептуальной модели системы. На этом подэтапе построения модели системы: а) описывается концептуальная модель M_k в абстрактных терминах и понятиях; б) дается описание модели с использованием типовых математических схем; в) принимаются окончательно гипотезы и предположения; г) обосновывается выбор процедуры аппроксимации реальных процессов при построении модели.

Таким образом, на этом подэтапе проводится подробный анализ задачи, рассматриваются возможные методы ее решения и дается детальное описание концептуальной модели M_k , которая затем используется на втором этапе моделирования.

1.10. Проверка достоверности концептуальной модели. После того как концептуальная модель M_k описана, необходимо проверить достоверность некоторых концепций модели перед тем, как перейти к следующему этапу моделирования системы S . Проверять достоверность концептуальной модели достаточно сложно, так как процесс ее построения является эвристическим и такая модель описывается в абстрактных терминах и понятиях. Один из методов проверки модели M_k - применение операций обратного перехода, позволяющий проанализировать модель, вернуться к принятым аппроксимациям и, наконец, рассмотреть снова реальные процессы, протекающие в моделируемой системе S . Проверка достоверности концептуальной модели M_k должна включать: а) проверку замысла модели; б) оценку достоверности исходной информации; в) рассмотрение постановки задачи моделирования; г) анализ принятых аппроксимаций; д) исследование гипотез и предположений.

Только после тщательной проверки концептуальной модели M_k следует переходить к этапу машинной реализации модели, так как ошибки в модели M_k не позволяют получить достоверные результаты моделирования.

1.11. Составление технической документации по первому этапу. В конце этапа построения концептуальной модели M_k и ее формализации составляется технический отчет по этапу, который включает в себя: а) подробную постановку задачи моделирования системы S ; б) анализ задачи моделирования системы; в) критерии оценки эффективности системы; г) параметры и переменные модели системы; д) гипотезы и предположения, принятые при построении модели; е) описание модели в абстрактных терминах и понятиях; ж) описание ожидаемых результатов моделирования системы S .

Составление технической документации - обязательное условие успешного проведения моделирования системы S , так как в процессе разработки модели большой системы и ее машинной реализации принимают участие на различных этапах коллективы специалистов разных профилей (начиная от постановщиков задач и кончая программистами) и документация является средством обеспечения их эффективного взаимодействия при решении поставленной задачи методом моделирования.

АЛГОРИТМИЗАЦИЯ МОДЕЛЕЙ СИСТЕМ И ИХ МАШИННАЯ РЕАЛИЗАЦИЯ

На втором этапе моделирования - этапе алгоритмизации модели и ее машинной реализации - математическая модель, сформированная на первом этапе, воплощается в конкретную машинную модель.

Этот этап представляет собой этап практической деятельности, направленной на реализацию идей и математических схем в виде машинной модели M_M процесса функционирования системы S . Прежде чем рассматривать подэтапы алгоритмизации и машинной реализации модели, остановимся на основных принципах построения моделирующих алгоритмов и формах их представления.

Принципы построения моделирующих алгоритмов. Процесс функционирования системы S можно рассматривать как последовательную смену ее состояний $\vec{z} = z(z_1(t), z_2(t), \dots, z_k(t))$ в k -мерном пространстве. Очевидно, что задачей моделирования процесса функционирования исследуемой системы S является построение функций z , на основе которых можно провести вычисление интересующих характеристик процесса функционирования системы. Для этого должны иметься соотношения, связывающие функции z с переменными, параметрами и временем, а также начальные условия $\vec{z}_0 = z(z_1(t_0), z_2(t_0), \dots, z_k(t_0))$ в момент времени $t = t_0$.

Рассмотрим процесс функционирования некоторой детерминированной системы S_D , в которой отсутствуют случайные факторы, т. е. вектор состояний такой системы можно определить как $\vec{z} = \Phi(\vec{z}_0, \vec{x}, t)$. Тогда состояние процесса в момент времени $t_0 + j\Delta t$ может быть однозначно определено из соотношений математической модели по известным начальным условиям.

Это позволяет строить моделирующий алгоритм процесса функционирования системы. Для этого преобразуем соотношения модели \mathbf{Z} к такому виду, чтобы сделать удобным вычисление $z_1(t + \Delta t), z_2(t + \Delta t), \dots, z_k(t + \Delta t)$ по значениям $z_i(\tau)$, $i = 1, k$, где $\tau \leq t$. Организуем счетчик системного времени, который в начальный момент показывает время t_0 . Для этого момента $z_i(t_0) = z_{i0}$. Прибавим интервал времени Δt , тогда счетчик будет показывать $t_1 = t_0 + \Delta t$. Вычислим значения $z_i(t_0 + \Delta t)$. Затем перейдем к моменту времени $t_2 = t_1 + \Delta t$ и т. д. Если шаг Δt достаточно мал, то таким путем можно получить приближенные значения z .

Рассмотрим процесс функционирования стохастической системы \mathbf{SR} , т. е. системы, на которую оказывают воздействия случайные факторы. Для такой системы функция состояний процесса z в момент времени $\tau \leq t$ и соотношения модели определяют лишь распределение вероятностей для $z_i(t_0 + \Delta t)$ в момент времени $t + \Delta t$. В общем случае и начальные условия z_0 могут быть случайными, задаваемыми соответствующим распределением вероятностей. При этом структура моделирующего алгоритма для стохастических систем в основном остается прежней. Только вместо состояния $z_i(t_0 + \Delta t)$ теперь необходимо вычислить распределение вероятностей для возможных состояний. Пусть счетчик системного времени показывает время t_0 . В соответствии с заданным распределением вероятностей выбирается z_{0i} . Далее, исходя из распределения, получается состояние $z_i(t_0 + \Delta t)$ и т. д., пока не будет построена одна из возможных реализаций случайного многомерного процесса $z_i(t)$ в заданном интервале времени.

Рассмотренный принцип построения моделирующих алгоритмов называется *принципом Δt* . Это наиболее универсальный принцип, позволяющий определить последовательные состояния процесса функционирования системы **S** через заданные интервалы времени Δt . Но с точки зрения затрат машинного времени он иногда оказывается неэкономичным.

При рассмотрении процессов функционирования некоторых систем можно обнаружить, что для них характерны два типа состояний: 1) особые, присущие процессу функционирования системы только в некоторые моменты времени (моменты поступления входных или управляющих воздействий, возмущений внешней среды и т. п.); 2) неособые, в которых процесс находится все остальное время. Особые состояния характерны еще и тем обстоятельством, что функции состояний $z_i(t)$ в эти моменты времени изменяются скачком, а между особыми состояниями изменение координат $z_i(t)$ происходит плавно и непрерывно или не происходит совсем. Таким образом, следя при моделировании системы **S** только за ее особыми состояниями в те моменты времени, когда эти состояния имеют место, можно получить информацию, необходимую для построения функций $z_i(t)$. Очевидно, для описанного типа систем могут быть построены моделирующие алгоритмы по "принципу особых состояний". Обозначим скачкообразное (релейное) изменение состояния z как δz , а "принцип особых состояний" - как *принцип δz* .

Например, для системы массового обслуживания (*Q-схемы*) в качестве особых состояний могут быть выбраны состояния в моменты поступления заявок на обслуживание в прибор Π и в моменты окончания обслуживания заявок каналами K , когда состояние системы, оцениваемое числом находящихся в ней заявок, меняется скачком.

Отметим, что характеристики процесса функционирования таких систем с особыми состояниями оцениваются по информации об особых состояниях, а неособые состояния при моделировании не рассматриваются. "Принцип δz " дает возможность для ряда систем существенно уменьшить затраты машинного времени на реализацию моделирующих алгоритмов по сравнению с "принципом δz ". Логика построения моделирующего алгоритма, реализующего "принцип δz ", отличается от рассмотренной для "принципа δz " только тем, что включает в себя процедуру определения момента времени t_{δ} , соответствующего следующему особому состоянию системы S . Для исследования процесса функционирования больших систем рационально использование комбинированного принципа построения моделирующих алгоритмов, сочетающего в себе преимущества каждого из рассмотренных принципов.

Формы представления моделирующих алгоритмов. Удобной формой представления логической структуры моделей процессов функционирования систем и машинных программ является схема. На различных этапах моделирования составляются обобщенные и детальные логические схемы моделирующих алгоритмов, а также схемы программ.

Обобщенная (укрупненная) схема моделирующего алгоритма задает общий порядок действий при моделировании системы без каких-либо уточняющих деталей. Обобщенная схема показывает, что необходимо выполнить на очередном шаге моделирования, например обратиться к датчику случайных чисел.

Детальная схема моделирующего алгоритма содержит уточнения, отсутствующие в обобщенной схеме. Детальная схема показывает не только, что следует выполнить на очередном шаге моделирования системы, но и как это выполнить.

Логическая схема моделирующего алгоритма представляет собой логическую структуру модели процесса функционирования системы **S**. Логическая схема указывает упорядоченную во времени последовательность логических операций, связанных с решением задачи моделирования. **Схема программы** отображает порядок программной реализации моделирующего алгоритма с использованием конкретного математического обеспечения. Схема программы представляет собой интерпретацию логической схемы моделирующего алгоритма разработчиком программы на базе конкретного алгоритмического языка. Различие между этими схемами заключается в том, что логическая схема отражает логическую структуру модели процесса функционирования системы, а схема программы - логику машинной реализации модели с использованием конкретных программно-технических средств моделирования.

Логическая схема алгоритма и схема программы могут быть выполнены как в укрупненной, так и в детальной форме. Для начертания этих схем используется набор символов, определяемых ГОСТ 19.701 - 90 (ИСО 5807 - 85) "Единая система программной документации."

Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения". Некоторые наиболее употребительные в практике моделирования на ЭВМ символы показаны на рис. 3, где изображены основные, специфические и специальные символы процесса. К ним относятся: основной символ: *а* - процесс - символ отображает функцию обработки данных любого вида (выполнение определенной операции или группы операций, приводящее к изменению значения, формы или размещения информации или к определению, по которому из нескольких направлений потока следует двигаться); специфические символы процесса: *б* - решение - символ отображает решение или функцию переключательного типа, имеющую один вход и ряд альтернативных выходов, один и только один из которых может быть активизирован после вычисления условий, определенных внутри этого символа (соответствующие результаты вычисления могут быть записаны по соседству с линиями, отображающими эти пути); *в* - подготовка - символ отображает модификацию команды или группы команд с целью воздействия на некоторую последующую функцию (установка переключателя, модификация индексного регистра или инициализация программы); *г* - predetermined process - символ отображает predetermined process, состоящий из одной или нескольких операций или шагов программы, которые определены в другом месте (в подпрограмме, модуле); *д* - ручная операция - символ отображает любой процесс, выполняемый человеком;

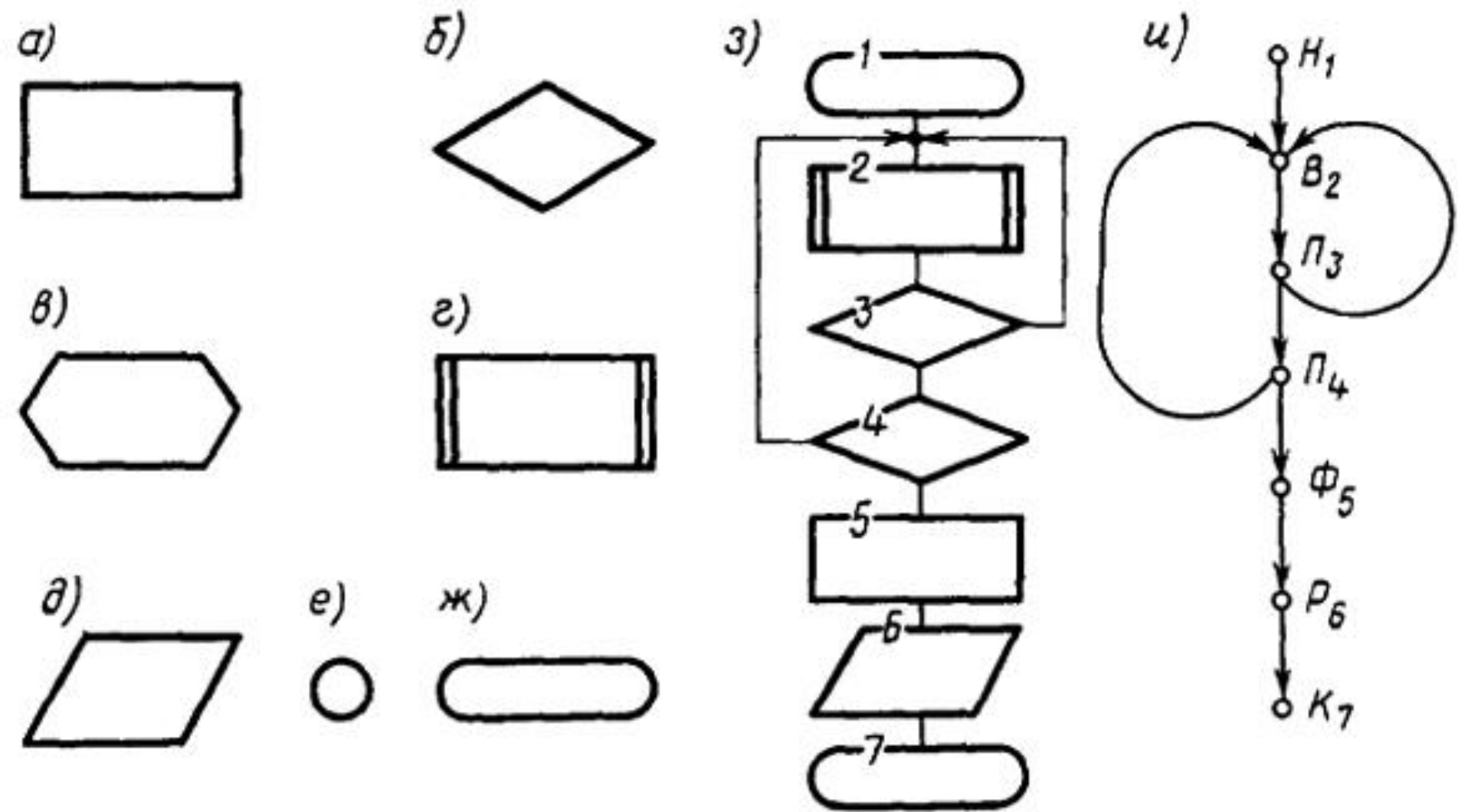


Рис. 3. Символы и схемы моделирующих алгоритмов

специальные символы: e - соединитель - символ отображает выход в часть схемы и вход из другой части этой схемы и используется для обрыва линии и продолжения ее в другом месте (соответствующие символы-соединители должны содержать одно и то же уникальное обозначение); $ж$ - терминатор - символ отображает выход во внешнюю среду и вход из внешней среды (начало или конец схемы алгоритма, внешнее использование или пункт назначения данных).

Пример изображения схемы моделирующего алгоритма показан на рис. 3, з.

Обычно схема является наиболее удобной формой представления структуры моделирующих алгоритмов. В ряде случаев используются и другие формы представления моделирующих алгоритмов, например форма *граф-схем* (рис. 3, и). Здесь H_i - начало, K_i - конец, B_i - вычисление, Φ_i - формирование, Π_i - проверка условия, C_i - счетчик, P_i - выдача результата, $i = 1, g$, где g - общее число операторов моделирующего алгоритма. В качестве пояснения к граф-схеме алгоритма в тексте дается раскрытие содержания операторов, что позволяет упростить представление алгоритма, но усложняет работу с ним.

Моделирующие алгоритмы могут быть также представлены в виде операторных схем. Обозначения операторов на такой схеме соответствуют обозначениям для граф-схем. Для рассмотренного примера операторная схема алгоритма имеет вид

$$H_1^{3,4} B_2 \Pi_3 \Pi_4 \Phi_5 P_6 K_7.$$

Более подробно с формой представления логической структуры моделирующих алгоритмов и машинных программ познакомимся при рассмотрении имитационных моделей процессов функционирования различных систем и способов их реализации на ЭВМ.

Подэтапы второго этапа моделирования. Рассмотрим подэтапы, выполненные при алгоритмизации модели системы и ее машинной реализации, обращая основное внимание на задачи каждого подэтапа и методы их решения.

2.1. Построение логической схемы модели. Рекомендуется строить модель по блочному принципу, т. е. в виде некоторой совокупности стандартных блоков. Построение модели систем **S** из таких блоков обеспечивает необходимую гибкость в процессе ее эксплуатации, особенно на стадии машинной отладки. При построении блочной модели проводится разбиение процесса функционирования системы на отдельные достаточно автономные подпроцессы. Таким образом, модель функционально подразделяется на подмодели, каждая из которых в свою очередь может быть разбита на еще более мелкие элементы. Блоки такой модели бывают двух типов: основные и вспомогательные. Каждый основной блок соответствует некоторому реальному подпроцессу, имеющему место в моделируемой системе **S**, а вспомогательные блоки представляют собой лишь составную часть машинной модели, они не отражают функции моделируемой системы и необходимы лишь для машинной реализации, фиксации и обработки результатов моделирования.

2.2. Получение математических соотношений. Одновременно с выполнением подэтапа построения логической схемы модели необходимо получить, если это возможно, математические соотношения в виде явных функций, т. е. построить аналитические модели. Этот подэтап соответствует неявному заданию возможных математических соотношений на этапе построения концептуальной модели. При выполнении первого этапа еще не может иметься информации о конкретном виде таких математических соотношений, а на втором этапе уже необходимо получить эти соотношения. Схема машинной модели M_k должна представлять собой полное отражение заложен ной в модели концепции и иметь: а) описание всех блоков модели с их наименованиями; б) единую систему обозначений и нумерацию блоков; в) отражение логики модели процесса функционирования системы; г) задание математических соотношений в явном виде.

Таким образом, в общем случае построенная машинная модель M_k системы будет иметь комбинированный характер, т. е. отражать аналитико-имитационный подход, когда часть процесса в системе описана аналитически, а другая часть имитируется соответствующими алгоритмами.

2.3. Проверка достоверности модели системы. Эта проверка является первой из проверок, выполняемых на этапе реализации модели. Так как модель представляет собой приближенное описание процесса функционирования реальной системы **S**, то до тех пор, пока не доказана достоверность модели **M_к**, нельзя утверждать, что с ее помощью будут получены результаты, совпадающие с теми, которые могли бы быть получены при проведении натурального эксперимента с реальной системой **S**. Поэтому определение достоверности модели можно считать наиболее важной проблемой при моделировании систем. От решения этой проблемы зависит степень доверия к результатам, полученным методом моделирования. Проверка модели на рассматриваемом подэтапе должна дать ответ на вопрос, насколько логическая схема модели системы и используемые математические соотношения отражают замысел модели, сформированный на первом этапе. При этом проверяются: а) возможность решения поставленной задачи; б) точность отражения замысла в логической схеме; в) полнота логической схемы модели; г) правильность используемых математических соотношений. Только после того, как разработчик убеждается путем соответствующей проверки в правильности всех этих положений, можно считать, что имеется логическая схема модели системы **S**, пригодная для дальнейшей работы по реализации модели на ЭВМ.

2.4. Выбор инструментальных средств для моделирования. На этом подэтапе необходимо окончательно решить вопрос о том, какую вычислительную машину (ЭВМ, АВМ, ГВК) и какое программное обеспечение целесообразно использовать для реализации модели системы **S**. Вообще, выбор вычислительных средств может быть проведен и на предыдущих подэтапах, но рассматриваемый подэтап является последним, когда этот выбор должен быть сделан окончательно, так как в противном случае возникнут трудности в проведении дальнейших работ по реализации модели. Вопрос о выборе ЭВМ сводится к обеспечению следующих требований: а) наличие необходимых программных и технических средств; б) доступность выбранной ЭВМ для разработчика модели; в) обеспечение всех этапов реализации модели; г) возможность своевременного получения результатов.

2.5. Составление плана выполнения работ по программированию. Такой план должен помочь при программировании модели, учитывая оценки объема программы и трудозатрат на ее составление. План при использовании универсальной ЭВМ должен включать в себя: а) выбор языка (системы) программирования модели; б) указание типа ЭВМ и необходимых для моделирования устройств; в) оценку примерного объема необходимой оперативной и внешней памяти; г) ориентировочные затраты машинного времени на моделирование; д) предполагаемые затраты времени на программирование и отладку программы на ЭВМ.

2.6. Спецификация и построение схемы программы. Спецификация программы - формализованное представление требований, предъявляемых к программе, которые должны быть удовлетворены при ее разработке, а также описание задачи, условия и эффекта действия без указания способа его достижения. Наличие логической блок-схемы модели позволяет построить схему программы, которая должна отражать: а) разбиение модели на блоки, подблоки и т. д.; б) особенности программирования модели; в) проведение необходимых изменений; г) возможности тестирования программы; д) оценку затрат машинного времени; е) форму представления входных и выходных данных.

