



СПбГЭТУ «ЛЭТИ»
ПЕРВЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ



В. В. Романцев

Анализ, моделирование и оптимизация систем

Конспект лекций

СПбГЭТУ «ЛЭТИ», 2021 г.





1 СИСТЕМНЫЕ ИССЛЕДОВАНИЯ КАК СОСТАВНАЯ ЧАСТЬ ОБЩЕЙ ТЕОРИИ СИСТЕМ

1.1 Основы теории систем

В научных исследованиях различных областей техники, экономики, социальных явлений и др. мы сталкиваемся с совокупностью объектов, которые называют сложными системами. Проблемы, возникающие при исследовании сложных систем, относятся к широко известной и чрезвычайно обширной в приложениях междисциплинарной ветви науки - системному анализу (СА).

Дадим определение термину система. Элементом назовем объект (материальный, энергетический, информационный) и будем обозначать через z , а всю их рассматриваемую совокупность - $[Z]$ или Z . Связью назовем важный для целей рассмотрения обмен между элементами, веществом, энергией, информацией. Обозначив связи между элементами z_1 и z_2 через e_{12} и e_{21} , можно изобразить систему в виде графа (см. рис. 1.1).

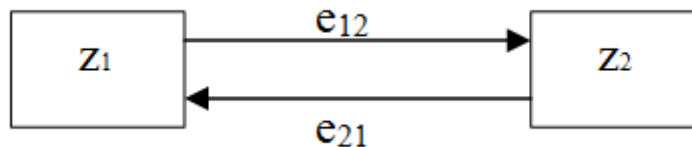


Рисунок 1.1 - Граф связей между двумя элементами

Системой назовем совокупность элементов, обладающих следующими признаками:

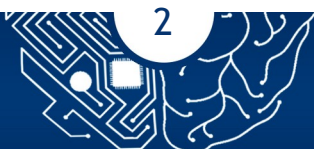
- связями, которые позволяют соединить два любых элемента совокупности;
- свойством (функцией) отличным от свойств отдельных элементов совокупности.

Первый признак назовем связностью системы $e \in E$, второй - ее эмерджентной функцией F . Тогда описание системы S будет иметь вид:

$$S = [Z, E, F] \quad (1)$$

Выражение (1) является наиболее простым описанием системы. Рассмотрим еще одно описание системы, основанное на явном учете динамических свойств объектов, входящих в систему.

Введем обозначения:



- $u(t)$ - входное воздействие, $u \in U$;
- x - состояния, $x \in X$;
- y - выходная величина, $y(t) \in Y$;
- φ - переходная функция - траектория системы;
- $x(t) = \varphi(t, x(\tau_0), u(t))$, τ_0 - начальный момент времени;
- η - выходное отображение, $y(t) = \eta(t, x(t))$.

Таким образом, описание системы в этом случае имеет вид:

$$S = (T, X, U, y, \varphi, \eta) \quad (2)$$

В зависимости от типа системы и целей анализа удобно использовать описание (1) или (2).

Так системы бывают открытые и замкнутые. Примеры открытых систем:

- системы, не полностью изолированные от окружающей среды;
- системы, реагирующие на внешние изменения опосредованно;
- системы с двусторонним взаимодействием.

Большинство систем относится к классу открытых. Замкнутые системы это, как правило, некоторая абстракция. Такое упрощение открытой системы целесообразно при проведении упрощенного анализа.

Введем понятие структуры системы. Итак, под структурой будем понимать совокупность элементов с указанием связей между ними, которая дает представление о системе в целом. Структура может быть охарактеризована по имеющимся или преобладающим в ней типам связей, которые могут быть последовательные, параллельные и обратные (см. рис.1.2).

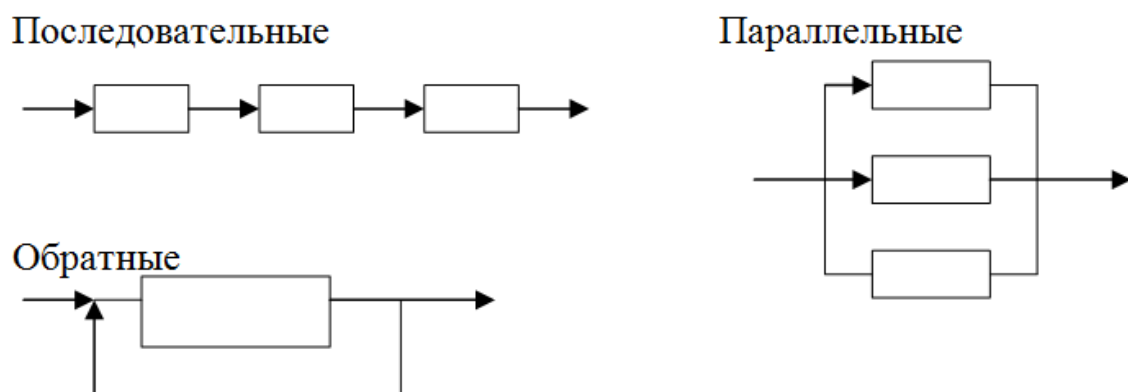


Рисунок 1.2 - Типы связей системы

Введем еще одно понятие в теории систем - это декомпозиция, то есть деление системы на части с целью упрощения сложной системы для рассмотрения ее целиком. Декомпозиция приводит к замене исходной

системы на некоторую другую - упрощенную или полностью соответствует исходной системе. Такая декомпозиция является строгой.

Иерархией назовём структуру с наличием подчинённости связей между элементами. Виды иерархических структур различны. Например, древовидная, ромбовидная, кольцевая (см. рис.1.3).

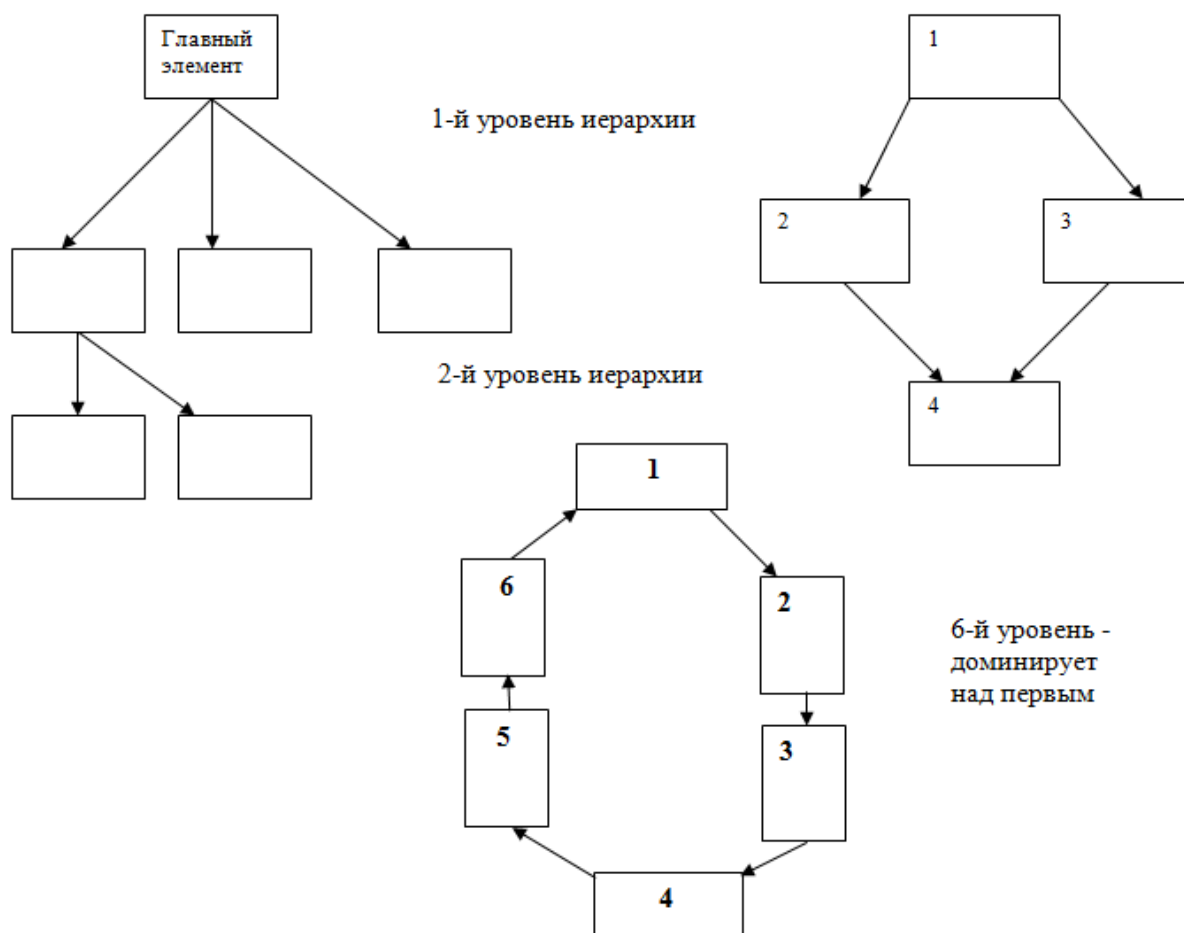


Рисунок 1.3 - Виды иерархических структур

Существует ещё один важный класс систем, обладающих свойством целенаправленности, т.е. систем с некоторой сформулированной целью. Под целью будем понимать задачу получения желаемого выходного воздействия или достижения желаемого состояния системы. Постановка цели перед системой, так называемой глобальной цели, приводит к необходимости формулировки локальных целей, стоящих перед элементами системы или группами элементов, или целенаправленного вмешательства в функционирование системы. Обе эти операции тесно связаны, хотя на практике, обычно, сначала разбивают глобальную цель на набор локальных, а потом ищут пути достижения локальных целей.



1.2 Основные свойства систем

При помощи описаний (1) и (2) можно проанализировать основные свойства систем.

Систему называют стационарной, если её реакция на входное воздействие не зависит от момента времени, в который это воздействие прилагается.

Различают системы с непрерывным или дискретным временем.

Рассмотрим влияние структуры пространства событий X на свойство и тип систем. Например, X - конечномерное линейное пространство, тогда S - конечномерная система. Если X - конечно, то S - конечно. Если X - конечно, U , Y также конечны, система стационарна и время дискретно, то S - конечный автомат.

Системы бывают линейными и нелинейными. Система S линейна, если является линейным отображением, то есть переходная функция φ является линейной при любых t и τ_0 , то есть $x(t) = \varphi(t, x(\tau_0), U(t))$ - линейная.

Системы бывают детерминированными или стохастическими.

Рассмотрим два очень важных свойства системы:

- управляемость
- наблюдаемость.

Управляемость - событие в системе (2) (τ_0, x) управляемо, если для $t \geq \tau_0$, можно перевести систему из (τ_0, x) в $(t, 0)$. Система (2) полностью управляема в τ_0 , если каждое событие (τ_0, x) , $x \in X$, $\tau_0 \in T$ является управляемым. Свойство управляемости обеспечивает возможность перестройки системы, т.е. перевода ее из состояния x в состояние 0.

Наблюдаемость - событие в системе (2) называется наблюдаемым, если для $t > \tau_0$, по результатам наблюдения $y(t)$ возможно восстановить состояние x в момент времени τ_0 . Соответственно система (2) полностью наблюдаема, если каждое событие (τ_0, x) является наблюдаемым. Свойство наблюдаемости необходимо для установления первопричины поведения системы по данным о поведении системы $y(t)$ в момент времени t .

Еще одно важное свойство системы - воспроизводимость - то есть способность воспроизводить на выходе заданные выходные величины $y(t) \in Y$. Понятие воспроизводимости приобретает глубокий смысл только тогда,





когда оно обеспечивает воспроизведение всех достаточно близких значений выходных откликов $y(t) \in Y$.

1.3 Модульное строение систем

Система может представляться набором модулей и сама рассматриваться как модуль. Модульное построение системы, как правило, определяет ее декомпозицию. В ряде случаев оно определяет и структуру системы.

Деление системы на модули - это удобный и наиболее распространенный прием работы с системами, включая этапы их создания, проверки, настройки, модификации. Именно модульное строение системы в сочетании с принципом введения все более крупных модулей позволяет рассматривать сколь угодно сложные системы. Примерами реализации этого положения на практике являются создание из множества элементов ЭВМ последних поколений, информационных систем и вычислительных сетей, систем организационного управления. Разработка таких систем обычно ведется «сверху - вниз», то есть с продумывания назначения и разработки описания спецификаций входов и выходов модулей верхнего уровня и далее вниз, все в большей степени детализируя описание системы.

2 СИСТЕМНЫЕ ОБЪЕКТЫ И ИХ ОБОБЩЕННАЯ ХАРАКТЕРИСТИКА

2.1 Принципы системного подхода

Системный анализ предполагает рассмотрение сложной системы как единого целого с одной стороны, и как состоящей из отдельных частей - с другой стороны. Такой противоречивый подход к анализу системы приводит к различным конфликтам. С целью их устранения, точнее недопущения, сформулируем основные положения, которые принято называть **принципами системного подхода**, обобщающими опыт анализа сложных систем.

В таблице приведены основные принципы системного подхода и их содержание.

