

Б.Я. СОВЕТОВ С.А. ЯКОВЛЕВ

Моделирование систем

**Издание третье,
переработанное и дополненное**

**Рекомендовано Министерством образования
Российской Федерации в качестве учебника
для студентов высших учебных заведений,
обучающихся по направлениям
«Информатика и вычислительная техника»
и «Информационные системы»**

УДК 519.87
ББК 22.18
С 56

Рецензент: д.т.н., профессор О. С. Чугреев
(Санкт-Петербургский государственный университет
телекоммуникаций)

Советов Б. Я., Яковлев С. А.

С 56 Моделирование систем: Учеб. для вузов — 3-е изд., перераб. и доп. — М.: Высш. шк., 2001. — 343 с.: ил.

ISBN 5-06-003860-2

Даются фундаментальные основы теории моделирования, приводятся определения основных понятий компьютерной имитации, рассматриваются подходы к моделированию процессов и явлений в природе и обществе, особое внимание уделяется математическому аппарату формализации процессов в сложных системах, методически последовательно показывается переход от концептуальных моделей систем к формальным, приводится методология статистического моделирования систем, обсуждаются проблемы интерпретации полученных с помощью компьютерной модели результатов применительно к объекту моделирования, т. е. исследуемой системе.

Третье издание (2-е — 1998 г.) отличается такими особенностями, как представление новой скорректированной методики имитационного моделирования сложных систем, рассмотрение интеллектуальной системы моделирования, а также скорректированным и расширенным математическим аппаратом.

Для студентов вузов, обучающихся по направлениям «Информатика и вычислительная техника» и «Информационные системы». Может быть полезной специалистам в области моделирования сложных информационных систем.

УДК 519.87
ББК 22.18

ISBN 5-06-003860-2

© ГУП «Издательство «Высшая школа», 2001

ПРЕДИСЛОВИЕ

Современный этап развития человечества отличается тем, что на смену века энергетики приходит век информатики. Происходит интенсивное внедрение новых информационных технологий во все сферы человеческой деятельности. Встает реальная проблема перехода в информационное общество, для которого приоритетным должно стать развитие образования. Изменяется и структура знаний в обществе. Все большее значение для практической жизни приобретают фундаментальные знания, способствующие творческому развитию личности. Важна и конструктивность приобретаемых знаний, умение их структурировать в соответствии с поставленной целью. На базе знаний формируются новые информационные ресурсы общества. Формирование и получение новых знаний должно базироваться на строгой методологии системного подхода, в рамках которого особое место занимает модельный подход. Возможности модельного подхода крайне многообразны как по используемым формальным моделям, так и по способам реализации методов моделирования. Физическое моделирование позволяет получить достоверные результаты для достаточно простых систем.

Сложные по внутренним связям и большие по количеству элементов системы экономически трудно поддаются прямым способам моделирования и зачастую для построения и изучения переходят к имитационным методам. Появление новейших информационных технологий увеличивает не только возможности моделирующих систем, но и позволяет применять большее многообразие моделей и способов их реализации. Совершенствование вычислительной и телекоммуникационной техники привело к дальнейшему развитию методов машинного моделирования, без которых невозможно изучение процессов и явлений, а также построение больших и сложных систем. Поэтому дисциплина «Моделирование систем» стала базовой в структуре подготовки бакалавров и специалистов по ряду направлений высшего профессионального образования. Среди этих направлений прежде всего следует отметить 654600 — «Информатика и вычислительная техника», 654700 — «Информационные системы», а также ряд специальностей, среди которых 654602 — «Автоматизированные системы обработки информации и управления» (АСОИУ), 654701 — «Информационные системы в технике и технологиях» (ИС) и др.

Материал предлагаемого учебника является третьим дополненным и переработанным изданием и соответствует программам дисциплины «Моделирование систем» учебного плана подготовки инженеров по специальностям 654602 — АСОИУ и 654701 — ИС. При подготовке третьего издания использована новая отечественная и зарубежная литература по моделированию систем различных классов, опыт разработки реальных систем, а также результаты использования ранее выпущенных учебников «Моделирование систем» в учебном процессе на кафедре Автоматизированных систем обработки информации и управления Санкт-Петербургского государственного электротехнического университета ЛЭТИ.

Автору благодарны д.т.н., профессору О. С. Чугрееву за ценные замечания, сделанные при рецензировании рукописи.

Пожелания по содержанию книги просим направлять по адресу: 127994, Москва, ГСП-4, Неглинная ул., 29/14, издательство «Высшая школа».

Авторы

ВВЕДЕНИЕ

СОВРЕМЕННОЕ СОСТОЯНИЕ ПРОБЛЕМЫ МОДЕЛИРОВАНИЯ СИСТЕМ

Моделирование (в широком смысле) является основным методом исследований во всех областях знаний и научно обоснованным методом оценок характеристик сложных систем, используемым для принятия решений в различных сферах инженерной деятельности. Существующие и проектируемые системы можно эффективно исследовать с помощью математических моделей (аналитических и имитационных), реализуемых на современных ЭВМ, которые в этом случае выступают в качестве инструмента экспериментатора с моделью системы.

В.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. Остановимся на философских аспектах моделирования, а точнее общей теории моделирования [35, 37, 43].

Методологическая основа моделирования. Все то, на что направлена человеческая деятельность, называется *объектом* (лат. *objectum* — предмет). Выработка методологии направлена на упорядочение получения и обработки информации об объектах, которые существуют вне нашего сознания и взаимодействуют между собой и внешней средой.

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 в качестве метода суждения имеет аналогия.

Аналогией называют суждение о каком-либо частном сходстве двух объектов, причем такое сходство может быть существенным и несущественным. Необходимо отметить, что понятия существенности и несущественности сходства или различия объектов условны и относительны. Существенность сходства (различия) зависит от уровня абстрагирования и в общем случае определяется конечной

целью проводимого исследования. 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.