Построение схемы программы представляет собой одну из основных задач на этапе машинной реализации модели. При этом особое внимание должно быть уделено особенностям выбранного для реализации модели языка: алгоритмического языка общего назначения или языка моделирования (например, *SIMULA*, *SIMS CRIPT*, *GPS S*).

2.7. Верификация и проверка достоверности схемы программы. Верификация программы - доказательство того, что поведение программы соответствует спецификации на программу. Эта проверка является второй на этапе машинной реализации модели системы. Очевидно, что нет смысла продолжать работу по реализации модели, если нет уверенности в том, что в схеме программы, по которой будет вестись дальнейшее программирование, допущены ошибки, которые делают ее неадекватной логической схеме модели, а следовательно, и неадекватной самому объекту моделирования. При этом проводится проверка соответствия каждой операции, представленной в схеме программы, аналогичной ей операции в логической схеме модели.

2.8. Проведение программирования модели. При достаточно подробной схеме программы, которая отражает все операции логической схемы модели, можно приступить к программированию модели. Если имеется адекватная схема программы, то программирование представляет собой работу только для программиста без участия и помощи со стороны разработчика модели. При использовании пакетов прикладных программ моделирования проводится не посредственная генерация рабочих программ для моделирования конкретного объекта, т. е. программирование модели реализуется в автоматизированном режиме.

2.9. Проверка достоверности программы. Эта последняя проверка на этапе машинной реализации модели, которую необходимо проводить: а) обратным переводом программы в исходную схему; б) проверкой отдельных частей программы при решении различных тестовых задач; в) объединением всех частей программы и проверкой ее в целом на контрольном примере моделирования варианта системы **S**.

На этом подэтапе необходимо также проверить оценки затрат машинного времени на моделирование. Полезно также получить достаточно простую аналитическую аппроксимацию зависимости затрат машинного времени от количества реализаций, что позволит разработчику модели (заказчику) правильно сформулировать требования к точности и достоверности результатов моделирования.

2.10. Составление технической документации по второму этапу. Для завершения этапа машинной реализации модели **Мм** необходимо составить техническую документацию, содержащую: а) логическую схему модели и ее описание; б) адекватную схему программы и принятые обозначения; в) полный текст программы; г) перечень входных и выходных величин с пояснениями; д) инструкцию по работе с программой; е) оценку затрат машинного времени на моделирование с указанием требуемых ресурсов ЭВМ.

Таким образом, на этом этапе разрабатывается схема модели системы **S**, проводится ее алгоритмизация и программирование с использованием конкретных программно-технических средств, т. е. строится машинная модель **Мм**, с которой предстоит работать для получения необходимых результатов моделирования по оценке характеристик процесса функционирования системы **S** (задача анализа) или для поиска оптимальных структур, алгоритмов и параметров системы **S** (задача синтеза).

ПОЛУЧЕНИЕ И ИНТЕРПРЕТАЦИЯ РЕЗУЛЬТАТОВ МОДЕЛИРОВАНИЯ СИСТЕМ

На третьем этапе моделирования - этапе получения и интерпретации результатов моделирования - ЭВМ используется для проведения рабочих расчетов по составленной и отлаженной программе. Результаты этих расчетов позволяют проанализировать и сформулировать выводы о характеристиках процесса функционирования моделируемой системы **S**.

Особенности получения результатов моделирования. При реализации моделирующих алгоритмов на ЭВМ вырабатывается информация о состояниях процесса функционирования исследуемых систем $z(t)$, Z . Эта информация является исходным материалом для определения приближенных оценок искомых характеристик, получаемых в результате машинного эксперимента, т. е. критериев оценки. *Критерием оценки* будем называть любой количественный показатель, по которому можно судить о результатах моделирования системы. Критериями оценки могут служить показатели, получаемые на основе процессов, действительно протекающих в системе, или получаемых на основе специально сформированных функций этих процессов.

В ходе машинного эксперимента изучается поведение исследуемой модели M процесса функционирования системы S на заданном интервале времени $[0, T]$. Поэтому критерий оценки является в общем случае векторной случайной функцией, заданной на этом же интервале:

$$\vec{q}(t) = (q_1(t), q_2(t), \dots, q_n(t)).$$

Часто используют более простые критерии оценки, например вероятность определенного состояния системы в заданный момент времени t^* , отсутствие отказов и сбоев в системе на интервале $[0, T]$ и т. д. При интерпретации результатов моделирования вычисляются различные статистические характеристики закона распределения критерия оценки.

Рассмотрим общую схему фиксации и обработки результатов моделирования системы, которая приведена на рис. 4. Будем рассматривать гипотетическую модель M , предназначенную для исследования поведения системы S на интервале времени $[0, T]$. В общем случае критерием интерпретации результатов моделирования является нестационарный случайный n -мерный процесс $\vec{q}(t)$, $0 \leq t \leq T$. Полагаем для определенности, что состояние моделируемой системы S проверяется каждые Δt временных единиц, т. е. используется "принцип Δt ". При этом вычисляют значения $\vec{q}(j\Delta t)$, $j = 0, 1, \dots, k$, критерия $\vec{q}(t)$. Таким образом, о свойствах случайного процесса $\vec{q}(t)$ судят по свойствам случайной последовательности $\vec{q}(j\Delta t)$, $j = 0, 1, \dots, k$, или, иначе говоря, по свойствам m -мерного вектора.

Процесс функционирования системы S на интервале $[0, T]$ моделируется N -кратно с получением независимых реализаций i , $i = 1, 2, \dots, N$, вектора \vec{q} . Работа модели на интервале $[0, T]$ называется *прогоном модели*.

На схеме, изображенной на рис. 4, обозначено: $I \equiv i$; $J \equiv j$; $K \equiv k$; $N \equiv N$; $T \equiv t$; $DT \equiv \Delta t$; $Q \equiv q$.

В общем случае алгоритмы фиксации и статистической обработки данных моделирования содержат три цикла. Полагаем, что имеется машинная модель M_m системы S .

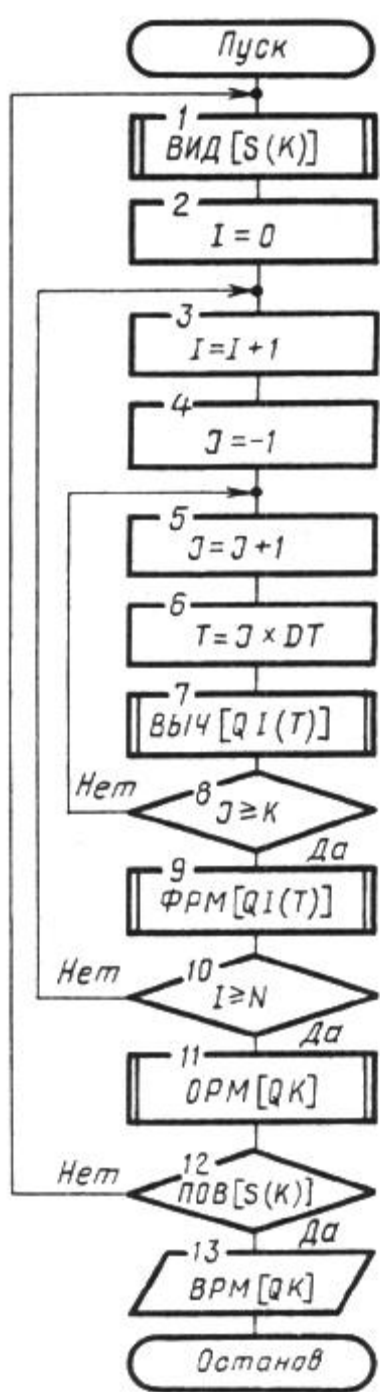


Рис. 4. Алгоритм фиксации и обработки результатов моделирования системы

Подэтапы третьего этапа моделирования. Прежде чем приступить к последнему, третьему, этапу моделирования системы, необходимо для его успешного проведения иметь четкий план действий, сводящийся к выполнению следующих основных подэтапов.

3.1. Планирование машинного эксперимента с моделью системы. Перед выполнением рабочих расчетов на ЭВМ должен быть составлен план проведения эксперимента с указанием комбинаций переменных и параметров, для которых должно проводиться моделирование системы **S**. Планирование машинного эксперимента призвано дать в итоге максимальный объем необходимой информации об объекте моделирования при минимальных затратах машинных ресурсов. При этом различают стратегическое и тактическое планирование машинного эксперимента. При стратегическом планировании эксперимента ставится задача построения оптимального плана эксперимента для достижения цели, поставленной перед моделированием (например, оптимизация структуры, алгоритмов и параметров системы **S**, исследуемой методом моделирования на ЭВМ). Тактическое планирование машинного эксперимента преследует частные цели оптимальной реализации каждого конкретного эксперимента из множества необходимых, заданных при стратегическом планировании (например, решение задачи выбора оптимальных правил остановки при статистическом моделировании системы **S** на ЭВМ). Для получения наиболее эффективного плана машинного эксперимента необходимо использовать статистические методы.

3.2. Определение требований к вычислительным средствам.

Необходимо сформулировать требования по времени использования вычислительных средств, т. е. составить график работы на одной или нескольких ЭВМ, а также указать те внешние устройства ЭВМ, которые потребуются при моделировании. При этом также рационально оценить, исходя из требуемых ресурсов, возможность использования для реализации конкретной модели персональной ЭВМ или локальной вычислительной сети.

3.3. Проведение рабочих расчетов. После составления программы модели и плана проведения машинного эксперимента с моделью системы **S** можно приступить к рабочим расчетам на ЭВМ, которые обычно включают в себя: а) подготовку наборов исходных данных для ввода в ЭВМ; б) проверку исходных данных, подготовленных для ввода; в) проведение расчетов на ЭВМ; г) получение выходных данных, т. е. результатов моделирования.

Проведение машинного моделирования рационально выполнять в два этапа: контрольные, а затем рабочие расчеты. Причем контрольные расчеты выполняются для проверки машинной модели **M_M** и определения чувствительности результатов к изменению исходных данных.

3.4. Анализ результатов моделирования системы. Чтобы эффективно проанализировать выходные данные, полученные в результате расчетов на ЭВМ, необходимо знать, что делать с результатами рабочих расчетов и как их интерпретировать. Эти задачи могут быть решены на основании предварительного анализа на двух первых этапах моделирования системы **S**. Планирование машинного эксперимента с моделью M_m позволяет вывести необходимое количество выходных данных и определить метод их анализа. При этом необходимо, чтобы на печать выдавались только те результаты, которые нужны для дальнейшего анализа. Также необходимо полнее использовать возможности ЭВМ с точки зрения обработки результатов моделирования и представления этих результатов в наиболее наглядном виде. Вычисление статистических характеристик перед выводом результатов из ЭВМ повышает эффективность применения машины и сводит к минимуму обработку выходной информации после ее вывода из ЭВМ.