Принципы системного подхода	Содержание
Конечности цели	Абсолютный приоритет конечной (глобальной) цели





Единства и связи	Совместное рассмотрение системы как целого, и как совокупности частей
Модульного построения	Выделение модулей в системе и рассмотрение системы как совокупности модулей
Иерархии	Введение иерархии частей или порядка анализа системы (ранжирования)
Функциональности	Совместное рассмотрение структуры и функции с приоритетом функции над структурой
Развития	Учет изменяемости системы, ее способность к развитию, замене частей, накоплению информации
Децентрализация	Сочетание централизации и децентрализации в принимаемых решениях и управлении
Неопределенности	Учет неопределенностей и случайностей в системе

Рассмотрим введенные принципы более подробно.

Первый из них означает, что в целенаправленной системе все должно быть подчинено глобальной цели. Любая попытка изменения, модификации и управления в такой системе должна оцениваться с точки зрения того, помогает или мешает оно достижению конечной цели. Это накладывает особую ответственность на выбор цели и ее четкую трактовку. Расплывчатые, не полностью определенные конечные цели влекут за собой неясность в структуре и управлении системой и, как следствие, неверные действия.

Если системы не являются целенаправленными, то понятие конечной цели заменяют понятием основной функции.

Составляющие формулировки второго принципа обладают довольно тесной взаимосвязью и объединяются в один принцип единства и связи. Однако в ряде случаев их полезно рассматривать отдельно. Так принцип единства - это ориентация на «взгляд вовнутрь» системы, а принцип связи - на «взгляд изнутри». На разных этапах исследований полезна либо одна, либо другая ориентация.

Принцип модульного построения указывает на возможности рассмотрения системы как совокупности входных и выходных воздействий и





величин. Он утверждает полезность абстрагирования от излишней детализации при сохранении возможности адекватно описывать системы.

Принцип иерархии подталкивает исследователя к отысканию и созданию в системе иерархического характера связей между элементами, модулями, целями. Поясним смысл слова «ранжирование» в формулировке принципа. Иерархические системы обычно исследуются и создаются «сверху-вниз». В случае отсутствия иерархии нужно знать, в каком порядке будут рассматриваться части системы. Таким образом, вводится порядок рассмотрения и анализа системы, который и называется ранжированием. Ранжирование применимо и в сочетании с иерархией системы, например, для внесения очередности рассмотрения модулей одного и того же уровня.

Поскольку можно определить функцию системы как ее свойства, необходимые для реализации целей системы, смысл и назначение этого принципа системного подхода не подлежит подробному обсуждению, так как функция системы - это один из самых главных компонент ее описания. Из утверждения принципа, что любая структура тесно связана с функцией системы, следует, что исследовать (создавать) структуру необходимо только после выяснения функции системы. Так, перестройка производства, связанная с введением автоматизации, ведет к созданию новых структурных подразделений. Следовательно, при введении в систему новых функций полезно пересмотреть ее структуру, а не пытаться втиснуться в рамки старой.

Принципы развития, децентрализации и неопределенности хорошо раскрыты в их формулировках.

2.2 Стадии системных исследований

Рассмотрим общую методику проведения системных исследований, представленную в виде стадий системного исследования. Отметим, что выделенные стадии носят общий характер и в конкретных условиях могут отличаться, удаляться и модифицироваться.

1 этап. Формирование общих представлений о системе.

1. Выделение главных функций (свойств, целей) системы.
2. Выявление основных частей (модулей) в системе и их функций.





3. Выявление основных процессов, протекающих в системе, их роли, условий существования; выявление скачков, смен состояний, основных управляющих факторов.

4. Выявление основных внешних воздействий.

5. Выявление неопределенностей и случайностей, возникающих в системе.

На стадии 2 следует обратить внимание на так называемые системообразующие факторы, т.е. на те связи и взаимообусловленности, которые делают объект исследований системой.

2 этап. Формирование углубленных представлений о системе.

6. Выявление разветвленной структуры, иерархии, формирование представлений о системе как о совокупности модулей, связанных входами и выходами.

7. Выявление и ранжирование элементов и связей в системе.

8. Учет изменений и неопределенностей в системе.

9. Исследование функций и процессов в системе с целью управления ими. Введение управления и процедур принятия решений.

Стадии 6 и 7 тесно связаны друг с другом. Важно на этапе 7 еще раз обратить внимание на системообразующие факторы.

3 этап: Моделирование системы как этап исследования.

10 - 14. Создание описания системы. Компромиссы «точность - сложность», «баланс точностей».

4 этап: Сопровождение системы.

15. Накопление опыта работы с системой и ее моделью.

16. Оценка предельных возможностей системы.

17. Расширение функций (свойств) системы.

5 этап: Особенности создания новой системы.

Процесс создания новой системы характеризуется многократным выполнением стадий 1 - 17, то есть имеет итеративный характер.

2.3 Проектирование систем

Увеличение производительности ЭВМ привело к осуществлению больших программных продуктов. При этом возникли проблемы





коллективной разработки ПО. Многие проблемы были решены при применении системного анализа в разработке больших программ.

Сложную программу можно разделить на ряд подпрограммных модулей (подпрограмм), соединенных входами и выходами. Входные и выходные параметры можно рассматривать как механизм сопряжения подпрограмм.

Опишем процесс декомпозиции. Программу представим тройкой

$$P = \langle F, I, J \rangle \text{ где}$$

F - функция, реализуемая программой;

I - множество входных параметров;

J - множество выходных параметров.

Пусть $P_i^{(1)} = \langle F_i^{(1)}, I_i^{(1)}, J_i^{(1)} \rangle$, где $I_i^{(1)} \in I$, $J_i^{(1)} \in J$, $\bigcup_i I_i^{(1)} = I$, $\bigcup_i J_i^{(1)} = J$, подпрограммы $P_i^{(1)}$ образуют первый уровень декомпозиции программ P . Тогда наборы программ второго уровня декомпозиции программ P получим в виде $P_{ij}^{(2)} = \langle F_{ij}^{(2)}, I_{ij}^{(2)}, J_{ij}^{(2)} \rangle$, где $I_{ij}^{(2)} \in I_i^{(1)}$, $J_{ij}^{(2)} \in J_i^{(1)}$, $\bigcup_{ij} I_{ij}^{(2)} = I_i^{(1)}$, $\bigcup_{ij} J_{ij}^{(2)} = J_i^{(1)}$ и т.д.

Таким образом, выделяя программные модули и поручая их разработку различным исследователям, можно сократить время проектирования больших программ (t_{np}). Однако, как известно, сокращение времени разработки программы не является линейной функцией от числа исполнителей, но и не является монотонной функцией. Так, качественный вид зависимости $t_{np} = \Phi(N)$ следующий (см. рис.2.1).

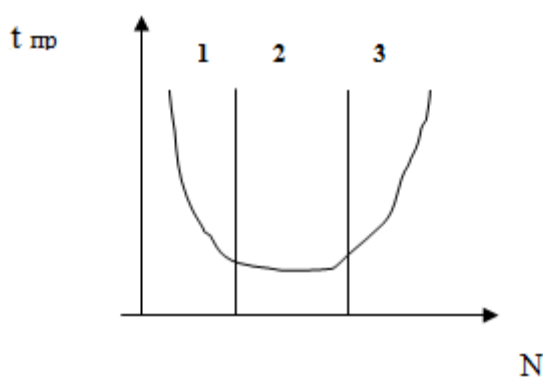


Рисунок 2.1 - Вид зависимости $t_{np} = \Phi(N)$

Здесь можно выделить три участка. Первый характеризуется почти линейным уменьшением - t_{np} от N , затем (участок 2) - время падает медленно и достигает минимума, а потом начинает расти. Такой характер зависимости $t_{np}(N)$ объясняется тем, что, как известно, во-первых, время программиста тратится не только непосредственно на разработку





программы, но и на уточнение характера и вида сопряжений программы (интерфейса) с другими программами верхнего и нижнего уровней. Во-вторых, непосредственно на декомпозицию программы требуется весьма значительное время, особенно если она является сложной. Таким образом можно сформулировать оптимизационную задачу: найти такую иерархическую структуру программы S' , которая позволяет минимизировать общее время разработки программы, при наличии ограничений на ресурсы и достичь заданные качества то есть

$$T_{np}(S') = \min_{S \in \Sigma} T(S)$$

Такая постановка не реализуема и поэтому чаще рассматривается следующая задача:

$T_{np}(S) > T_z$, где T_z - заданное время создание сложной программы.

Особенности применения системного анализа при создании сложных программ связаны с понятием программного продукта и жизненным циклом программного продукта.

В жизненном цикле ПО выделяют шесть основных этапов (см. рис.2.2):

- 1 Анализ требований предъявляемых в системе
- 2 Определение спецификации
- 3 Проектирование
- 4 Кодирование
- 5 Тестирование
- 6 Эксплуатация и сопровождение



Рисунок 2.2 - Жизненный цикл ПО





3 ОСНОВНЫЕ ФУНКЦИИ МОДЕЛЕЙ И ИХ КЛАССИФИКАЦИЯ

3.1 Моделирование как метод научного познания.

В настоящее время нельзя назвать область человеческой деятельности, в которой в той или иной степени не использовались бы методы моделирования. Особенно это относится к сфере управления различными системами, где основными являются процессы принятия решений на основе получаемой информации.

В научных исследованиях большую роль играют гипотезы, т.е. определенные предсказания, основывающиеся на небольшом количестве опытных данных, наблюдений, догадок. Быстрая и полная проверка выдвигаемых гипотез может быть проведена в ходе специально поставленного эксперимента. При формулировании и проверке правильности гипотез большое значение имеет метод аналогий (суждений о каком-либо частном сходстве двух объектов). Современная научная гипотеза создается, как правило, по аналогии с проверенными на практике научными положениями. Таким образом, аналогия связывает гипотезу с экспериментом.

Гипотезы и аналогии, отражающие реальный, объективно существующий мир, должны обладать наглядностью или сводиться к удобным для исследования логическим схемам; такие логические схемы, упрощающие рассуждения и логические построения или позволяющие проводить эксперименты, уточняющие природу явлений, называются моделями. Другими словами, модель (лат. *modulus* — мера) — это объект-заместитель объекта-оригинала, обеспечивающий изучение некоторых свойств оригинала.

Замещение одного объекта другим с целью получения информации о важнейших свойствах объекта-оригинала с помощью объекта-модели называется моделированием. Таким образом, моделирование может быть определено как представление объекта моделью для получения информации об этом объекте путем проведения экспериментов с его моделью. Теория замещения одних объектов (оригиналов) другими объектами (моделями) и исследования свойств объектов на их моделях называется теорией моделирования.





Обобщенно моделирование можно определить как метод опосредованного познания, при котором изучаемый объект-оригинал находится в некотором соответствии с другим объектом - моделью, причем модель способна в том или ином отношении замещать оригинал на некоторых стадиях познавательного процесса. Стадии познания, на которых происходит такая замена, а также формы соответствия модели и оригинала могут быть различными:

1) моделирование как познавательный процесс, содержащий переработку информации, поступающей из внешней среды, о происходящих в ней явлениях, в результате чего в сознании появляются образы, соответствующие объектам;

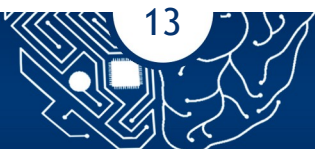
2) моделирование, заключающееся в построении некоторой системы-модели (второй системы), связанной определенными соотношениями подобия с системой-оригиналом (первой системой), причем в этом случае отображение одной системы в другую является средством выявления зависимостей между двумя системами, отраженными в соотношениях подобия, а не результатом непосредственного изучения поступающей информации.

Процесс моделирования предполагает наличие объекта исследования; исследователя, перед которым поставлена конкретная задача; модели, создаваемой для получения информации об объекте и необходимой для решения поставленной задачи. Причем по отношению к модели исследователь является, по сути дела, экспериментатором, только в данном случае эксперимент проводится не с реальным объектом, а с его моделью.

3.2 Характеристика проблемы моделирования систем.

Одновременно с развитием теоретических методов анализа и синтеза совершенствуются и методы экспериментального изучения реальных объектов, появляются новые средства исследования. Однако эксперимент был и остается одним из основных и существенных инструментов познания.

Подобие и моделирование позволяют по-новому описать реальный процесс и упростить экспериментальное его изучение. Совершенствуется и само понятие моделирования.





Познание реальной действительности является длительным и сложным процессом. Определение качества функционирования большой системы, выбор оптимальной структуры и алгоритмов поведения, построение системы S в соответствии с поставленной перед нею целью – основная проблема при проектировании современных систем, поэтому моделирование можно рассматривать как один из методов, используемых при проектировании и исследовании больших систем.

Моделирование базируется на некоторой аналогии реального и мысленного эксперимента. Аналогия – основа для объяснения изучаемого явления, однако критерием истины может служить только практика, только опыт. Хотя современные научные гипотезы могут создаваться чисто теоретическим путем, но, по сути, базируются на широких практических знаниях. Для объяснения реальных процессов выдвигаются гипотезы, для подтверждения которых ставится эксперимент либо проводятся такие теоретические рассуждения, которые логически подтверждают их правильность.

Различают пассивный эксперимент, когда исследователь наблюдает протекающий процесс, и активный, когда наблюдатель вмешивается и организует протекание процесса.

В основе моделирования лежат информационные процессы, поскольку само создание модели M базируется на информации о реальном объекте. В процессе реализации модели получается информация о данном объекте, одновременно в процессе эксперимента с моделью вводится управляющая информация, существенное место занимает обработка полученных результатов, т. е. информация лежит в основе всего процесса моделирования.

3.3 Классификация

В зависимости от характера изучаемых процессов в системе S все виды моделирования могут быть разделены на детерминированные и стохастические, статические и динамические, дискретные, непрерывные и дискретно-непрерывные. Детерминированное моделирование отображает детерминированные процессы, т. е. процессы, в которых предполагается отсутствие всяких случайных воздействий; стохастическое моделирование





отображает вероятностные процессы и события. В этом случае анализируется ряд реализаций случайного процесса и оцениваются средние характеристики, т. е. набор однородных реализаций. Статическое моделирование служит для описания поведения объекта в какой-либо момент времени, а динамическое моделирование отражает поведение объекта во времени. Дискретное моделирование служит для описания процессов, которые предполагаются дискретными, соответственно непрерывное моделирование позволяет отразить непрерывные процессы в системах, а дискретно-непрерывное моделирование используется для случаев, когда хотят выделить наличие как дискретных, так и непрерывных процессов.

Математическое моделирование. Для исследования характеристик процесса функционирования любой системы математическими методами, включая и машинные, должна быть проведена формализация этого процесса, т. е. построена математическая модель.

Под математическим моделированием будем понимать процесс установления соответствия данному реальному объекту некоторого математического объекта, называемого математической моделью, и исследование этой модели, позволяющее получать характеристики рассматриваемого реального объекта. Вид математической модели зависит как от природы реального объекта, так и задач исследования объекта и требуемой достоверности и точности решения этой задачи. Любая математическая модель, как и всякая другая, описывает реальный объект лишь с некоторой степенью приближения к действительности. Математическое моделирование для исследования характеристик процесса функционирования систем можно разделить на аналитическое, имитационное и комбинированное.

Для аналитического моделирования характерно то, что процессы функционирования элементов системы записываются в виде некоторых функциональных соотношений (алгебраических, интегро-дифференциальных, конечно-разностных и т. п. или логических условий).

Аналитическая модель может быть исследована следующими методами: а) аналитическим, когда стремятся получить в общем виде явные зависимости для искомых характеристик; б) численным, когда, не умея





решать уравнений в общем виде, стремятся получить числовые результаты при конкретных начальных данных; в) качественным, когда, не имея решения в явном виде, можно найти некоторые свойства решения (например, оценить устойчивость решения).

Численный метод позволяет исследовать по сравнению с аналитическим методом более широкий класс систем, но при этом полученные решения носят частный характер. Численный метод особенно эффективен при использовании ЭВМ.

В настоящее время распространены методы машинной реализации исследования характеристик процесса функционирования больших систем. Для реализации математической модели на ЭВМ необходимо построить соответствующий моделирующий алгоритм.

При имитационном моделировании реализующий модель алгоритм воспроизводит процесс функционирования системы S во времени, причем имитируются элементарные явления, составляющие процесс, с сохранением их логической структуры и последовательности протекания во времени, что позволяет по исходным данным получить сведения о состояниях процесса в определенные моменты времени, дающие возможность оценить характеристики системы S .

Основным преимуществом имитационного моделирования по сравнению с аналитическим является возможность решения более сложных задач. Имитационные модели позволяют достаточно просто учитывать такие факторы, как наличие дискретных и непрерывных элементов, нелинейные характеристики элементов системы, многочисленные случайные воздействия и др., которые часто создают трудности при аналитических исследованиях.

Когда результаты, полученные при воспроизведении на имитационной модели процесса функционирования системы S , являются реализациями случайных величин и функций, тогда для нахождения характеристик процесса требуется его многократное воспроизведение с последующей статистической обработкой информации. Целесообразно в качестве метода машинной реализации имитационной модели использовать метод статистического моделирования. Методом статистического моделирования будем в дальнейшем называть метод машинной реализации имитационной





модели, а методом статистических испытаний (Монте-Карло) – численный метод решения аналитической задачи.

Комбинированное (аналитико-имитационное) моделирование при анализе и синтезе систем позволяет объединить достоинства аналитического и имитационного моделирования. При построении комбинированных моделей проводится предварительная декомпозиция процесса функционирования объекта на составляющие подпроцессы и для тех из них, где это возможно, используются аналитические модели, а для остальных подпроцессов строятся имитационные модели. Такой комбинированный подход позволяет охватить качественно новые классы систем, которые не могут быть исследованы с использованием только аналитического и имитационного моделирования в отдельности.

4 МЕТОДЫ ИССЛЕДОВАНИЯ МОДЕЛЕЙ И ТЕХНОЛОГИЯ МАШИННОГО МОДЕЛИРОВАНИЯ

4.1 Методы исследования моделей

При исследовании больших систем всё чаще применяются методы имитационного моделирования. Имитационная система реализуется на ЭВМ и позволяет исследовать имитационную модель M , задаваемую в виде определенной совокупности отдельных блочных моделей и связей между ними в их взаимодействии в пространстве и времени при реализации какого-либо процесса. Можно выделить три основные группы блоков: блоки, характеризующие моделируемый процесс функционирования системы S ; блоки, отображающие внешнюю среду E и ее воздействие на реализуемый процесс; блоки, играющие служебную вспомогательную роль, обеспечивая взаимодействие первых двух, а также выполняющие дополнительные функции по получению и обработке результатов моделирования. Кроме того, имитационная система характеризуется набором переменных, с помощью которых удастся управлять изучаемым процессом, и набором начальных условий, когда можно изменять условия проведения машинного эксперимента.

Таким образом, имитационная система есть средство проведения машинного эксперимента, причем эксперимент может ставиться многократно, заранее планироваться, могут определяться условия его



проведения. Необходимо при этом выбрать методику оценки адекватности получаемых результатов и автоматизировать как процессы получения, так и процессы обработки результатов в ходе машинного эксперимента.

Имитационная система может рассматриваться как машинный аналог сложного реального процесса. Позволяет заменить эксперимент с реальным процессом функционирования системы экспериментом с математической моделью этого процесса в ЭВМ. В настоящее время имитационные эксперименты широко используют в практике проектирования сложных систем, когда реальный эксперимент невозможен.

4.2 Последовательность разработки и машинной реализации моделей 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. 4.1.

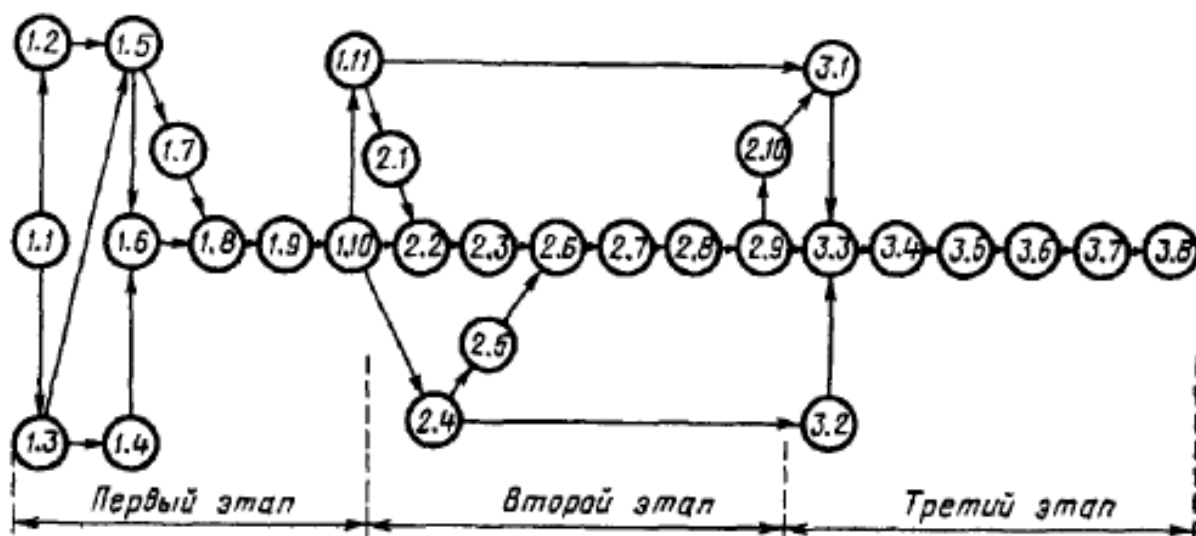


Рисунок 4.1 - Взаимосвязь этапов моделирования

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; 3.1 – плакирование 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_x 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_M 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.

4.3 Построение концептуальной модели системы и ее формализация.

На первом этапе машинного моделирования — построения концептуальной модели M_x 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.

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 .

Рассмотрим более подробно основные подэтапы построения концептуальной модели M_x системы и ее формализации.

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.

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. 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.

1.4. Выдвижение гипотез и принятие предположений. Гипотезы при построении модели системы S служат для заполнения «пробелов» в понимании задачи исследователем. 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.

1.5. Определение параметров и переменных модели. Прежде чем перейти к описанию математической модели, необходимо определить параметры системы, входные и выходные переменные, воздействия внешней среды.

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.

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) детерминированная; б) вероятностная; в) определение 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 в абстрактных терминах и понятиях; б) дается описание модели M_x 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_x , которая затем используется на втором этапе моделирования.

1.10. Проверка достоверности концептуальной модели.





Один из методов проверки модели M_x — 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_x 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.

1.11. Составление технической документации по первому этапу. В конце этапа построения концептуальной модели M_x 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 .

4.4 Алгоритмизация модели и ее машинная реализация

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.

2.2. Получение математических соотношений. Одновременно с выполнением подэтапа построения логической схемы модели необходимо построить аналитические модели.

Схема машинной модели M_x должна представлять собой полное отражение заложенной 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_x 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. 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.

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 .

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. 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*, *SIMSCRIPT*, *GPSS*).

2.7. Верификация и проверка достоверности схемы программы.

Верификация программы — доказательство того, что поведение программы соответствует спецификации на программу.

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.

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 .

2.10. Составление технической документации по второму этапу. Для завершения этапа машинной реализации модели M_x 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.





4.5 Получение и интерпретация результатов 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. 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. 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.

3.6. Интерпретация результатов моделирования. Основное содержание этого подэтапа — переход от информации, полученной в результате машинного эксперимента с моделью $M_{м,к}$ 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. При подведении итогов моделирования должны быть отмечены главные особенности, полученные в соответствии с планом эксперимента над моделью $M_{м}$ результатов, проведена проверка гипотез и предположений и сделаны выводы на основании этих результатов.

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.

Таким образом, процесс моделирования системы S сводится к выполнению перечисленных этапов моделирования. На этапе построения





концептуальной модели M_x проводится исследование моделируемого объекта, определяются необходимые аппроксимации, 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 .

5 ФОРМАЛИЗАЦИЯ ОПИСАНИЯ МОДЕЛЕЙ СИСТЕМ И АНАЛИТИЧЕСКИЕ МОДЕЛИ СИСТЕМ МАССОВОГО ОБСЛУЖИВАНИЯ

При построении математических моделей процессов функционирования систем можно выделить следующие основные подходы: непрерывно-детерминированный (например, дифференциальные уравнения); дискретно-детерминированный (конечные автоматы); дискретно-стохастический (вероятностные автоматы); непрерывно-стохастический (системы массового обслуживания); обобщенный, или универсальный (агрегативные системы). Рассмотрим подробнее некоторые из перечисленных подходов.

5.1 Дискретно-детерминированные модели (F-схемы)

Особенности дискретно-детерминированного подхода на этапе формализации процесса функционирования систем рассмотрим на примере использования в качестве математического аппарата теории автоматов.

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 $z_0, z_0 \in Z$; функцией переходов $\varphi(z, x)$; функцией выходов $\psi(z, x)$. Автомат, задаваемый F-схемой: $F = \{Z, X, Y, \varphi, \psi, z_0\}$, — функционирует в дискретном автоматном времени, моментами которого являются такты, т. е. примыкающие друг к другу равные интервалы времени, каждому из которых соответствуют постоянные значения входного и выходного сигналов и внутренние состояния. Обозначим состояние, а также входной и выходной сигналы,





соответствующие некоторому m -му такту при $t=1,2,\dots$ через $z(t), x(t), y(t)$. При этом, по условию, $z(0)=z_0$, а $z(t) \in Z, x(t) \in X, y(t) \in Y$.

Абстрактный конечный автомат имеет один входной и один выходной каналы. В каждый момент $t=1,2,\dots$ дискретного времени F -автомат находится в определенном состоянии $z(t)$ из множества Z состояний автомата, причем в начальный момент времени $t=0$ он всегда находится в начальном состоянии $z(0)=z_0$. В момент t , будучи в состоянии $z(t)$, автомат способен воспринять на входном канале сигнал $x(t) \in X$ и выдать на выходном канале сигнал $y(t)=\psi(z(t), x(t))$, переходя в состояние $z(t+1)=\varphi(z(t), x(t))$, $z(t) \in Z, y(t) \in Y$. Абстрактный конечный автомат реализует некоторое отображение множества слов входного алфавита X на множество слов выходного алфавита Y . Другими словами, если на вход конечного автомата, установленного в начальное состояние z_0 , подавать в некоторой последовательности буквы входного алфавита $x(0), x(1), x(2), \dots$, т. е. входное слово, то на выходе автомата будут последовательно появляться буквы выходного алфавита $y(0), y(1), y(2), \dots$, образуя выходное слово.