3.5. Представление результатов моделирования. Как уже отмечалось, необходимо на третьем этапе моделирования уделить внимание форме представления окончательных результатов моделирования в виде таблиц, графиков, диаграмм, схем и т. п. Целесообразно в каждом конкретном случае выбрать наиболее подходящую форму, так как это существенно влияет на эффективность их дальнейшего употребления заказчиком. В большинстве случаев наиболее простой формой считаются таблицы, хотя графики более наглядно иллюстрируют результаты моделирования системы **S**. При диалоговых режимах моделирования наиболее рациональными средствами оперативного отображения результатов моделирования являются средства мультимедиа технологии.

3.6. Интерпретация результатов моделирования. Получив и проанализировав результаты моделирования, их нужно интерпретировать по отношению к моделируемому объекту, т. е. системе **S**. Основное содержание этого подэтапа - переход от информации, полученной в результате машинного эксперимента с моделью M_m , к информации применительно к объекту моделирования, на основании которой и будут делаться выводы относительно характеристик процесса функционирования исследуемой системы **S**.

3.7. Подведение итогов моделирования и выдача рекомендаций. Проведение этого подэтапа тесно связано с предыдущим вторым этапом (см. п. 3.3). При подведении итогов моделирования должны быть отмечены главные особенности, полученные в соответствии с планом эксперимента над моделью M_m результатов, проведена проверка гипотез и предположений и сделаны выводы на основании этих результатов. Все это позволяет сформулировать рекомендации по практическому использованию результатов моделирования, на пример на этапе проектирования системы **S**.

3.8. Составление технической документации по третьему этапу. Эта

документация должна включать в себя: а) план проведения машинного эксперимента; б) наборы исходных данных для моделирования; в) результаты моделирования системы; г) анализ и оценку результатов моделирования; д) выводы по полученным результатам моделирования; указание путей дальнейшего совершенствования машинной модели и возможных областей ее приложения.

Полный комплект документации по моделированию конкретной системы **S** на ЭВМ должен содержать техническую документацию по каждому из трех рассмотренных этапов.

Таким образом, процесс моделирования системы **S** сводится к выполнению перечисленных этапов моделирования. На этапе построения концептуальной модели M_k проводится исследование моделируемого объекта, определяются необходимые аппроксимации и строится обобщенная схема модели, которая преобразуется в машинную модель M_m на втором этапе моделирования путем последовательного построения логической схемы модели и схемы программы. На последнем этапе моделирования проводят рабочие расчеты на ЭВМ, получают и интерпретируют результаты моделирования системы **S**.

Рассмотренная последовательность этапов и подэтапов отражает наиболее общий подход к построению и реализации модели системы **S**. В дальнейшем остановимся на наиболее важных составляющих процесса моделирования.

Тема 3

Математические схемы моделирования систем

Непрерывно-детерминированные модели (D-схемы)

В качестве математических моделей в *D*-схемах (от английского *dynamic*) используются дифференциальные уравнения.

Дифференциальными уравнениями называются такие уравнения, в которых неизвестными будут функции одной или нескольких переменных, причем в уравнение входят не только функции, но и их производные различных порядков. Если неизвестные – функции многих переменных, то уравнения называются уравнениями в частных производных, в противном случае при рассмотрении функций только одной независимой переменной уравнения называются обыкновенными дифференциальными уравнениями.

Обычно в таких математических моделях в качестве независимой переменной, от которой зависят неизвестные искомые функции, служит время t .

Тогда математическое соотношение для детерминированных систем в общем виде будет

$$\begin{cases} \vec{y}' = \vec{f}(\vec{y}, t), \\ \vec{y}(t_0) = \vec{y}_0, \end{cases}$$

где $\vec{y} = (y_1, y_2, \dots, y_n)$ n -мерный вектор,

$\vec{f} = (f_1, f_2, \dots, f_n)$ n -мерная функция.

Задачей системы является изменение выходной переменной (выходного сигнала) $y(t)$ согласно заданному закону с определенной точностью (с допустимой ошибкой). При проектировании и эксплуатации систем автоматического управления необходимо выбрать такие параметры системы, которые обеспечили бы требуемую точность управления, а также устойчивость системы в переходном процессе.

Если система устойчива, то представляет практический интерес поведение системы во времени, максимальное отклонение регулируемой переменной $y(t)$ в переходном процессе, время переходного процесса и т. п. Выводы о свойствах систем автоматического управления различных классов можно сделать по виду дифференциальных уравнений, приближенно описывающих процессы в системах. Порядок дифференциального уравнения и значения его коэффициентов полностью определяются статическими и динамическими параметрами системы

Дискретно-детерминированные модели (F -схемы)

При использовании такого рода моделей система представляется в виде автомата, перерабатывающего дискретную информацию и меняющего свои внутренние состояния лишь в допустимые моменты времени. Автомат можно представить как некоторое устройство (черный ящик), на которое подаются входные сигналы и снимаются выходные и которое может иметь некоторые внутренние состояния.

Конечным автоматом называется автомат, у которого множество внутренних состояний и входных сигналов (а следовательно, и множество выходных сигналов) являются конечными множествами.

Абстрактно конечный автомат (англ. finite automata) можно представить как математическую схему (F -схему), характеризующуюся шестью элементами:

- конечным множеством X входных сигналов (входным алфавитом);
- конечным множеством Y выходных сигналов (выходным алфавитом);
- конечным множеством Z внутренних состояний (внутренним алфавитом или алфавитом состояний);
- начальным состоянием z_0 , функцией переходов $\Phi(z, x)$, функцией выходов $\Psi(z, x)$.

Автомат, задаваемый F -схемой, функционирует в дискретном автоматном времени, моментами которого являются такты, т. е. примыкающие друг к другу равные интервалы времени. Изменение состояния автомата и выходного сигнала может произойти только в тактовые моменты.

Таким образом, работа конечного автомата происходит по следующей схеме: в каждом t -м такте на вход автомата, находящегося в состоянии $z(t)$, подается некоторый сигнал $x(t)$, на который он реагирует переходом в $(t+1)$ -м такте в новое состояние $z(t+1)$ и выдачей некоторого выходного сигнала. Это можно описать следующими уравнениями:

$$z(t+1) = \Phi[z(t), x(t)], \quad t=0, 1, 2 \dots ;$$
$$y(t) = \Psi[z(t), x(t)], \quad t=0, 1, 2 \dots .$$

Различают два вида конечных автоматов. Приведенные выше уравнения описывают работу автомата первого рода, называемого также автоматом Мили.

Задание конечного автомата Мили

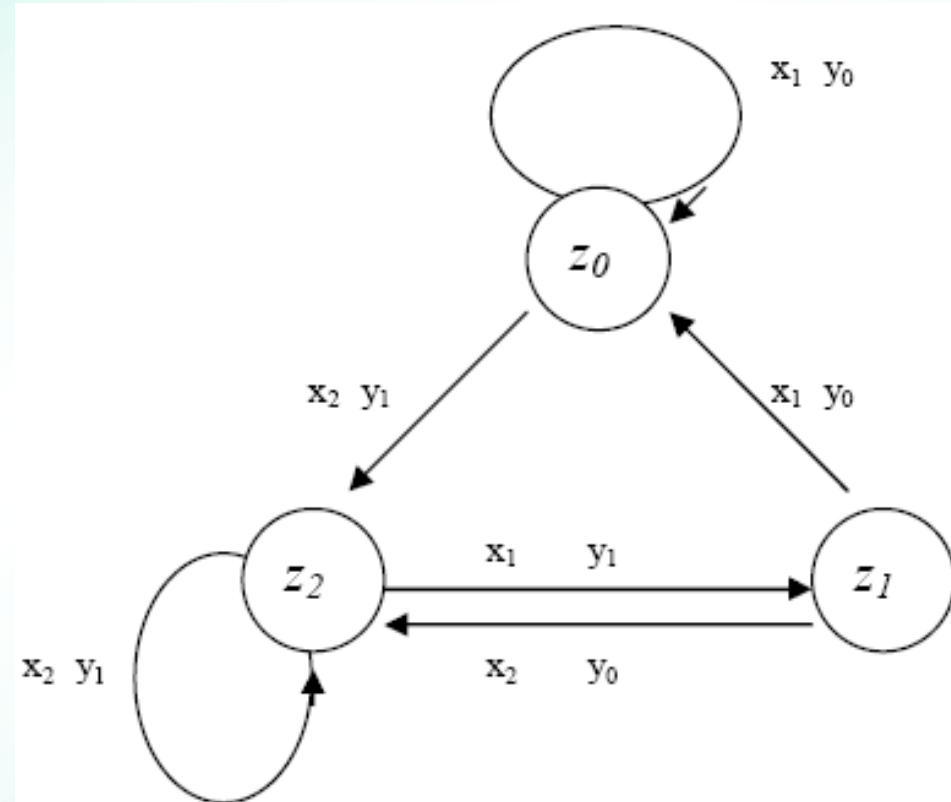
Функция переходов $z_j = \Phi(z_i, x_k)$

Входной сигнал	Состояния		
	z_0	z_1	z_2
x_1	z_0	z_0	z_1
x_2	z_2	z_2	z_2

Функция выходов $y_j = \Psi(z_i, x_k)$

Входной сигнал	Состояния		
	z_0	z_1	z_2
x_1	y_0	y_0	y_1
x_2	y_1	y_0	y_1

табличный способ



с помощью графа

Для F-автомата второго рода

$$z(t+1) = \Phi[z(t), x(t)], \quad t=0, 1, 2, \dots;$$
$$y(t) = \Psi[z(t), x(t-1)], \quad t=1, 2, 3, \dots$$

Автомат второго рода, для которого функция выходов не зависит от входной переменной $x(t)$ называется автоматом Мура. Для него

$$z(t+1) = \Phi[z(t), x(t)], \quad t=0, 1, 2, \dots;$$
$$y(t) = \Psi[z(t)], \quad t=1, 2, 3, \dots$$

По числу состояний различают конечные автоматы с памятью и без памяти. Автоматы с памятью имеют более одного состояния, а автоматы без памяти (комбинационные или логические схемы) обладают лишь одним состоянием. При этом работа комбинационной схемы заключается в том, что она ставит в соответствие каждому входному сигналу $x(t)$ определенный выходной сигнал $y(t)$.