Таким образом, работа конечного автомата происходит по следующей схеме: в каждом m -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. Сказанное выше можно описать следующими уравнениями: для F -автомата первого рода, называемого также автоматом Мили,

$$z(t+1)=\varphi[z(t), x(t)], t=0,1,2,\dots;$$

$$y(t)=\psi[z(t), x(t)], t=0,1,2,\dots;$$

для F -автомата второго рода

$$z(t+1)=\varphi[z(t), x(t)], t=0,1,2,\dots;$$

$$y(t)=\psi[z(t), x(t-1)], t=1,2,3,\dots$$

Автомат второго рода, для которого

$$y(t)=\psi[z(t)], t=0,1,2,\dots$$

т. е. функция выходов не зависит от входной переменной $x(t)$, называется автоматом Мура.





5.2 Сетевые модели (N-схемы)

В практике моделирования объектов часто приходится решать задачи, связанные с формализованным описанием и анализом причинно-следственных связей в сложных системах, где одновременно параллельно протекает несколько процессов. Самым распространенным в настоящее время формализмом, описывающим структуру и взаимодействие параллельных систем и процессов, являются сети Петри (англ. Petri Nets).

Формально сеть Петри задается четверкой вида

$$N = \langle B, D, I, O \rangle,$$

где B — конечное множество символов, называемых позициями, $B \neq \emptyset$,
 D — конечное множество символов, называемых переходами, $D \neq \emptyset$, $B \cap D = \emptyset$, I — входная функция (прямая функция инцидентности), $I: B \times D \rightarrow \{0, 1\}$;
 O — выходная функция (обратная функция инцидентности), $O: D \times B \rightarrow \{0, 1\}$.
Таким образом, входная функция I отображает переход d_j во множество входных позиций $b_i \in I(d_j)$, а выходная функция O отображает переход d_j во множество выходных позиций $b_i \in O(d_j)$. Для каждого перехода $d_j \in D$ можно определить множество входных позиций перехода $I(d_j)$ и выходных позиций перехода $O(d_j)$ как

$$I(d_j) = \{b_i \in B \mid I(b_i, d_j) = 1\},$$

$$O(d_j) = \{b_i \in B \mid O(d_j, b_i) = 1\}, \quad i = \overline{1, n}, \quad j = \overline{1, m}, \quad n = |B|, \quad m = |D|$$

Аналогично, для каждого перехода $b_i \in B$ вводятся определения множества входных переходов позиции $I(b_i)$ и множества выходных переходов позиции $O(b_i)$:

$$I(b_i) = \{d_j \in D \mid I(d_j, b_i) = 1\},$$

$$O(b_i) = \{d_j \in D \mid O(b_i, d_j) = 1\}.$$

Графически N -схема изображается в виде двудольного ориентированного мультиграфа, представляющего собой совокупность позиций и переходов (рис. 5.1).



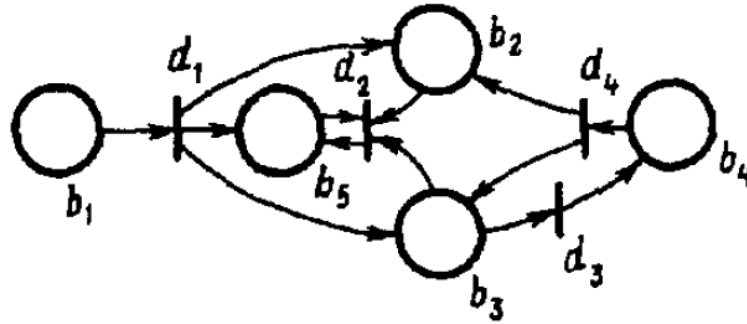


Рисунок 5.1 - Графическое изображение N-схемы

Как видно из этого рисунка, граф *N-схемы* имеет два типа узлов: позиции и переходы, изображаемые 0 и 1 соответственно. Ориентировочные дуги соединяют позиции и переходы, причем каждая дуга направлена от элемента одного множества (позиции или перехода) к элементу другого множества (переходу или позиции). Граф *N-схемы* является мультиграфом, так как он допускает существование кратных дуг от одной вершины к другой.

5.3 Дискретно-стохастические модели (Р-схемы)

Рассмотрим особенности построения математических схем при дискретно-стохастическом подходе к формализации процесса функционирования исследуемой системы. Так как сущность дискретизации времени при этом подходе остается аналогичной рассмотренным конечным автоматам, то влияние фактора стохастичности проследим также на разновидности таких автоматов, а именно на вероятностных (стохастических) автоматах.

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.

Применение схем вероятностных автоматов (Р-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.

Введем математическое понятие Р-автомата, используя понятия, введенные для F-автомата. Рассмотрим множество G , элементами которого являются всевозможные пары (x_i, z_t) . Если существуют две такие функции φ и ψ , что с их помощью осуществляются отображения $G \rightarrow Z$ и $G \rightarrow Y$, то говорят, что $F = \{Z, X, Y, \varphi, \psi\}$ определяет автомат детерминированного типа.

Введем в рассмотрение более общую математическую схему. Пусть Φ — множество всевозможных пар вида (x_k, y_j) , где y_j — элемент выходного подмножества Y . Потребуем, чтобы любой элемент множества G индуцировал на множестве Φ некоторый закон распределения следующего вида:

Элементы из Φ	...	$(z_1, y_1)...$	$(z_1, y_2)...$	...	(z_K, y_{J-1})	(z_K, y_J)
(x_i, z_k)	...	b_{11}	b_{12}	...	$b_{K(J-1)}$	b_{KJ}

При этом $\sum_{k=1}^K \sum_{j=1}^J b_{kj} = 1$, где b_{kj} — вероятности перехода автомата в состояние z_k и появления на выходе сигнала y_j если он был в состоянии z_s и на его вход в этот момент времени поступил сигнал x_i . Число таких распределений, представленных в виде таблиц, равно числу элементов множества G . Обозначим множество этих таблиц через B . Тогда четверка элементов $P = \{Z, X, Y, B\}$ называется вероятностным автоматом (Р-автоматом).

Пусть элементы множества G индуцируют некоторые законы распределения на подмножествах Y и Z , что можно представить соответственно в виде:

Элементы из Y	...	y_1	y_2	...	y_{J-1}	y_J
(x_i, z_s)	...	q_1	q_2	...	q_{J-1}	q_J
Элементы из Z	...	z_1	z_2	...	z_{K-1}	z_K
(x_i, z_s)	...	z_1	z_2	...	z_{K-1}	z_K





При этом $\sum_{k=1}^K z_k = 1$ и $\sum_{k=1}^J q_k = 1$, где z_k и q_k — вероятности перехода Р-автомата в состояние z_k и появления на выходе сигнала y_j если он был в состоянии z_s и на его вход в этот момент времени поступил сигнал x_i

Если для всех k и j имеет место соотношение $q_k z_i = b_{ki}$, то такой Р-автомат называется вероятностным автоматом Мили. Это требование означает выполнение условия независимости распределений для нового состояния Р-автомата и его выходного сигнала.

Пусть теперь определение выходного сигнала Р-автомата зависит лишь от того состояния, в котором находится автомат в данном такте работы. Другими словами, пусть каждый элемент выходного подмножества Y индуцирует распределение вероятностей выходов, имеющее следующий вид:

Элементы из Y	...	y_1	y_2	...	y_{K-1}	y_K
z_K	...	s_1	s_2	...	s_{I-1}	s_I

Здесь $\sum_{i=1}^I s_i = 1$, где s_i — вероятность появления выходного сигнала y_i при условии, что Р-автомат находился в состоянии z_k .

5.4 Комбинированные модели (А-схемы)

Этот подход позволяет описывать поведение непрерывных и дискретных, детерминированных и стохастических систем, т. е. по сравнению с рассмотренными выше подходами является обобщенным (универсальным) и базируется на понятии агрегативной системы (от англ. Aggregate system), представляющей собой формальную схему общего вида, которую будем называть А-схемой.

По традиции, установившейся в математике вообще и в прикладной математике в частности, при агрегативном подходе сначала дается формальное определение объекта моделирования — агрегативной системы, которая является математической схемой, отображающей системный характер изучаемых объектов. При агрегативном описании сложный объект (система) разбивается на конечное число частей (подсистем), сохраняя при этом связи, обеспечивающие их взаимодействие. Если некоторые из полученных подсистем оказываются в свою очередь еще достаточно





сложными, то процесс их разбиения продолжается до тех пор, пока не образуются подсистемы, которые в условиях рассматриваемой задачи моделирования могут считаться удобными для математического описания. В результате такой декомпозиции сложная система представляется в виде многоуровневой конструкции из взаимосвязанных элементов, объединенных в подсистемы различных уровней.

В качестве элемента A -схемы выступает агрегат, а связь между агрегатами (внутри системы S и с внешней средой E) осуществляется с помощью оператора сопряжения R . Очевидно, что агрегат сам может рассматриваться как A -схема, т. е. может разбиваться на элементы (агрегаты) следующего уровня.

Любой агрегат характеризуется следующими множествами: моментов времени T , входных X и выходных Y сигналов, состояний Z в каждый момент времени t . Состояние агрегата в момент времени $t \in T$ обозначается как $z(t) \in Z$, а входные и выходные сигналы — как $x(t) \in X$ и $y(t) \in Y$ соответственно.

Будем полагать, что переход агрегата из состояния $z(t_1)$ в состояние $z(t_2) \neq z(t_1)$ происходит за малый интервал времени, т. е. имеет место скачок δz . Переходы агрегата из состояния $z(t_1)$ в $z(t_2)$ определяются собственными (внутренними) параметрами самого агрегата $h(t) \in H$ и входными сигналами $x(t) \in X$.

В начальный момент времени t_0 состояния z имеют значения, равные z^0 , т. е. $z^0 = z(t_0)$, задаваемые законом распределения процесса $z(t)$ в момент времени t_0 , а именно $L[z(t_0)]$. Предположим, что процесс функционирования агрегата в случае воздействия входного сигнала x_n описывается случайным оператором V . Тогда в момент $t_n \in T$ поступления в агрегат входного сигнала x_n можно определить состояние

$$z(t_n + 0) = V[t_n, z(t_n), x_n].$$

Обозначим полуинтервал времени $t_1 < t \leq t_2$ как $(t_1, t_2]$, а полуинтервал $t_1 \leq t < t_2$ как $[t_1, t_2)$. Если интервал времени (t_n, t_{n+1}) не содержит ни одного момента поступления сигналов, то для любого $t \in (t_n, t_{n+1})$ состояние агрегата определяется случайным оператором U в соответствии с соотношением

$$z(t) = U[t, t_n, z(t_n + 0)].$$

Совокупность случайных операторов V и U рассматривается как оператор переходов агрегата в новые состояния. При этом процесс





функционирования агрегата состоит из скачков состояний δz в моменты поступления входных сигналов x (оператор V) и изменений состояний между этими моментами t_n и t_{n+1} (оператор U). На оператор U не накладывается никаких ограничений, поэтому допустимы скачки состояний δz в моменты времени, не являющиеся моментами поступления входных сигналов x . В дальнейшем моменты скачков δz будем называть особыми моментами времени t_δ , а состояния $z(t_\delta)$ — особыми состояниями A -схемы. Для описания скачков состояний δz в особые моменты времени t_δ будем использовать случайный оператор W , представляющий собой частный случай оператора U , т. е.

$$z(t_\delta + 0) = W[t_\delta, z(t_\delta)].$$

В множестве состояний Z выделяется такое подмножество $Z^{(Y)}$, что если $z(t_\delta)$ достигает $Z^{(Y)}$, то это состояние является моментом выдачи выходного сигнала, определяемого оператором выходов

$$y = G[t_\delta, z(t_\delta)].$$

Таким образом, под агрегатом будем понимать любой объект, определяемый упорядоченной совокупностью рассмотренных множеств $T, X, Y, Z, Z^{(Y)}$ случайных операторов V, U, W, G .

Последовательность входных сигналов, расположенных в порядке их поступления в A -схему, будем называть входным сообщением или x -сообщением. Последовательность выходных сигналов, упорядоченную относительно времени выдачи, назовем выходным сообщением или y -сообщением.

5.5 Потоки заявок

Основным объектом изучения теории массового обслуживания (ТМО) являются системы массового обслуживания (СМО). СМО представляют собой класс математических схем, разработанных для формализации процессов функционирования систем, которые по своей сути являются процессами обслуживания в широком смысле этого слова. ТМО изучает системы со случайным характером поступления заявок и случайным временем обслуживания.

Основными элементами СМО являются входной поток заявок, очереди заявок, ожидающих обслуживания, каналы (приборы) обслуживания,





выходной поток обслуженных заявок. СМО классифицируются по следующим признакам: количеству каналов обслуживания, организации ожидания заявок, количеству фаз обслуживания, взаимосвязи с потоками заявок. По количеству каналов обслуживания СМО делятся на одно- и многоканальные, по организации ожидания заявок - на системы с отказами и системы с ожиданием или с очередями. СМО последнего типа бывают с приоритетами или без приоритетов (с равноценными или неравноценными заявками), а также комбинированного типа. По количеству фаз обслуживания СМО бывают одно- и многофазные. Системы, сочетающие в себе свойства многоканальных и многофазных систем, называются сетями массового обслуживания, для которых характерны более сложные связи, как например повторное обслуживание заявок на определенных фазах. По взаимосвязи СМО с потоками заявок системы делятся на разомкнутые (открытые) и замкнутые. Если интенсивность входного потока заявок не зависит ни от количества заявок, находящихся в СМО, ни от количества обслуженных заявок, то СМО называются разомкнутыми. В противном случае СМО являются замкнутыми.

5.6 Основы марковских процессов

Состояние СМО определяется количеством занятых каналов обслуживания и числом мест в очереди. Естественно, что эти параметры являются целочисленными и меняются скачкообразно в случайные моменты времени, определяемые появлением заявок во входном потоке. Исследование такой системы существенно упрощается, если переход СМО из одного состояния в другое может быть описан марковским процессом.

Определение марковского процесса. Пусть имеется система S , состояние которой изменяется во времени случайным, непредсказуемым образом и представляет собой случайный процесс. Этот процесс называется марковским, если для каждого момента времени t_0 вероятность любого состояния при $t > t_0$ (в будущем) зависит только от вероятности, состояния в момент t_0 (в настоящем) и не зависит от вероятностей состояний при $t < t_0$ (в прошлом).

Иными словами, свойство марковского процесса заключается в том, что на вероятности достижений будущих состояний "предыстория" процесса



не оказывает никакого влияния. Если некоторая система меняет свое состояние скачкообразно, причем переходы из одного состояния в другое обладают марковским свойством, то случайный процесс в такой системе называется марковской цепью.

Марковская цепь называется дискретной, если переход из одного состояния в другое происходит в строго фиксированные промежутки времени, отделенные друг от друга равными интервалами. Если же эти переходы возможны в любой момент времени t , то соответствующая марковская цепь называется непрерывной. Переход из одного состояния в другое может быть отображен графом состояний, в котором вершины представляют собой возможные состояния системы, а дуги графа отражают переходы из одного состояния в другое. Если две вершины i и j соединяются дугой (i, j) , то это означает, что возможен непосредственный переход из состояния i в состояние j . Марковская цепь может, таким образом, быть представлена как случайное блуждание на графе состояний системы.

Поскольку переход из одного состояния в другое для СМО возможен в любой момент времени, определяемый появлением заявки во входном потоке, то для изучения СМО применяются непрерывные марковские цепи.

Одна из важнейших задач теории марковских процессов вообще и ТМО в частности заключается в нахождении вероятностей состояний цепи. Эти вероятности для непрерывных марковских цепей определяются с помощью дифференциальных уравнений Колмогорова.

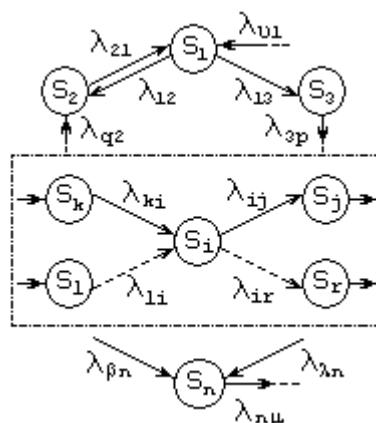


Рисунок 5.2 - Граф состояний

Рассмотрим некоторую произвольную систему, граф состояний которой приведен на рис.5.2. Система имеет n состояний $S_1, S_2, \dots, S_n$. Процесс



перехода из одного состояния в другое описывается непрерывной цепью Маркова. Пусть $P_i(t)$ - вероятность того, что в момент времени t система будет находиться в состоянии $S_i (i=1,2,...,n)$. Поскольку система не может находиться одновременно в двух состояниях, то события, заключающиеся в нахождении системы в состояниях $S_1, S_2, ..., S_n$, несовместны и образуют полную систему событий. Отсюда следует

$$\sum_{i=1}^n p_i(t) = 1. \quad (5.1)$$

Это соотношение называется условием нормировки. Задача заключается в определении вероятности любого состояния $P_i(t)$ в любой момент времени t .

Введем понятие вероятности перехода системы из состояния i , где она находилась в момент времени t , в состояние j за время Δt $P_{ij}(t, \Delta t)$. Очевидно, что $P_{ij}(t, \Delta t)$ представляет собой условную вероятность того, что в момент времени $t + \Delta t$ система окажется в состоянии S_j при условии, что в момент времени t она находилась в состоянии S_i : $p_{ij}(t, \Delta t) = p(S_j(t + \Delta t) | S_i(t))$.

Предел отношения вероятности перехода $p_{ij}(t, \Delta t)$ к длине интервала времени Δt назовем плотностью вероятности перехода

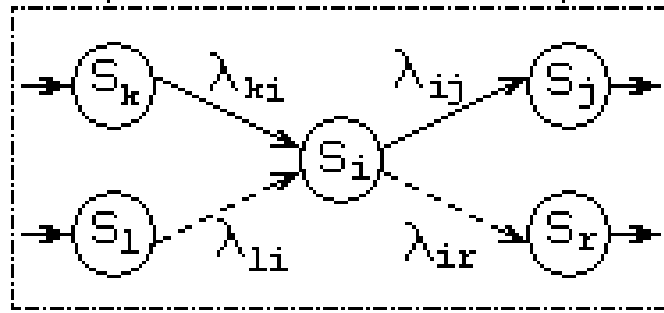
$$\lambda_{ij}(t) = \lim_{\Delta t \rightarrow 0} \frac{p_{ij}(t, \Delta t)}{\Delta t}. \quad (5.2)$$

Плотность вероятности перехода определим только для случаев $i \neq j$.

Если $\lambda_{ij}(t) = \text{const}$, то марковская цепь называется однородной. В противном случае, когда $\lambda_{ij}(t)$ являются функциями времени, цепь называется неоднородной. При расчетах вероятностей состояний марковской цепи предполагается, что все эти плотности вероятностей переходов λ_{ij} известны.

Если у каждой дуги графа состояний системы проставить плотность вероятности перехода по данной дуге, то полученный граф назовем размеченным графом состояний (см. рис.5.2).





Уравнения Колмогорова составляются в соответствии с размеченным графом состояний. Рассмотрим фрагмент размеченного графа состояний (рис.5.2), обведенный штрихпунктирной линией.

Отбросим вначале дуги, изображенные пунктиром, и определим вероятность нахождения системы в состоянии S_i в момент времени $t+\Delta t$. С учетом того, что вершина S_i связана только с вершинами S_k и S_j , указанное событие будет иметь место в двух случаях:

1. - система находилась в состоянии S_i в момент времени t и за время Δt из этого состояния не вышла;
2. - система находилась в состоянии S_k в момент времени t и за время Δt перешла из S_k в S_i .

Если отрезок Δt достаточно мал, то вероятность перехода $p_{ij}(t, \Delta t)$

может быть определена приближенно с помощью (5.2):

$$\lambda_{ij}(t) = \lim_{\Delta t \rightarrow 0} \frac{p_{ij}(t, \Delta t)}{\Delta t}$$

$$p_{ij}(t, \Delta t) \approx \lambda_{ij}(t) \Delta t \quad (5.3)$$

С учетом (5.3) и свойства марковости процесса вероятность первого случая (отсутствие перехода по дуге (S_i, S_j)):

$$p_I = (1 - \lambda_{ij}(t) \Delta t) p_i(t).$$

Вероятность второго случая с учетом (5.3):

$$p_{II} \approx \lambda_{ki}(t) p_k(t) \Delta t.$$

Тогда можно определить искомую вероятность как

$$p_i(t + \Delta t) = p_I + p_{II} = (1 - \lambda_{ij}(t) \Delta t) p_i(t) + \lambda_{ki}(t) p_k(t) \Delta t$$

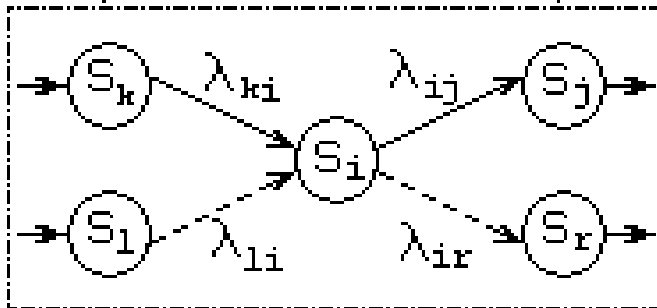
или

$$\frac{p_i(t + \Delta t) - p_i(t)}{\Delta t} = \lambda_{ki}(t) p_k(t) - \lambda_{ij}(t) p_i(t). \quad (5.4)$$

Переходя в (5.4) к пределу при $\Delta t \rightarrow 0$, получим

$$\frac{dp_i(t)}{dt} = \lambda_{kj}(t)p_k(t) - \lambda_{ij}(t)p_i(t) \quad (5.5)$$

Теперь добавим к вершине S_i дуги, обозначенные на рис.5.2 пунктиром.



Тогда при вычислении $p_i(t+\Delta t)$ необходимо учитывать возможный переход из S_i в S_j и S_r (с вероятностью P_I) и переходы из S_k и S_l в S_i (с вероятностью P_{II}). В этом случае:

$$P_I = [1 - (\lambda_{ij}(t) + \lambda_{ir}(t))\Delta t]p_i(t),$$

$$P_{II} = \lambda_{ki}(t)\Delta t p_k(t) + \lambda_{li}(t)\Delta t p_l(t).$$

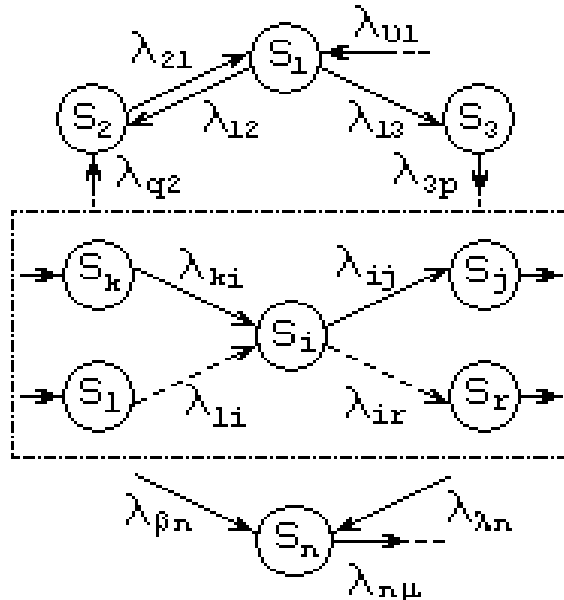
Повторяя вышеописанные рассуждения, получим

$$\frac{dp_i(t)}{dt} = \lambda_{ki}(t)p_k(t) + \lambda_{li}(t)p_l(t) - \lambda_{ij}(t)p_i(t) - \lambda_{ir}(t)p_i(t) \quad (5.6)$$

На основании (5.5) и (5.6) можно сформулировать **четыре правила составления уравнений Колмогорова** по размеченному графу состояний непрерывной марковской цепи:

1. Система дифференциальных уравнений Колмогорова имеет форму Коши. Каждое уравнение составляется с помощью рассмотрения вероятности состояния, представленного соответствующей вершиной в размеченном графе. Число уравнений системы равно числу вершин графа.
2. Число слагаемых правой части каждого уравнения равно числу дуг, инцидентных соответствующей вершине.
3. Дугам с положительной инцидентией соответствуют отрицательные слагаемые, а дугам с отрицательной инцидентией - положительные.
4. Каждое слагаемое правой части равно произведению вероятности состояния, соответствующего началу рассматриваемой дуги, на плотность вероятности перехода по данной дуге.

Начальные условия для системы уравнений Колмогорова определяются начальным состоянием системы. Например, если начальное состояние было S_2 , то начальные условия имеют вид: $p_1(0)=0$; $p_2(0)=1$; $p_3(0)=0$;...; $p_n(0)=0$. Уравнения (5.5) и (5.6) были выведены для общего случая неоднородной марковской цепи. Для однородной марковской цепи все $\lambda_{ij}(i,j=1,...,n)$ постоянны.



Рассмотрим одно важное свойство уравнений Колмогорова (5.5), которое может быть представлено в виде

$$d \bar{p} / dt = \Lambda \cdot \bar{p}, \quad (5.7)$$

где $\bar{P}$ - n-мерный вектор вероятностей состояний системы;
 $p = \{p_1(t), ..., p_n(t)\}$; Λ - $n \times n$ -матрица плотностей перехода.

В соответствии с вышеописанными правилами составления уравнений Колмогорова одна и та же плотность вероятности перехода λ_{ij} будет входить в одно из уравнений со знаком "+", а в другое - со знаком "-", поскольку для двух смежных вершин дуга, соединяющая их, будет обладать положительной инцидентностью по отношению к одной вершине и отрицательной по отношению к другой. Это приведет к тому, что сумма всех элементов в каждом столбце матрицы будет равна нулю. Тогда любая строка матрицы Λ будет равна сумме остальных строк. Следовательно, матрица Λ является всегда вырожденной.

Рассмотрим систему с размеченным графом состояний, изображенным на рис.5.3.