Чтобы задать конечный F -автомат, необходимо описать все элементы множества $F = (Z, X, Y, \Phi, \Psi)$, т. е. входной, внутренней и выходной алфавиты, а также функции переходов и выходов. Причем среди множества состояний необходимо выделить состояние z_0 , в котором автомат находился в момент времени $t = 0$. Существуют несколько способов задания работы F -автоматов, но наиболее часто используются табличный, графический и матричный.

Простейший табличный способ задания конечного автомата основан на использовании таблиц переходов и выходов, строки которых соответствуют входным сигналам автомата, а столбцы — его состояниям. При этом обычно первый слева столбец соответствует начальному состоянию z_0 . На пересечении i -й строки и k -го столбца таблицы переходов помещается соответствующее значение $\Phi(z_k, x_i)$ функции переходов, а в таблице выходов — соответствующее значение $\Psi(z_k, x_i)$ функции выходов. Для F -автомата Мура обе таблицы можно совместить, получив так называемую отмеченную таблицу переходов, в которой над каждым состоянием z_k автомата, обозначающим столбец таблицы, стоит соответствующий этому состоянию выходной сигнал $\Psi(z_k)$.

При другом способе задания конечного автомата используется понятие направленного графа. Граф автомата представляет собой набор вершин, соответствующих различным состояниям автомата и соединяющих вершины дуг графа, соответствующих тем или иным переходам автомата.

Если входной сигнал x_k вызывает переход из состояния z_i в состояние z_j , то на графе автомата дуга, соединяющая вершину z_i с вершиной z_j , отмечается сигналом x_k . Для того чтобы задать функцию выходов, дуги графа необходимо отметить соответствующими выходными сигналами $\Psi(z_i, x_k)$.

Для автомата Мура выходные сигналы связаны только с состояниями и поэтому значениями $\Psi(z_k)$ отмечают соответствующие вершины графа.

Задание конечного автомата Мура

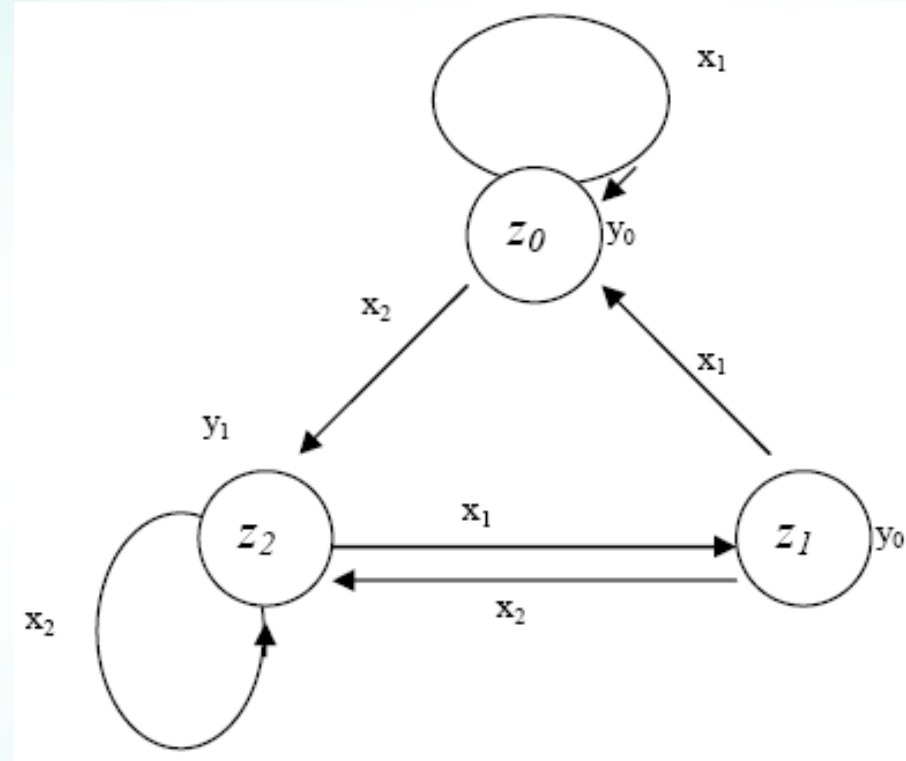
Функция переходов $z_j = \Phi(z_i, x_k)$

Входной сигнал	Состояния		
	z_0	z_1	z_2
x_1	z_0	z_0	z_1
x_2	z_2	z_2	z_2

Функция выходов $y_j = \Psi(z_i)$

Состояния		
z_0	z_1	z_2
y_0	y_0	y_1

табличный способ



с помощью графа

Дискретно-стохастические модели (P -схемы)

В общем виде *вероятностный автомат* (англ. probabilistic automat) можно определить как дискретный потактный преобразователь информации с памятью, функционирование которого в каждом такте зависит только от состояния памяти в нем и может быть описано статистически.

Сущность дискретизации времени при этом подходе остается аналогичной рассмотренным ранее конечным автоматам, но при этом добавляется влияние фактора стохастичности. Применение схем вероятностных автоматов (P -схем) имеет важное значение для разработки методов проектирования дискретных систем, проявляющих статистически закономерное случайное поведение, для выяснения алгоритмических возможностей таких систем и обоснования границ целесообразности их использования, а также для решения задач синтеза по выбранному критерию дискретных стохастических систем, удовлетворяющих заданным ограничениям.

Для того, чтобы задать вероятностный автомат надо, как и для конечного автомата определить множество $X = (x_1, x_2, \dots, x_n)$ входных сигналов, множество $Y = (y_1, y_2, \dots, y_r)$ выходных сигналов и множество $Z = (z_1, z_2, \dots, z_s)$ внутренних состояний. Описание процесса функционирования автомата осуществляется путем задания ряда распределений вероятностей.

Рассмотрим множество G пар $(x_i z_j)$ и множество D пар $(z_k y_h)$. Для задания вероятностного автомата надо определить для каждой пары из множества G вероятности b_{kh} перехода автомата в состояние Z_k и появления на выходе сигнала Y_h :

Элементы из D	$(z_1 y_1)$	$(z_2 y_2)$...	$(z_s y_{r-1})$	$(z_s y_r)$
$x_i z_j$	b_{11}	b_{12}	...	b_{sr-1}	b_{sr}

При этом $\sum_{k=1}^s \sum_{h=1}^r b_{kh} = 1.$

Число таких распределений, представленных в виде таблиц, равно числу элементов множества G . Обозначим множество таких распределений как B . Тогда совокупность множеств Z, X, Y, B определяет вероятностный автомат.

Это наиболее общий случай. Если распределения для нового состояния P -автомата и его выходного сигнала независимы, то это вероятностный автомат Мили. Для его задания надо определить множества распределений

элементы из Y	y_1	y_2	...	y_{r-1}	y_r
$x_i z_j$	q_1	q_2	...	q_{r-1}	q_r
элементы из Z	z_1	z_2	...	z_{s-1}	z_s
$x_i z_j$	p_1	p_2	...	p_{s-1}	p_s

Число таких распределений равно числу элементов множества G .

Говорят, что задан вероятностный автомат Мили, если заданы два распределения вероятностей

элементы из Y	$y_1 y_2 y_3 \dots y_Y$
$(x_i z_\xi)$	$q_1 q_2 q_3 \dots q_Y$

элементы из Z	$z_1 z_2 z_3 \dots z_k$
$(x_i z_\xi)$	$p_1 p_2 p_3 \dots p_k$

Где $\sum_{k=1}^Y q_k = 1, \sum_{k=1}^k p_k = 1$ при этом распределения для q и p

независимы.

Вероятностный автомат Мура имеет место, если определение выходного сигнала P -автомата зависит лишь от того состояния, в котором находится автомат в данном такте работы, а элементы из Z определяются как у автомата Мили.

Детерминированные автоматы это частный случай P -автомата. Также частными случаями являются Y – детерминированный и Z – детерминированные автоматы. Если выходной сигнал P - автомата определяется детерминировано, то такой автомат называется Y -детерминированным вероятностным автоматом. Аналогично, Z -детерминированным вероятностным автоматом называется P - автомат, у которого выбор нового состояния является детерминированным.

Марковский случайный процесс

Пусть имеется некоторая система, состояние которой может меняться с течением времени. Если состояние меняется случайным, непредсказуемым образом, будем говорить, что в системе протекает случайный процесс.

Случайный процесс называется процессом с дискретными состояниями, если возможные состояния системы:

$$S_1, S_2, S_3, \dots$$

можно перечислить (пронумеровать) одно за другим, а сам процесс состоит в том, что время от времени система скачком (мгновенно) переходит из одного состояния в другое.

Случайный процесс называется процессом с дискретным временем, если переходы из состояния в состояние возможны только в строго определенные моменты времени. В промежутках между этими моментами система сохраняет свое состояние.

Случайный процесс называется процессом с непрерывным временем, если переход системы из состояния в состояние возможен в любой наперед неизвестный, случайный момент.

Случайный процесс, протекающий в системе, называется *марковским процессом* (цепь Маркова) или процессом без последствия, если для каждого момента времени t_i вероятность любого последующего состояния системы зависит только от текущего состояния и не зависит от того, когда и каким путем система пришла в это состояние (т.е. от того, как развивался процесс в прошлом).

Иными словами, воздействие всей предыстории процесса на его будущее полностью сосредоточено в текущем значении процесса. Отсюда следует, что цепь Маркова должна обладать свойством отсутствия последствия. Это означает, что вероятность перехода в следующее состояние не должна зависеть от того, сколько времени процесс пребывал в текущем состоянии.

При моделировании систем, в которых случайные события, приводящие к изменению состояний, могут происходить только в моменты времени, отстоящие друг от друга на величину, кратную значению тактового интервала T (*дискретно – стохастические модели*), для описания интервалов времени между событиями используют регулярный просеянный поток. Его можно получить, удаляя в регулярном потоке события с вероятностью q и оставляя с вероятностью $1-q$. **Просеянный поток иногда называют дискретным пуассоновским**, так как его свойства аналогичны для моментов времени, кратных периоду T , свойствам простейшего потока. К просеянному регулярному потоку приводит, например регулярный поток данных, передаваемый по каналам связи с контролем наличия сбоев в передаваемом коде и исправлением путем повторной передачи.

Вероятность того, что величина интервала между событиями в просеянном потоке окажется равным i тактам:

$$P_i = q^{i-1} (1 - q)$$

Это выражение соответствует геометрическому распределению.