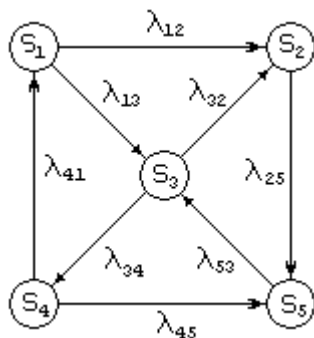


Рисунок 5.3 - Произвольный граф состояний

Система уравнений Колмогорова и матрица Λ для этого случая в соответствии с правилами 1-4 будут иметь вид:

$$\begin{aligned} dp_1/dt &= -(\lambda_{12} + \lambda_{13})p_1 + \lambda_{41}p_4, \\ dp_2/dt &= \lambda_{12}p_1 - \lambda_{25}p_2 + \lambda_{32}p_3, \\ dp_3/dt &= \lambda_{13}p_1 - (\lambda_{32} + \lambda_{34})p_3 + \lambda_{53}p_5, \\ dp_4/dt &= \lambda_{34}p_3 - (\lambda_{41} + \lambda_{45})p_4, \\ dp_5/dt &= \lambda_{25}p_2 + \lambda_{45}p_4 - \lambda_{53}p_5. \end{aligned} \quad (5.7')$$

Исключение любого уравнения из этой системы нарушает указанное соотношение для строк матрицы Λ , следовательно, ранг матрицы Λ будет равен $n-1$. Для того чтобы система уравнений Колмогорова имела единственное решение при заданных начальных условиях, необходимо исключить любое из уравнений системы (5.7') и заменить его условием

нормировки (5.1): $\sum_{i=1}^n p_i(t) = 1.$

Итак, решение системы (5.7) без одного уравнения (безразлично какого) с условием (5.1) определяет в любой момент времени поведение вероятностей состояний марковской цепи при заданных начальных условиях.

Получить это решение можно с помощью любых численных методов (например, Рунге-Кутты, Эйлера-Коши и т.д.), реализуемых на ЭВМ. Только в самых простых случаях система уравнений Колмогорова может быть проинтегрирована в квадратурах. В большинстве практических случаев для расчета вероятностей состояний используются не решения систем уравнений Колмогорова в любой момент времени, а асимптотические оценки этих решений при $t \rightarrow \infty$.



6 ИМИТАЦИОННЫЕ МОДЕЛИ. ОРГАНИЗАЦИЯ СТАТИСТИЧЕСКОГО МОДЕЛИРОВАНИЯ

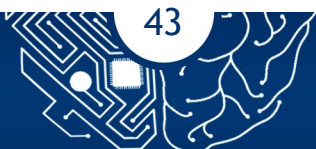
6.1 Характеристика метода статистического моделирования

Сущность метода статистического моделирования сводится к построению для процесса функционирования исследуемой системы S некоторого моделирующего алгоритма, имитирующего поведение и взаимодействие элементов системы с учетом случайных входных воздействий и воздействий внешней среды E , и реализации этого алгоритма с использованием программно-технических средств ЭВМ.

В результате статистического моделирования системы S получается серия частных значений искомых величин или функций, статистическая обработка которых позволяет получить сведения о поведении реального объекта или процесса в произвольные моменты времени. Если количество реализаций N достаточно велико, то полученные результаты моделирования системы приобретают статистическую устойчивость и с достаточной точностью могут быть приняты в качестве оценок искомых характеристик процесса функционирования системы S .

6.2 Псевдослучайные числа и процедуры их машинной генерации

При статистическом моделировании систем одним из основных вопросов является учет стохастических воздействий. Количество случайных чисел, используемых для получения статистически устойчивой оценки характеристики процесса функционирования системы S при реализации моделирующего алгоритма на ЭВМ, колеблется в достаточно широких пределах в зависимости от класса объекта моделирования, вида оцениваемых характеристик, необходимой точности и достоверности результатов моделирования. Для метода статистического моделирования на ЭВМ характерно, что большое число операций, а соответственно и большая доля машинного времени расходуется на действия со случайными числами. Кроме того, результаты статистического моделирования существенно зависят от качества исходных (базовых) последовательностей случайных чисел. Поэтому наличие простых и экономичных способов формирования, последовательностей случайных чисел требуемого качества во многом определяет возможность практического использования машинного



моделирования систем. На практике используются три основных способа генерации случайных чисел: аппаратный (физический), табличный(файловый) и алгоритмический (программный).

При моделировании системна ЭВМ программная имитация случайных воздействий любой сложности сводится к генерированию некоторых стандартных(базовых) процессов и к их последующему функциональному преобразованию.

Вообще говоря, в качестве базового может быть принят любой удобный в случае моделирования конкретной системы S процесс (например, пуассоновский поток при моделировании Q -схемы).

Однако при дискретном моделировании базовым процессом является последовательность чисел $\{x_i\} = x_0, x_1, \dots, x_N$, представляющих собой реализации независимых, равномерно распределенных на интервале $(0, 1)$ случайных величин $\{\xi_i\} = \xi_0, \xi_1, \dots, \xi_N$ или — в статистических терминах — повторную выборку из равномерно распределенной на $(0, 1)$ генеральной совокупности значений величины ξ .

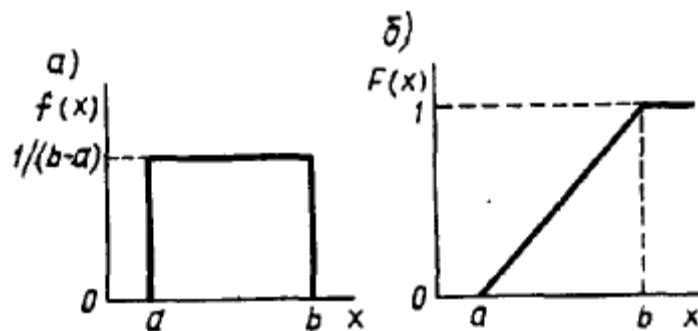


Рисунок 6.1 - Равномерное распределение случайной величины

Непрерывная случайная величина ξ имеет равномерное распределение в интервале (a, b) , если ее функции плотности (рис. 6.1, а) и распределения (рис. 6.1, б) соответственно примут вид:

$$f(x) = \begin{cases} \frac{1}{b-a}, & a \leq x \leq b \\ 0, & x < a, x > b \end{cases}$$

$$F(x) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & x > b \end{cases}$$



Определим числовые характеристики случайной величины ξ , принимающей значения x - математическое ожидание, дисперсию и среднее квадратическое отклонение соответственно:

$$M(\xi) = \int_a^b xf(x)dx = \int_a^b xdx / (b-a) = (a+b)/2$$

$$D(\xi) = \int_a^b (x - M(x))^2 f(x) dx = (b-a)^2 / 12$$

$$\sigma_\xi = \sqrt{D(\xi)} = (b-a) / 2\sqrt{3}$$

При моделировании систем на ЭВМ приходится иметь дело со случайными числами интервала (0, 1), когда границы интервала $a=0$ и $b=1$. Поэтому рассмотрим частный случай равномерного распределения, когда функция плотности и функция распределения соответственно имеют вид:

$$f(x) = \begin{cases} 1, & 0 \leq x \leq 1 \\ 0, & x < 0, x > 1 \end{cases}$$

$$F(x) = \begin{cases} 0, & x < 0 \\ x, & 0 \leq x \leq 1 \\ 1, & x > 1 \end{cases}$$

Такое распределение имеет математическое ожидание $M(\xi) = 1/2$ и дисперсию $D(\xi) = 1/12$.

Это распределение требуется получить на ЭВМ. Но получить его на цифровой ЭВМ невозможно, так как машина оперирует с n -разрядными числами. Поэтому на ЭВМ вместо непрерывной совокупности равномерных случайных чисел интервала (0, 1) используют дискретную последовательность 2^n случайных чисел того же интервала. Закон распределения такой дискретной последовательности называют квазиравномерным распределением.

Случайная величина ξ , имеющая квазиравномерное распределение в интервале (0, 1), принимает значения $x_i = i / (2^n - 1)$ с вероятностями $p_i = 1/2^n$, $i = \overline{0, 2^n - 1}$.

Математическое ожидание и дисперсия квазиравномерной случайной величины соответственно имеют вид:

$$M(\xi) = \sum_{i=0}^{2^n-1} \frac{i}{2^n-1} \frac{1}{2^n} = \frac{1}{(2^n-1)2^n} \sum_{i=0}^{2^n-1} i = \frac{(2^n-1)2^n}{(2^n-1)2^n \cdot 2} = \frac{1}{2}$$

$$D(\xi) = \sum_{i=0}^{2^n-1} \frac{1}{2^n} \left[\frac{i}{2^n-1} - \frac{1}{2} \right]^2 = \frac{1}{12} \cdot \frac{2^n+1}{2^n-1}$$





Таким образом, математическое ожидание квазиравномерной случайной величины совпадает с математическим ожиданием равномерной случайной последовательности интервала $(0, 1)$, а дисперсия отличается только множителем $\frac{2^n+1}{2^n-1}$, который при достаточно больших n близок к единице.

На ЭВМ невозможно получить идеальную последовательность случайных чисел хотя бы потому, что на ней можно оперировать только с конечным множеством чисел. Кроме того, для получения значений x случайной величины ξ используются формулы (алгоритмы).

Поэтому такие последовательности, являющиеся по своей сути детерминированными, называются псевдослучайными.

6.3 Проверка качества последовательности псевдослучайных чисел

Результаты анализа системы S , полученные методом статистического моделирования на ЭВМ, существенно зависят от качества используемых псевдослучайных квазиравномерных последовательностей чисел. Поэтому все применяемые генераторы случайных чисел должны перед моделированием системы пройти тщательное предварительное тестирование, которое представляет собой комплекс проверок по различным статистическим критериям, включая в качестве основных проверки (тесты) на равномерность, стохастичность и независимость.

6.4 Методы генерации случайных воздействий

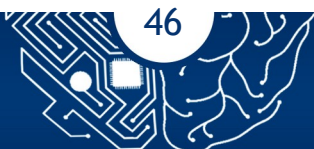
Моделирование дискретных случайных величин. Дискретная случайная величина η принимает значения $y_1 \leq y_2 \leq \dots \leq y_j \leq \dots$ с вероятностями $p_1, p_2, \dots, p_j, \dots$ составляющими дифференциальное распределение вероятностей P или закон распределения $f_\eta(y)$.

При этом интегральная функция распределения непрерывна слева

$$F_\eta(y) = P(\eta < y) = \sum_{j=1}^m p_j; \quad y_m \leq y < y_{m+1}; \quad m=1, 2, \dots;$$

$$F_\eta(y) = 0; \quad y < y_1;$$

Для получения дискретных случайных величин можно использовать метод обратной функции. Если ξ — равномерно распределенная на





интервале $(0, 1)$ случайная величина, то искомая случайная величина η получается с помощью преобразования

$$\eta = F_{\eta}^{-1}(\xi),$$

где F_{η}^{-1} — функция, обратная F_{η} .

Алгоритм вычисления сводится к следующим действиям:

если $x_m < \sum_{j=1}^m p_j$, то $\eta = y_m$ ($j=1, \dots, m$).

Моделирование непрерывных случайных величин. Непрерывная случайная величина η задана интегральной функцией распределения

$$F_{\eta}(y) = P(\eta \leq y) = \int_{-\infty}^y f_{\eta}(y) dy,$$

где $f_{\eta}(y)$ — плотность вероятностей.

Для получения непрерывных случайных величин с заданным законом распределения, как и для дискретных величин, можно воспользоваться методом обратной функции. Взаимно однозначная монотонная функция $\eta = F_{\eta}^{-1}(\xi)$, полученная решением относительно η уравнения $F_{\eta}(y) = \xi$, преобразует равномерно распределенную на интервале $(0, 1)$ величину ξ в η с требуемой плотностью $f_{\eta}(y)$.

Действительно, если случайная величина η имеет плотность распределения $f_{\eta}(y)$, то распределение случайной величины

$$\xi = \int_0^{\eta} f_{\eta}(y) dy$$

является равномерным в интервале $(0, 1)$. На основании этого можно сделать следующий вывод. Чтобы получить число, принадлежащее последовательности случайных чисел $\{y_j\}$, имеющих функцию плотности $f_{\eta}(y)$, необходимо разрешить относительно y_j уравнение

$$\int_{-\infty}^{y_j} f_{\eta}(y) dy = x_i$$

Моделирование случайных векторов. При исследовании характеристик Процессов Функционирования Систем методом статистического моделирования на ЭВМ возникает необходимость в формировании реализаций *случайных векторов*, обладающих заданными вероятностными характеристиками. Случайный вектор можно задать проекциями на оси координат, причем эти проекции являются случайными





величинами, описываемыми совместным законом распределения. В простейшем случае, когда рассматриваемый случайный вектор расположен на плоскости xOy , он может быть задан совместным законом распределения его проекций ξ и η на оси Ox и Oy .

Рассмотрим дискретный случайный процесс, когда двумерная случайная величина (ξ, η) является дискретной и ее составляющая ξ принимает возможные значения $x_1, x_2, \dots, x_n$, а составляющая η — значения $y_1, y_2, \dots, y_n$, причем каждой паре (x_i, y_j) соответствует вероятность p_{ij} . Тогда каждому возможному значению x_i случайной величины ξ , будет соответствовать $p_i = \sum_{j=1}^n p_{ij}$, $j=1, \dots, n$.

Тогда в соответствии с этим распределением вероятностей можно определить конкретное значение x_i случайной величины ξ (по правилам, рассмотренным ранее) и из всех значений p_{ij} выбрать последовательность

$$p_{i_1 1}, p_{i_1 2}, \dots, p_{i_1 n}, \quad (6.1)$$

которая описывает условное распределение величины η при условии, что $\xi = x_{i_1}$. Затем по тем же правилам определяем конкретное значение y_{i_1} случайной величины η в соответствии с распределением вероятностей (6.1). Полученная пара (x_{i_1}, y_{i_1}) будет первой реализацией моделируемого случайного вектора. Далее аналогичным образом определяем возможные значения x_{i_2} , выбираем последовательность

$$p_{i_2 1}, p_{i_2 2}, \dots, p_{i_2 n}, \quad (6.2)$$

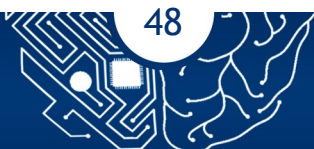
и находим y_{i_2} в соответствии с распределением (6.2). Это дает реализацию вектора (x_{i_2}, y_{i_2}) и т. д.

Рассмотрим моделирование непрерывного случайного вектора с составляющими ξ и η . В этом случае двумерная случайная величина (ξ, η) описывается совместной функцией плотности $f(x, y)$.

Эта функция может быть использована для определения функции плотности случайной величины ξ как

$$f_{\xi}(x) = \int_{-\infty}^{+\infty} f(x, y) dy$$

Имея функцию плотности $f_{\xi}(x)$, можно найти случайное число x_i , а затем при условии, что $\xi = x_i$, определить условное распределение случайной величины η :





$$f_{\eta}(y|\xi=x_i)=f(x,y)/f_{\xi}(x_i)$$

В соответствии с этой функцией плотности можно определить случайное число y_i . Тогда пара чисел (x_i, y_j) будет являться искомой реализацией вектора (ξ, η) .

7 СРЕДСТВА МОДЕЛИРОВАНИЯ СИСТЕМ

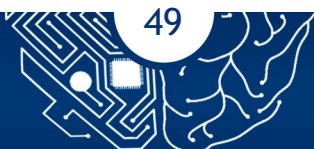
7.1 Моделирование систем и языки программирования.

Алгоритмические языки при моделировании систем служат вспомогательным аппаратом разработки, машинной реализации и анализа характеристик моделей систем S . Каждый язык моделирования должен отражать определенную структуру понятий для описания широкого класса явлений.

Выбрав для решения задачи моделирования процесса функционирования системы S конкретный язык, исследователь получает в распоряжение тщательно разработанную систему абстракций, предоставляющих ему основу для формализации процесса функционирования исследуемой системы S . Высокий уровень проблемной ориентации языка моделирования значительно упрощает программирование моделей, а специально предусмотренные в нем возможности сбора, обработки и вывода результатов моделирования позволяют быстро и подробно анализировать возможные исходы имитационного эксперимента с моделью M_M .

Язык моделирования представляет собой процедурно-ориентированный язык, обладающий специфическими чертами. Основные языки моделирования разрабатывались в качестве программного обеспечения имитационного подхода к изучению процесса функционирования определенного класса систем.

Целесообразность использования языков имитационного моделирования (ЯИМ) вытекает из двух основных причин: 1) удобство программирования модели системы, играющее существенную роль при машинной реализации моделирующих алгоритмов; 2) концептуальная направленность языка на класс систем, необходимая на этапе построения модели системы и выборе общего направления исследований в планируемом машинном эксперименте. Практика моделирования систем показывает, что





именно использование ЯИМ во многом определило успех имитации как метода экспериментального исследования сложных реальных объектов.

ЯИМ делятся на две самостоятельные группы, которые соответствуют двум видам имитации, развивавшимся независимо друг от друга: для имитации непрерывных и дискретных процессов.

Для моделирования непрерывных процессов схема моделирующего алгоритма преобразуется в систему совместно рассматриваемых дифференциальных уравнений. Моделирование в этом случае сводится, по сути дела, к отысканию численных решений этих уравнений при использовании некоторого стандартного пошагового метода.

Примером языка моделирования непрерывных систем на ЭВМ путем представления моделируемой системы в виде уравнений в конечных разностях является язык *DYNAMO*, для которого уравнения устанавливают соотношения между значениями функций в моменты времени t и $t+dt$ между значениями их производных в момент времени $t+dt/2$. И в этом случае моделирование, по существу, представляет собой пошаговое решение заданной системы дифференциальных уравнений.

Универсальная ЭВМ — устройство дискретного типа, а поэтому должна обеспечивать дискретную аппроксимацию процесса функционирования исследуемой системы S . Непрерывные изменения в процессе функционирования реальной системы отображаются в дискретной модели M_M , реализуемой на ЭВМ, некоторой последовательностью дискретных событий, и такие модели называются *моделями дискретных событий*. Отдельные события, отражаемые в дискретной модели, могут определяться с большой степенью приближения к действительности, что обеспечивает адекватность таких дискретных моделей реальным процессам, протекающим в непрерывных системах S .

Архитектуру ЯИМ, т. е. концепцию взаимосвязей элементов языка как сложной системы, и технологию перехода от системы S к ее машинной модели M_M можно представить следующим образом: 1) объекты моделирования (системы S) описываются (отображаются в языке) с помощью некоторых атрибутов языка; 2) атрибуты взаимодействуют с процессами, адекватными реально протекающим явлениям в моделируемой системе S ; 3) процессы требуют конкретных условий, определяющих логическую основу и



последовательность взаимодействия этих процессов во времени; 4) условия влияют на события, имеющие место внутри объекта моделирования (системы S) и при взаимодействии с внешней средой E ; 5) события изменяют состояния модели системы M_M в пространстве и во времени.

Типовая схема архитектуры ЯИМ и технология его использования при моделировании систем показана на рис. 7.1.

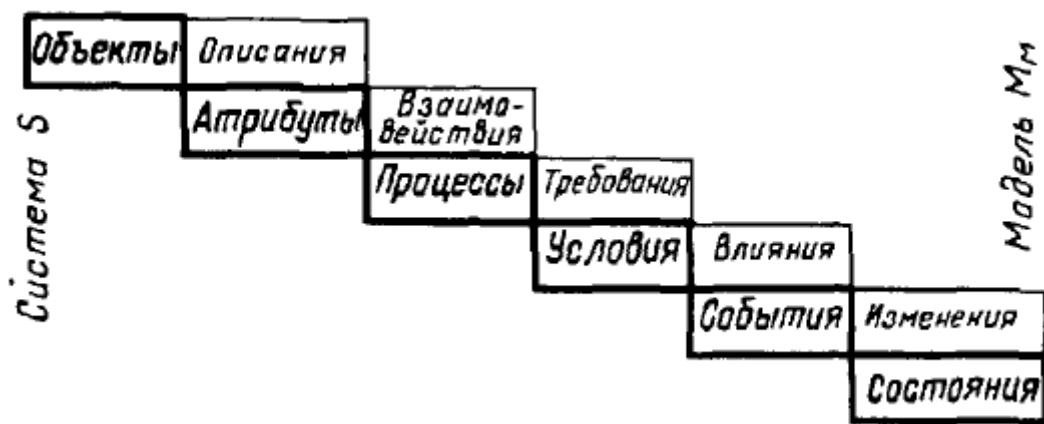


Рисунок 7.1 - Типовая схема архитектуры ЯИМ

В большинстве случаев с помощью машинных моделей исследуются характеристики и поведение системы S на определенном отрезке времени, поэтому одной из наиболее важных задач при создании модели системы M_M и выборе языка программирования модели является реализация двух функций: 1) корректировка временной координаты состояния системы («продвижение» времени, организация «часов»); 2) обеспечение согласованности различных блоков и событий в системе (синхронизация во времени, координация с другими блоками).

Таким образом, функционирование модели M_M должно протекать в искусственном (не в реальном и не в машинном) времени. Искусственное время обеспечивает появление событий в требуемом логикой работы исследуемой системы порядке и с надлежащими временными интервалами между ними. При этом надо учитывать, что элементы реальной системы S функционируют одновременно (параллельно), а компоненты машинной модели M_M действуют последовательно, так как реализуются с помощью ЭВМ последовательного действия. Поскольку в различных частях объекта моделирования события могут возникать одновременно, то для сохранения адекватности причинно-следственных временных связей необходимо в ЯИМ

создать «механизм» задания времени для синхронизации действий элементов модели системы.

7.2 Сравнительный анализ языков имитационного моделирования

Для сравнения существующих ЯИМ рассмотрим классификацию, представленную на рисунке 7.2.

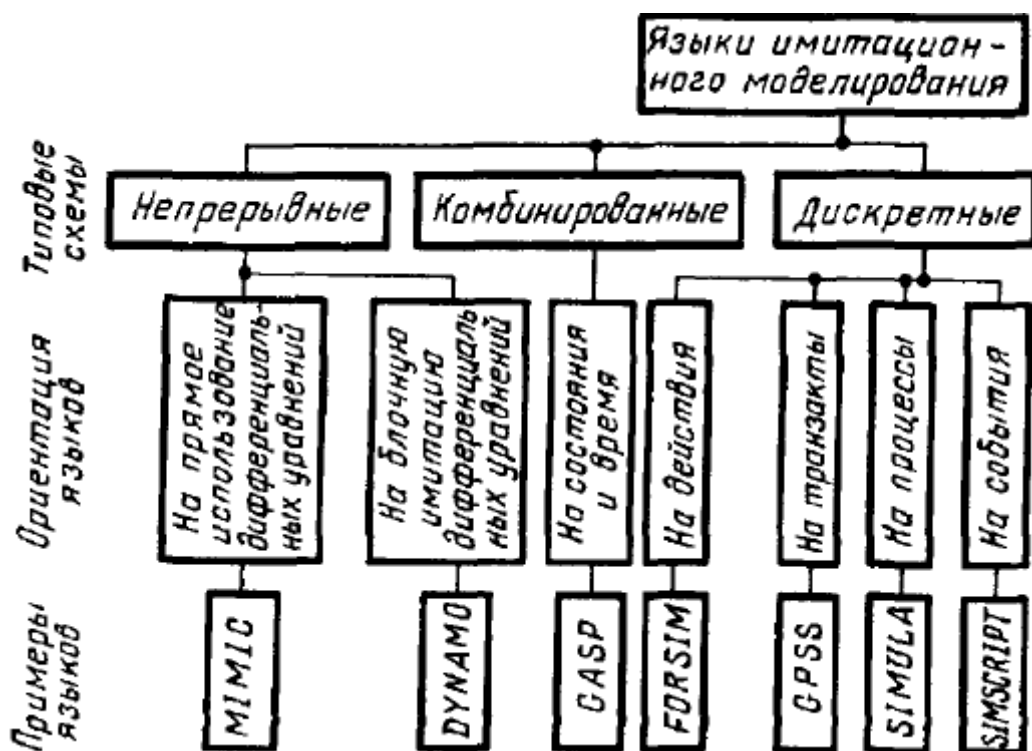


Рисунок 7.2 - Классификация ЯИМ

Непрерывное представление системы S сводится к составлению уравнений, с помощью которых устанавливается связь между эндогенными и экзогенными переменными модели. Примером такого непрерывного подхода является использование дифференциальных уравнений. Причем в дальнейшем дифференциальные уравнения могут быть применены для непосредственного получения характеристик системы, это, например, реализовано в языке *MIMIC*. А в том случае, когда экзогенные (входные) переменные модели принимают дискретные значения, уравнения являются разностными. Такой подход реализован, например, в языке *DYNAMO*.

Представление системы S в виде типовой схемы, в которой участвуют как непрерывные, так и дискретные величины, называется **комбинированным**. Примером языка, реализующего комбинированный подход, является *GASP*, построенный на базе языка *FORTRAN*. Язык *GASP* включает в себя набор программ, с помощью которых моделируемая



система S представляется в следующем виде. Состояние модели системы $M(S)$ описывается набором переменных, некоторые из которых меняются во времени непрерывно. Законы изменения непрерывных переменных заложены в структуру, объединяющую дифференциальные уравнения и условия относительно этих переменных. Предполагается, что в системе могут наступать события двух типов: 1) события, зависящие от состояния z_j , 2) события, зависящие от времени t_i . События первого типа наступают в результате выполнения условий, относящихся к законам изменения непрерывных переменных. Для событий второго типа процесс моделирования состоит в продвижении системного времени от момента наступления события до следующего аналогичного момента. События приводят к изменениям состояния модели системы и законов изменения непрерывных компонент. При использовании языка *GASP* на пользователя возлагается работа по составлению на языке *FORTTRAN* подпрограмм, в которых он описывает условия наступления событий, зависящих от процесса функционирования системы S , законы изменения непрерывных переменных, а также правила перехода из одного состояния в другое.