Математическое ожидание и среднее квадратическое отклонение для интервала времени между событиями в таком потоке равны соответственно:

$$m = \frac{T}{1 - q}$$

$$\sigma = m \sqrt{q} = \frac{T \sqrt{q}}{1 - q}$$

Если число состояний системы конечно и из каждого состояния можно перейти (за то или иное число шагов) в любое другое, то существуют предельные (финальные) вероятности состояний, которые не зависят от времени и начального состояния системы. Сумма предельных вероятностей всех состояний системы равна единице:

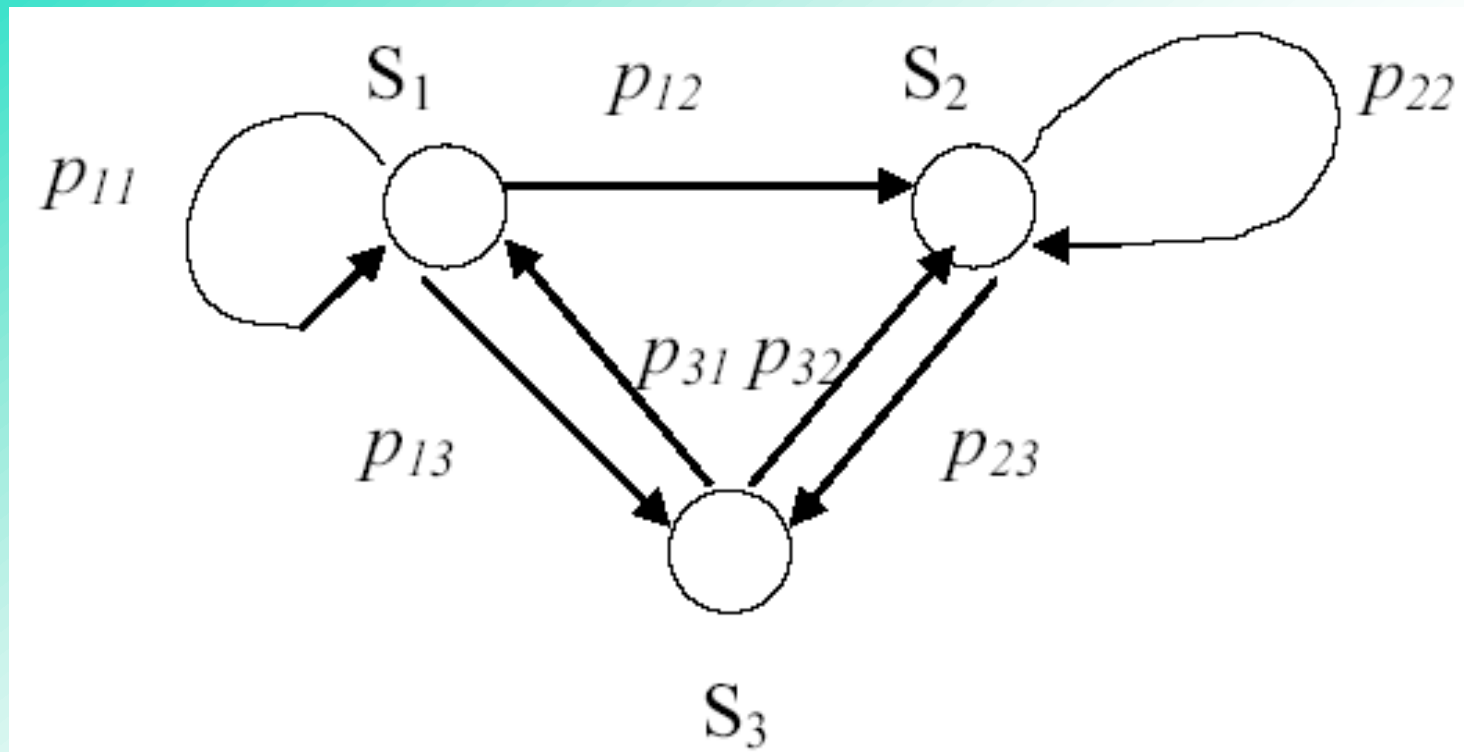
$$\sum_{i=1}^n P_i = 1.$$

Таким образом, при $t \rightarrow \infty$ в системе устанавливается некоторый предельный **стационарный режим**, который состоит в том, что система случайным образом меняет свои состояния, но вероятность каждого из них уже не зависит от времени. Эта вероятность представляет собой не что иное, как относительное время пребывания системы в данном состоянии.

При моделировании систем процесс их функционирования удобно представлять в виде графа, вершинами которого являются состояния S_i , а направленные дуги описывают переходы между состояниями. Если процесс является марковским и известны вероятности переходов из состояния в состояние, то вероятности состояний P_i могут быть найдены исходя из того, что вероятность любого состояния S_i равна сумме произведений вероятностей состояний S_j , из которых есть переход в данное состояние на вероятности этих переходов p_{ji} , т.е

$$P_i = \sum_j P_j p_{ji}$$

Рассмотрим на примере эту процедуру.



Система может находиться в одном из трех состояний: S_1 , S_2 или S_3 .

Систему уравнений для определения вероятностей состояний:

$$P_1 = p_{11}P_1 + p_{31}P_3 ;$$

$$P_2 = p_{12}P_1 + p_{32}P_3 + p_{22}P_2 ;$$

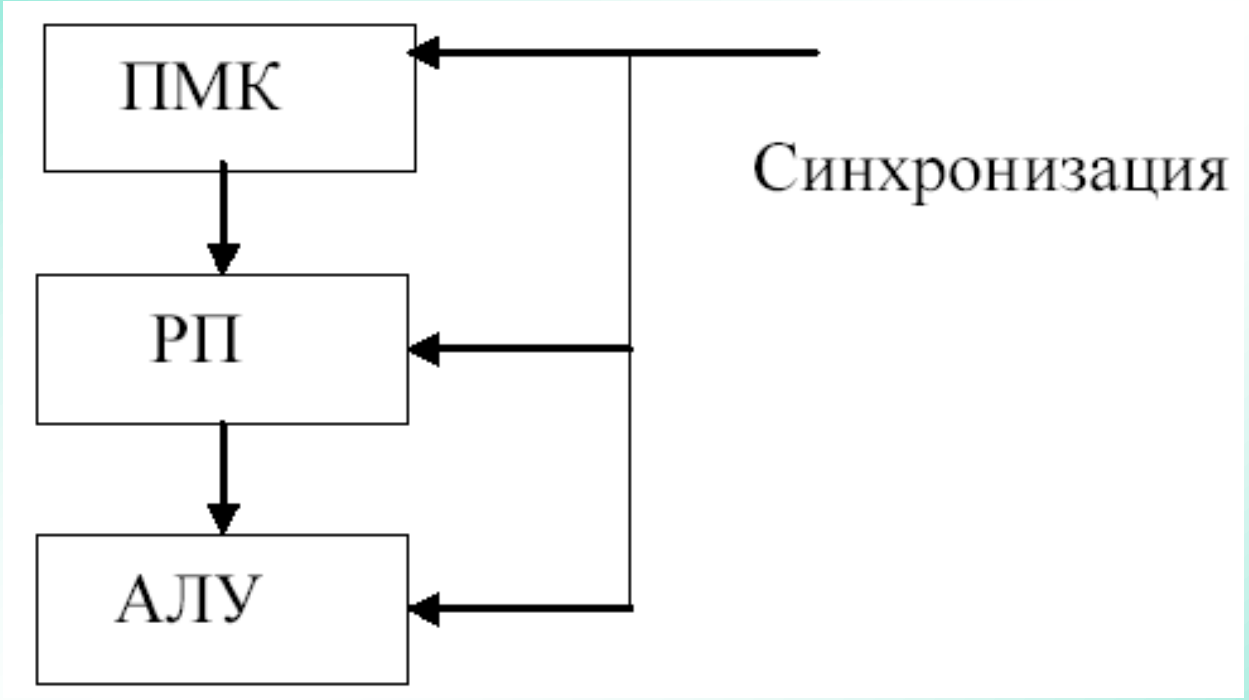
$$P_3 = p_{13}P_1 + p_{23}P_2 .$$

Попытка решить эту систему непосредственно неизбежно приведет к тождеству. Но решение существует и может быть получено, если воспользоваться нормировочным уравнением

$$P_1 + P_2 + P_3 = 1.$$

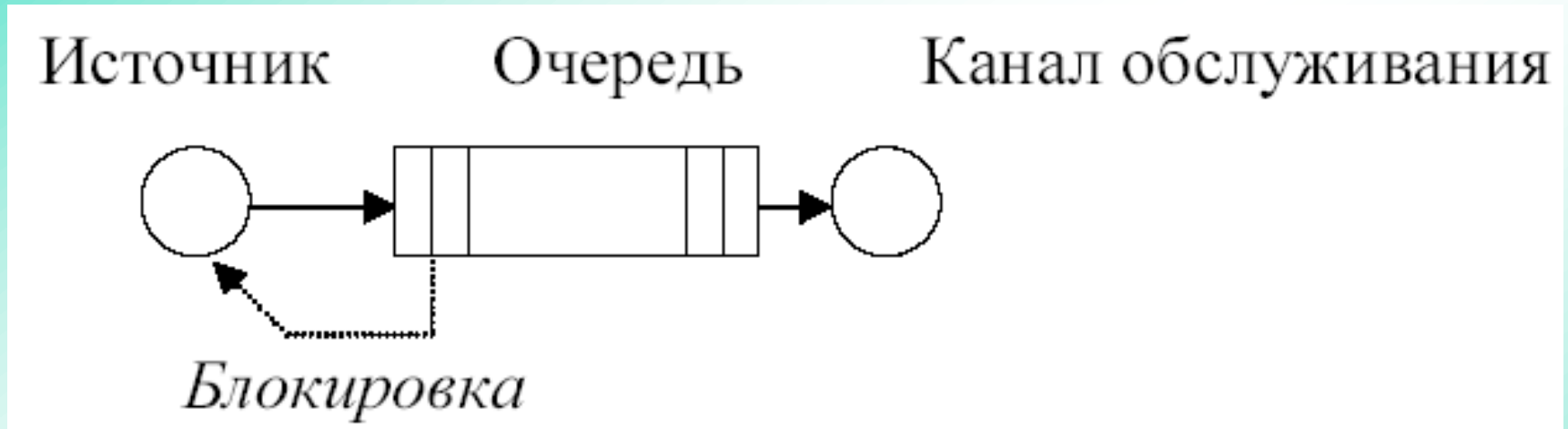
Если подставить его вместо одной из строк системы, то можно будет получить значение вероятностей состояний.

Рассмотрим пример исследования вычислительного узла с использованием аппарата Марковской дискретной цепи.



Из памяти макрокоманд (МК) по синхросигналу через каждые два такта в арифметико-логическое устройство (АЛУ) считываются макрокоманды. Если АЛУ готово принять макрокоманду, она обрабатывается, иначе макрокоманда загружается в регистровую память (РП) объемом N ячеек (мест ожидания) и ожидает освобождения АЛУ. При полном заполнении РП макрокоманда блокируется в памяти макрокоманд и процесс дальнейшей выборки приостанавливается. Интервалы времени обработки макрокоманд в АЛУ случайны и имеют геометрическое распределение с параметром π , т.е. с вероятностью $(1 - \pi)$ обработка макрокоманды в АЛУ по окончании очередного тактового интервала завершится, а с вероятностью π продлится еще на один интервал.

При построении и исследовании модели будем пользоваться представлением данного устройства как системы массового обслуживания (СМО).



Т.к. поток обслуживаний представляет собой просеянный регулярный поток, то процесс будет марковским, и мы можем определить финальные вероятности состояний этой системы.

Будем определять состояние системы трехкомпонентным вектором: jt_1t_2 .

Комбинаторная составляющая этого вектора j - количество заявок, находящихся в накопителе (длина очереди), $j = 0, 1, 2, \dots, n$.

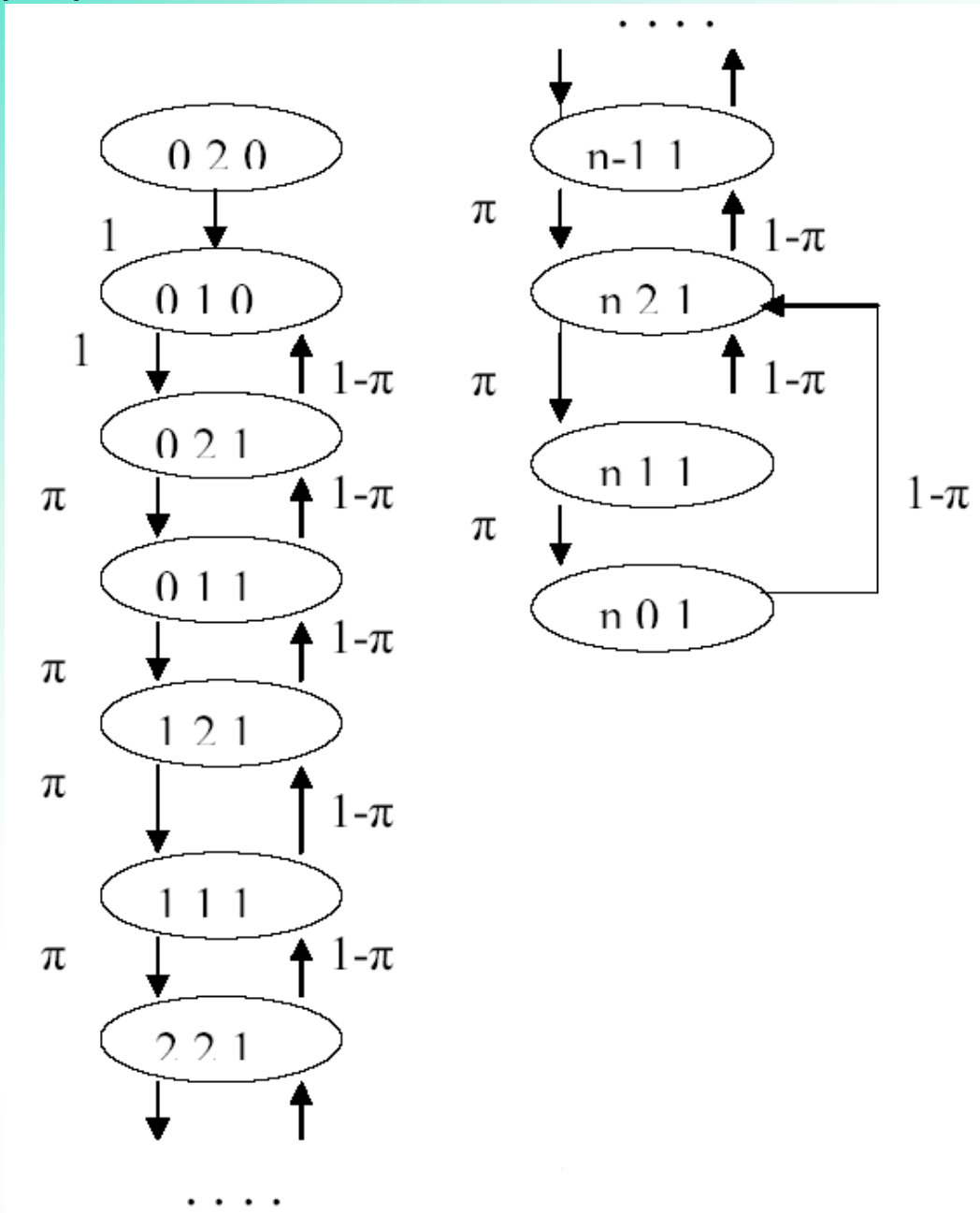
Временная составляющая t_1 - число тактов, оставшихся до появления заявки на выходе источника ($t_1 = 0, 1, 2$). Значение 0 означает, что источник заблокирован.

Составляющая t_2 определяет состояние канала обслуживания (АЛУ) и может принимать два значения:

$t_2 = 0$ - канал свободен;

$t_2 = 1$ - канал занят обслуживанием заявки.

Построим граф



система уравнений для стационарных (финальных) вероятностей состояний $P_{jt_1t_2}$.

В состояние P_{020} система больше не вернется, поэтому $P_{020} = 0$.

$$1. P_{010} = (1 - \pi)P_{021} + P_{020}$$

$$2. P_{021} = (1 - \pi)P_{011} + P_{010}$$

$$3. P_{i21} = \pi P_{i-111} + (1 - \pi)P_{i11} \quad (i = \overline{1, n-1})$$

$$4. P_{i11} = (1 - \pi)P_{i+121} + \pi P_{i21} \quad (i = \overline{0, n-1})$$

$$5. P_{n21} = \pi P_{n-111} + (1 - \pi)P_{n11} + (1 - \pi)P_{n01}$$

$$6. P_{n11} = \pi P_{n21}$$

$$7. P_{n01} = \pi P_{n11} + \pi P_{n01}$$

Выполнив преобразования получим, что сумма - это сумма геометрической прогрессии

$$p = \frac{2(1 - \pi) - 1}{2(1 - \pi) - \omega^{2n+2}}.$$

Используя полученное значение p (фактически, это вероятность простоя АЛУ) и рассчитав вероятности всех остальных состояний, можно найти другие интересующие нас характеристики системы.

а) Среднее время обслуживания заявки системой в целом (время пребывания заявки в системе) S .

$$S := \frac{T}{1 - \pi} \cdot \frac{1}{1 - P}$$

б) Интенсивность потока обработанных заявок (абсолютная пропускная способность):

$$\lambda = (1 - p)(1 - \pi) \frac{1}{T} ,$$

где $(1 - p)$ – вероятность того, что канал обрабатывал заявку, $(1 - \pi)$ – вероятность того, что обработка закончилась.

в) Средняя длина очереди

$$L_{оч} = \sum_j j P_{jt1t2} .$$

Непрерывно – стохастические модели (Q – схемы)

Особенностью непрерывно – стохастического подхода при моделировании систем и процессов является использование в качестве типовых математических схем систем массового обслуживания (англ. Queueing system).

Системы массового обслуживания представляют собой класс математических схем, разработанных в теории массового обслуживания и различных приложениях для формализации процессов функционирования систем, которые по своей сути являются процессами обслуживания.

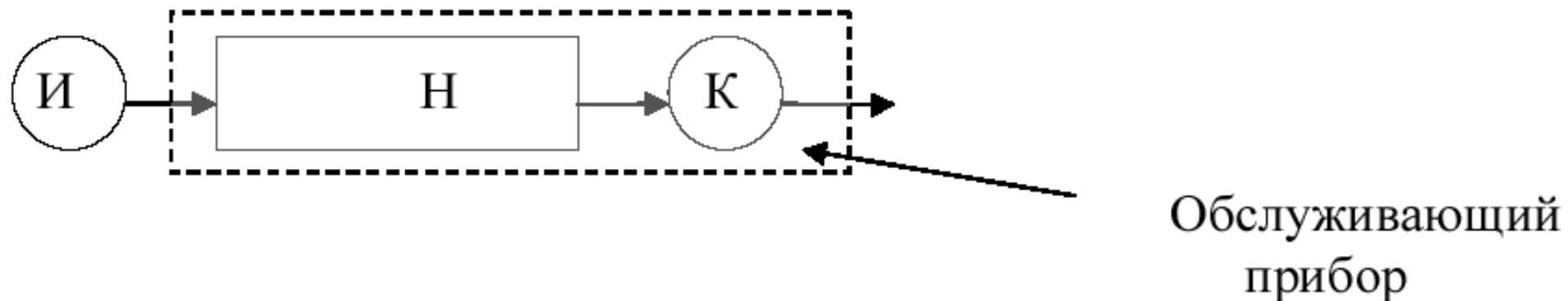
Системы массового обслуживания. Поток событий

Реальные системы могут быть представлены при моделировании как системы массового обслуживания (СМО), если при их функционировании можно выделить два процесса - поступление заявок на обслуживание и обслуживание заявок. Таким образом могут быть представлены различные по своей физической природе процессы – экономические, технические, производственные и т.п.

Для описания работы СМО используется понятие *поток событий* последовательность событий, происходящих одно за другим в некоторые моменты времени.

В СМО будем выделять три потока:

- входной поток: множество моментов времени поступления в систему заявок;
- поток обслуживаний: множество моментов времени окончания обработки системой заявок в предположении, что обслуживание осуществляется непрерывно;
- выходной поток: последовательность моментов времени ухода из системы обслуженных заявок.



И – источник заявок;
Н – накопитель (очередь);
К – канал обслуживания.

СМО состоит из какого-то числа обслуживающих единиц, которые называются **каналами обслуживания**.

В зависимости от количества каналов СМО могут быть одноканальными и многоканальными.

СМО могут быть двух типов.

1. *СМО с отказами*. В таких системах заявка, поступившая в момент, когда все каналы заняты, получает отказ и покидает систему не обслуженной.