В рамках *дискретного подхода* можно выделить несколько принципиально различных групп ЯИМ. Первая группа ЯИМ подразумевает наличие списка событий, отличающих моменты начала выполнения операций. Продвижение времени осуществляется по событиям, в моменты наступления которых производятся необходимые операции, включая операции пополнения списка событий. Примером языка событий является язык *SIMSCRIPT*. Разработчики языка *SIMSCRIPT* исходили из того, что каждая модель M_m состоит из элементов, с которыми происходят события, представляющие собой последовательность предложений, изменяющих состояния моделируемой системы в различные моменты времени. Моделирование с помощью языка *SIMSCRIPT* включает в себя следующие этапы: а) элементы моделируемой системы S описываются и вводятся с помощью карт определений; б) вводятся начальные условия; в) фиксируются и вводятся исходные значения временных параметров; г) составляются подпрограммы для каждого события; д) составляется перечень событий и указывается время свершения каждого эндогенного события. Команды языка *SIMSCRIPT* группируются следующим образом: операции над временными





объектами, арифметические и логические операции и команды управления, команды ввода-вывода, специальные команды обработки результатов. К центральным понятиям языка *SIMSCRIPT* относятся обработка списков с компонентами, определяемыми пользователем, и последовательность событий в системном времени. При этом имеются специальные языковые средства для работы с множествами.

Сравнение эффективности языков. При анализе эффективности использования для моделирования конкретной системы S того или иного ЯИМ выделяют несколько важных свойств языков: возможность описания структуры и алгоритмов поведения исследуемой системы S в терминах языка; простота применения для построения модели M , ее машинной реализации M_M и обработки результатов моделирования; предпочтение пользователя, обычно отдаваемое языку, который ему более знаком или который обладает большей степенью универсальности, и т. д. При этом, естественно, большее количество команд ЯИМ обеспечивает лучшие возможности при написании программы моделирования. Однако вместе с увеличением числа команд возрастают трудности использования ЯИМ, поэтому пользователь обычно отдает предпочтение языкам, обладающим большей гибкостью при минимальном количестве команд.

Исходя из этих соображений, приводились экспертные оценки для сравнения различных языков при моделировании широкого класса систем. Результаты оценок сведены в табл. 7.1. Языки даны в порядке уменьшения их эффективности.

Таблица 7.1 - Сравнение некоторых ЯИМ

Возможности языка	Простота применения	Предпочтение пользователя
<i>SIMULA</i>	<i>GPSS</i>	<i>SIMSCRIPT</i>
<i>SIMSCRIPT</i>	<i>SIMSCRIPT</i>	<i>GPSS</i>
<i>GPSS</i>	<i>SIMULA</i>	<i>SIMULA</i>

Перечисленные особенности ЯИМ во многом определяют возможности выбора того или иного языка для целей проведения имитационного эксперимента с моделью системы S , причем в каждом конкретном случае на выбор языка моделирования оказывают влияние многие факторы его практической реализации. Задачи выбора ЯИМ должны рассматриваться как





одна из комплекса задач, решаемых при автоматизации процесса моделирования систем с использованием современных ЭВМ.

7.3 Пакеты программ моделирования

Пакеты прикладных программ моделирования (ППМ) являются одной из основных форм специализированного программного обеспечения. ППМ— это комплекс взаимосвязанных программ моделирования и средств системного обеспечения (программных и языковых), предназначенных для автоматизации решения задач моделирования.

Весь круг работ, связанных с разработкой алгоритмов и программ моделирования, а также с подготовкой и проведением машинных экспериментов, называется автоматизацией моделирования и реализуется в виде конкретных автоматизированных систем моделирования (АСМ).

В структуре ППМ можно выделить три основных компонента: функциональное наполнение, язык заданий и системное наполнение.

Функциональное наполнение ППМ отражает специфику предметной области применительно к конкретному объекту моделирования, т. е. системе S , и представляет собой совокупность модулей. Под модулем здесь понимается конструктивный элемент, используемый на различных стадиях функционирования пакета. Язык (языки), на котором записываются модули функционального наполнения, будем называть *базовым языком* ППМ. Состав функционального наполнения пакета, его мощность или полнота охвата им предметной области отражают объем прикладных знаний, заложенных в ППМ, т. е. потенциальный уровень тематической квалификации пакета.

Одной из ключевых проблем разработки ППМ является *модуляризация*, т. е. разбиение функционального наполнения пакета на модули. Тщательно выполненный анализ объекта моделирования и проведенная на его основе модуляризация позволяют сократить объем работ по реализации ППМ, повышают его надежность и облегчают дальнейшую эволюцию пакета.

Число разнообразных форм модулей, используемых в пакетах, весьма велико. Прежде всего, следует выделить программные модули, модули данных и модули документации. Для программных модулей известны, например, такие формы, как подпрограмма; конструкция алгоритмического языка, допускающая автономную трансляцию; макроопределение; файл,





содержащий такой текст фрагмента программы, который рассматривается как самостоятельный объект для изучения или редактирования; набор указаний, задающих способ построения конкретной версии программы; реализация абстрактного типа данных и др.

Язык заданий НИМ является средством общения пользователя (разработчика или исследователя машинной модели M_M процесса функционирования системы S) с пакетом. Он позволяет описывать последовательность выполнения различных операций, обеспечивающих решение задачи моделирования, или постановку задачи моделирования, по которой эта последовательность строится автоматически.

Главная цель разработки языка заданий ППМ, обеспечивающего решение задач моделирования в режиме пассивной дисциплины, заключается в том, чтобы «спрятать» от конечного пользователя основную массу алгоритмических подробностей моделирования его конкретной системы S , или, другими словами, повысить уровень непроцедурности языка. Такие языки, называемые *языками запросов*, ориентированы обычно на формулирование содержательных постановок задач, т. е. запросов, указывающих, «что необходимо получить», без явного задания того, «как это получить». Пользователь тем самым избавляется от необходимости конкретизировать способы и средства решения его задачи моделирования конкретной системы S , что позволяет понизить порог требований к уровню его программистской подготовки.

Системное наполнение ППМ представляет собой совокупность программ, которые обеспечивают выполнение заданий и взаимодействие пользователя с пакетом, адекватное дисциплине работы в данной прикладной деятельности. Можно сказать, что системное наполнение организует использование потенциала знаний, заложенных в функциональном наполнении, в соответствии с возможностями, предусмотренными в языке заданий ППМ.

8 ПЛАНИРОВАНИЕ МАШИННЫХ ЭКСПЕРИМЕНТОВ

Машинный эксперимент с моделью системы при ее исследовании и проектировании проводится с целью получения информации о характеристиках процесса функционирования рассматриваемого объекта.





Эта информация может быть получена как для анализа характеристик, так и для их оптимизации при заданных ограничениях, т. е. для синтеза структуры, алгоритмов и параметров системы.

Рассмотрим основные понятия теории планирования экспериментов. Наиболее подходящей моделью последнего является абстрактная схема, называемая «черным ящиком». При таком подходе различают входные и выходные переменные: $x_1, x_2, \dots, x_k$; $y_1, y_2, \dots, y_i$. В зависимости от того, какую роль играет каждая переменная в проводимом эксперименте, она может являться либо фактором, либо реакцией. Пусть, например, имеют место только две переменные: x и y . Тогда если цель эксперимента – изучение влияния переменной x на переменную y , то x – фактор, а y – реакция. В экспериментах с машинными моделями системы фактор является экзогенной или управляемой (входной) переменной, а реакция – эндогенной (выходной) переменной.

Каждый фактор x_i , может принимать в эксперименте одно из нескольких значений, называемых уровнями. Фиксированный набор уровней факторов определяет одно из возможных состояний рассматриваемой системы. Одновременно этот набор представляет собой условия проведения одного из возможных экспериментов.

Существует вполне определенная связь между уровнями факторов и реакцией (откликом) системы, которую можно представить в виде соотношения

$$y_l = \psi_l(x_1, x_2, \dots, x_k), l = \overline{1, m}$$

Функцию ψ , связывающую реакцию с факторами, называют функцией реакции, а геометрический образ, соответствующий функции реакции, – поверхностью реакции. Исследователю заранее не известен вид зависимостей $\psi_l, l = \overline{1, m}$ поэтому используют приближенные соотношения: $\tilde{y}_l = \varphi_l(x_1, x_2, \dots, x_k), l = \overline{1, m}$.

Приближенные зависимости $\varphi_l, l = \overline{1, m}$ находятся по данным эксперимента. Последний необходимо поставить так, чтобы при минимальных затратах ресурсов (например, минимальном числе испытаний), варьируя по специально сформулированным правилам значения входных переменных, построить математическую модель системы и оценить ее характеристики.





Виды планов экспериментов. Эксперимент, в котором реализуются все возможные сочетания уровней факторов, называется полным факторным экспериментом (ПФЭ).

Математическая модель эксперимента с n факторами описывается квадратичным уравнением регрессии:

$$y = b_0 + b_1 X_1 + b_2 X_2 + \dots + b_n X_n + b_{12} X_1 X_2 + \dots + b_{(n-1)n} X_{(n-1)} X_n$$

Таким образом, общее число опытов в случае ПФЭ будет равно $N = 2^n$.

Для удобства нахождения коэффициентов регрессии для двух факторов x_1, x_2 рассмотрим только два факторных уровня со значениями $\{+1, -1\}$.

Рассмотрим матрицу-таблицу полного двухфакторного и трёхфакторного эксперимента.

$$y = b_0 + b_1 X_1 + b_2 X_2 + b_{12} X_1 X_2$$

№ эксп	x_1 факт	x_2 факт	$x_1 x_2$	Реакция
1	-1	-1	+1	y_1
2	+1	-1	-1	y_2
3	-1	+1	-1	y_3
4	+1	+1	+1	y_4

$$y = b_0 + b_1 X_1 + b_2 X_2 + b_3 X_3 + b_{12} X_1 X_2 + b_{13} X_1 X_3 + b_{23} X_2 X_3 + b_{123} X_1 X_2 X_3$$

№ эксп	x_1 факт	x_2 факт	x_3	$x_1 x_2$	$x_1 x_3$	$x_2 x_3$	$x_1 x_2 x_3$	Реакция
1	-1	-1	-1	+1	+1	+1	-1	y_1
2	-1	-1	+1	+1	-1	-1	+1	y_2
3	-1	+1	-1	-1	+1	-1	+1	y_3
4	-1	+1	+1	-1	-1	+1	-1	y_4
5	+1	-1	-1	-1	-1	+1	+1	y_5



6	+1	-1	+1	-1	+1	-1	-1	y_6
7	+1	+1	-1	+1	-1	-1	-1	y_7
8	+1	+1	+1	+1	+1	+1	+1	y_8

Как видно из рассмотренных планов экспериментов типов 2^2 и 2^3 , количество испытаний в ПФЭ значительно превосходит число определяемых коэффициентов линейной модели плана эксперимента, т. е. ПФЭ обладает большой избыточностью и поэтому возникает проблема сокращения их количества.

Дробный факторный эксперимент (ДФЭ) — эксперимент, в котором реализуется лишь некоторые части ПФЭ (дробные реплики).

Рассмотрим построение плана дробного трёхфакторного эксперимента. Как показано выше, при ПФЭ число испытаний $N = 8$, число факторов $n=3$.

Пусть имеется простейший полный факторный эксперимент типа 2^2 . Используя матрицу планирования, приведенную в табл. Плана двухфакторного эксперимента, можно вычислить коэффициенты и представить результаты в виде уравнения $y = b_0 + b_1 X_1 + b_2 X_2 + b_{12} X_1 X_2$.

Если в выбранных интервалах варьирования уровня x_1, x_2 процесс можно описать линейной моделью, то достаточно определить три коэффициента: b_0, b_1, b_2 . Таким образом, остается одна степень свободы, которую можно использовать для минимизации числа испытаний. При линейном приближении $b_{12} \rightarrow 0$ и вектор-столбец $X_1 X_2$ можно использовать для нового фактора X_3 .

В этом случае отдельных оценок, которые имели место в ПФЭ типа 2^k , уже не будет и оценки смещаются следующим образом:

$$b_1 \rightarrow \beta_1 + \beta_{23}, b_2 \rightarrow \beta_2 + \beta_{13}, b_3 \rightarrow \beta_3 + \beta_{12}.$$

Таким образом, вместо восьми испытаний в ПФЭ типа 2^3 необходимо провести только четыре. Правило проведения дробного факторного эксперимента формулируется так: для сокращения числа испытаний новому фактору присваивается значение вектора-столбца матрицы, принадлежащего взаимодействию, которым можно пренебречь.



Для обозначения дробных реплик, в которых d линейных эффектов приравнены к эффектам взаимодействия, пользуются условным обозначением 2^{k-d} .

Применяя системный подход к проблеме планирования машинных экспериментов с моделями систем, можно выделить две составляющие планирования: *стратегическое и тактическое планирование*.

Стратегическое планирование ставит своей целью решение задачи получения необходимой информации о системе с помощью модели, реализованной на ЭВМ, с учетом ограничений на ресурсы, имеющиеся в распоряжении экспериментатора. По своей сути стратегическое планирование аналогично внешнему проектированию при создании системы, только здесь в качестве объекта выступает процесс моделирования системы.

Применяя системный подход к проблеме стратегического планирования машинных экспериментов, можно выделить следующие этапы: 1) построение структурной модели; 2) построение функциональной модели. При этом структурная модель выбирается исходя из того, что должно быть сделано, а функциональная — из того, что может быть сделано.

Тактическое планирование представляет собой определение способа проведения каждой серии испытаний машинной модели, предусмотренных планом эксперимента. Для тактического планирования также имеется аналогия с внутренним проектированием системы, но опять в качестве объекта рассматривается процесс работы с моделью.