2. *СМО с ожиданием*. Заявка, заставшая все каналы занятыми, становится в очередь и ожидает освобождения одного из каналов.

Число мест в очереди m может быть как неограниченным, так и ограниченным. В первом случае заявка может стать в очередь в любой момент, в то время как во втором она вынуждена покинуть систему, если очередь превышает заданную длину в момент прихода заявки или в процессе ожидания.

Порядок выборки заявок из очереди **определяется дисциплиной обслуживания.**

Некоторые наиболее употребляемые дисциплины:

1) FIFO (first in – first out) – в порядке поступления;

2) LIFO (last in – first out) – первой обслуживается заявка,

поступившая последней;

3) SIRO (service in random order) – обслуживание в случайном

порядке;

4) приоритетные системы – заявки обслуживаются в соответствии с

их приоритетами.

Наиболее часто на практике используются следующие правила.

1) Каналы занимаются в порядке их номеров. Канал с большим номером не может быть привлечен к обслуживанию заявки, если имеется свободный канал с меньшим номером.

2) Каналы занимаются в порядке очереди. Освободившийся канал поступает в очередь и не начинает обслуживания заявок до загрузки всех ранее освободившихся каналов.

3) Каналы занимаются в случайном порядке в соответствии с заданными вероятностями.

Реальный процесс функционирования системы массового обслуживания для удобства исследования можно представлять в виде последовательности отдельных актов (фаз) обслуживания, выполняемых различными устройствами. При этом, как правило, соблюдается такой порядок, при котором следующее устройство может приступить к обслуживанию заявки лишь тогда, когда работа предыдущего с данной заявкой полностью закончена.

В частном случае обслуживание может быть однофазным.

Характеристики СМО существенно зависят от вида и параметров входного потока и потока обслуживаний.

Поток событий называется *однородным*, если он характеризуется только моментами наступления событий и задается последовательностью $\{t_n\}$, где

$$\{t_n\} = \{0 \leq t_1 \leq t_2 \leq \dots \leq t_n \leq \dots\}.$$

Поток событий является *неоднородным*, если он задается последовательностью $\{t_n, f_n\}$, где t_n – моменты наступления событий, а f_n – набор признаков событий (приоритеты, принадлежность тому или иному источнику, тип канала для его обслуживания и т. д.).

Поток *регулярный*, если события поступают через равные промежутки времени.

Поток *нерегулярный*, если интервалы между событиями представляют собой случайные величины.

Если промежутки времени между последовательными событиями представляют собой независимые, одинаково распределенные случайные величины, **поток называется рекуррентным** (поток Пальма, поток с ограниченным последствием).

Для краткой характеристики СМО Д. Кендалл ввел символику (нотацию):

$$A/B/s/m/k$$

Где:

s - число обслуживающих приборов;

m – количество мест ожидания (если не указано, то считается, что $m=0$, т.е. это система с отказами); при неограниченной очереди в качестве m ставят символ ∞ ;

k – количество источников заявок. Если k отсутствует, то по умолчанию количество источников предполагается равным одному.

А и В характеризуют соответственно: поток требований и поток обслуживания, задавая функцию распределения интервалов между заявками во входном потоке и функцию распределения времен обслуживания.

А и В могут принимать значения:

D – детерминированное распределение;

M – показательное;

E_r – распределение Эрланга порядка r ;

H_r - гиперпоказательное;

G – распределение общего вида.

При этом подразумевается, что потоки являются рекуррентными, т.е. интервалы между событиями независимы и имеют одинаковое распределение.

Обязательными в этой нотации являются первые три позиции.

Простейший поток

Простейшим называется поток, обладающий следующими тремя свойствами: стационарность, ординарность и отсутствие последействия.

Поток является *стационарным*, если вероятность наступления заданного числа событий в течение интервала времени фиксированной длины зависит только от продолжительности интервала и не зависит от его расположения на временной оси. Иначе говоря, вероятностные характеристики и интенсивность такого потока со временем не изменяются.

Поток является *ординарным*, если вероятность появления двух или более событий в течение элементарного интервала времени $\Delta t \rightarrow 0$ есть величина бесконечно малая по сравнению с вероятностью появления одного события на этом интервале. Другими словами, два и более событий в таком потоке произойти одновременно не могут.

Поток называется потоком *без последействия*, если для любых неперекрывающихся интервалов времени число событий, попадающих на один из них, не зависит от числа событий, попадающих на другие. Иногда это свойство формулируют следующим образом: распределение времени до ближайшего события не зависит от времени наблюдения, т.е. от того, сколько времени прошло после последнего события. Отсутствие последействия в потоке означает, что события, образующие поток появляются в последовательные моменты времени независимо друг от друга.

Для простейшего потока число событий, попадающих на любой фиксированный интервал времени подчиняется закону Пуассона, поэтому его иначе **называют стационарным пуассоновским**.

Вероятность того, что за интервал времени τ в простейшем потоке произойдет ровно m событий

$$P_m(\tau) = \frac{(\lambda\tau)^m}{m!} e^{-\lambda\tau}$$

Где λ – интенсивность, т.е. среднее число событий в единицу времени.

Вероятность того, что в течении интервала времени τ не произойдет не одного события

$$P_0(\tau) = e^{-\lambda\tau} .$$

Вероятность того, что за это время произойдет хотя бы одно событие

$$P_{\geq 1}(\tau) = 1 - e^{-\lambda\tau}.$$

При построении и анализе непрерывно – стохастических моделей исследуются вероятности появления или не появления событий в течении элементарного интервала времени $\Delta t \rightarrow 0$.

Функция распределения вероятностей и функция плотности вероятностей имеют вид:

$$F(t) = 1 - e^{-\lambda t},$$

$$f(t) = \lambda e^{-\lambda t}.$$

Математическое ожидание и среднее квадратическое отклонение для T :

$$m_T = \sigma_T = \frac{1}{\lambda}$$

Одной из числовых характеристик для потоков является коэффициент вариации – отношение среднего квадратического к математическому ожиданию:

$$V = \frac{\sigma}{m}$$

Для реальных потоков значение этого коэффициента лежит в пределах от 0 до 1. Коэффициент вариации характеризует степень стохастичности, непредсказуемости потока.

Для регулярного потока с равными интервалами между событиями $V = 0$. Для простейшего потока $V = 1$, то есть этот поток является наиболее стохастичным среди всех потоков.

Простейший поток обладает следующими особенностями:

1. Сумма M независимых, ординарных, стационарных потоков заявок с интенсивностями λ_i ($i = 1, \dots, M$) сходится к простейшему потоку с интенсивностью, равной сумме интенсивностей исходных потоков при условии, что складываемые потоки оказывают приблизительно одинаково малое влияние на суммарный поток. Сходимость суммарного потока к простейшему осуществляется очень быстро. Практически можно считать, что сложение четырех-пяти стационарных, ординарных, независимых потоков, сравнимых по интенсивности, достаточно для того, чтобы суммарный поток был близок к простейшему.

Таким образом, для выяснения всех свойств суммарного потока достаточно знать лишь интенсивности суммируемых потоков и практически не требуется знать внутреннюю структуру этих потоков.

Простейший поток обладает устойчивостью, состоящей в том, что при суммировании независимых простейших потоков получается снова простейший поток, причем интенсивности складываемых потоков суммируются.

2. Поток заявок, полученный путем случайного разрежения исходного потока, когда каждая заявка с определенной вероятностью p исключается из потока независимо от того, исключены другие заявки или нет, образует простейший поток с интенсивностью

$$\lambda_p = p \lambda,$$

где λ — интенсивность исходного потока. В отношении исходного потока заявок делается предположение лишь об ординарности и стационарности.

3. Для простейшего потока характерно, что поступление заявок через короткие промежутки времени более вероятно, чем через длинные,— 63% промежутков времени между заявками имеют длину, меньшую среднего периода $1/\lambda$.

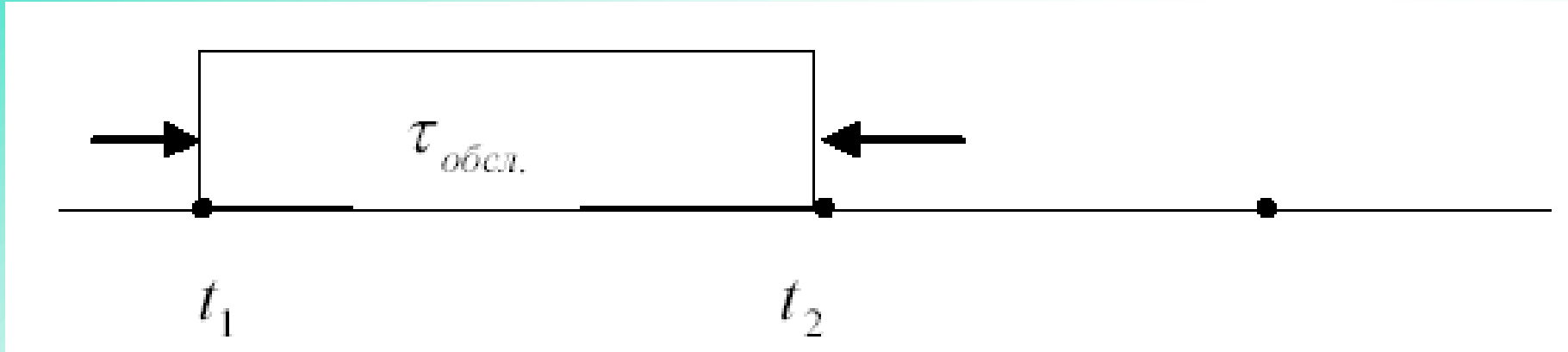
Следствием этого является то, что простейший поток по сравнению с другими видами потоков создает наиболее тяжелый режим работы системы. Поэтому предположение о том, что на вход системы поступает простейший поток заявок, приводит к определению предельных значений характеристик качества обслуживания. Если реальный поток отличен от простейшего, то система будет функционировать не хуже, чем это следует из полученных оценок.

4. Интервал времени между произвольным моментом времени и моментом поступления очередной заявки имеет такое же распределение с тем же средним $M[\tau] = 1/\lambda$, что и интервал времени между двумя последовательными заявками.

Эта особенность простейшего потока является следствием отсутствия последствия.

Выходные потоки обычно имеют последствие, даже если входные потоки им не обладают.

Пример СМО с фиксированным временем обслуживания $\tau_{\text{обсл}}$



Если в момент t_1 на выходе появилась обслуженная заявка, то на интервале $(t_1, t_1 + \tau_{\text{обсл}})$ заявка не появится.

Непрерывные марковские цепи

Уравнения Колмогорова