Определение моделирования. 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* [5, 36, 46].

Определяя гносеологическую роль теории моделирования, т. е. ее значение в процессе познания, необходимо прежде всего отвлечься от имеющегося в науке и технике многообразия моделей и выделить то общее, что присуще моделям различных по своей природе объектов реального мира. Это общее заключается в наличии некоторой структуры (статической или динамической, материальной или мысленной), которая подобна структуре данного объекта. В процессе изучения модель выступает в роли относительного самостоятельного квазиобъекта, позволяющего получить при исследовании некоторые знания о самом объекте.

Если результаты моделирования подтверждаются и могут служить основой для прогнозирования процессов, протекающих в исследуемых объектах, то говорят, что модель адекватна объекту. При этом *адекватность* модели зависит от цели моделирования и принятых критериев.

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.

Следует отметить, что с точки зрения философии моделирование — эффективное средство познания природы. 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. Такой эксперимент для инженера есть инструмент непосредственного решения организационно-технических задач.

Надо иметь в виду, что любой эксперимент может иметь существенное значение в конкретной области науки только при специальной его обработке и обобщении. Единичный эксперимент никогда не может быть решающим для подтверждения гипотезы, проверки теории. Поэтому инженеры (исследователи и практики) должны быть знакомы с элементами современной методологии теории познания и, в частности, не должны забывать основного положения материалистической философии, что именно экспериментальное исследование, опыт, практика являются критерием истины.

В.2. ИСПОЛЬЗОВАНИЕ МОДЕЛИРОВАНИЯ ПРИ ИССЛЕДОВАНИИ И ПРОЕКТИРОВАНИИ СЛОЖНЫХ СИСТЕМ

Одна из проблем современной науки и техники — разработка и внедрение в практику проектирования новейших методов исследования характеристик сложных информационно-управляющих и информационно-вычислительных систем различных уровней (например, автоматизированных систем научных исследований и комплексных испытаний, систем автоматизации проектирования, АСУ технологическими процессами, а также интегрированных АСУ, вычислительных систем, комплексов и сетей, информационных систем, цифровых сетей интегрального обслуживания и т. д.). При проектировании сложных систем и их подсистем возникают многочисленные задачи, требующие оценки количественных и качественных закономерностей процессов функционирования таких

систем, проведения структурного алгоритмического и параметрического их синтеза [4, 11, 37, 45].

Особенности разработки систем. Рассматриваемые в данном учебнике системы информатики и вычислительной техники, автоматизированные системы обработки информации и управления, информационные системы относятся к классу больших систем, этапы проектирования, внедрения, эксплуатации и эволюции которых в настоящее время невозможны без использования различных видов моделирования. На всех перечисленных этапах для сложных видов различных уровней необходимо учитывать следующие особенности: сложность структуры и стохастичность связей между элементами, неоднозначность алгоритмов поведения при различных условиях, большое количество параметров и переменных, неполноту и недетерминированность исходной информации, разнообразие и вероятностный характер воздействий внешней среды и т. д. Ограниченность возможностей экспериментального исследования больших систем делает актуальной разработку методики их моделирования, которая позволила бы в соответствующей форме представить процессы функционирования систем, описание протекания этих процессов с помощью математических моделей, получение результатов экспериментов с моделями по оценке характеристики исследуемых объектов. Причем на разных этапах создания и использования перечисленных систем для всего многообразия входящих в них подсистем применение метода моделирования преследует конкретные цели, а эффективность метода зависит от того, насколько грамотно разработчик использует возможности моделирования [34].

Независимо от разбиения конкретной сложной системы на подсистемы при проектировании каждой из них необходимо выполнить внешнее проектирование (макропроектирование) и внутреннее проектирование (микропроектирование). Так как на этих стадиях разработчик преследует различные цели, то и используемые при этом методы и средства моделирования могут существенно отличаться.

На стадии макропроектирования должна быть разработана обобщенная модель процесса функционирования сложной системы, позволяющая разработчику получить ответы на вопросы об эффективности различных стратегий управления объектом при его взаимодействии с внешней средой. Стадию внешнего проектирования можно разбить на анализ и синтез. При анализе изучают объект управления, строят модель воздействий внешней среды, определяют критерии оценки эффективности, имеющиеся ресурсы, необходимые ограничения. Конечная цель стадии анализа — построение модели объекта управления для оценки его характеристик. При синтезе на этапе внешнего проектирования решаются задачи выбора стратегии управления на основе модели объекта моделирования, т. е. сложной системы.

На стадии микропроектирования разрабатывают модели с целью создания эффективных подсистем. Причем используемые методы и средства моделирования зависят от того, какие конкретно обеспечивающие подсистемы разрабатываются: информационные, математические, технические, программные и т. д.

Особенности использования моделей. Выбор метода моделирования и необходимая детализация моделей существенно зависят от этапа разработки сложной системы [34, 37, 46]. На этапах обследования объекта управления, например промышленного предприятия, и разработки технического задания на проектирование автоматизированной системы управления модели в основном носят описательный характер и преследуют цель наиболее полно представить в компактной форме информацию об объекте, необходимую разработчику системы.

На этапах разработки технического и рабочего проектов систем, модели отдельных подсистем детализируются, и моделирование служит для решения конкретных задач проектирования, т. е. выбора оптимального по определенному критерию при заданных ограничениях варианта из множества допустимых. Поэтому в основном на этих этапах проектирования сложных систем используются модели для целей синтеза [10, 18, 37].

Целевое назначение моделирования на этапе внедрения и эксплуатации сложных систем — это проигрывание возможных ситуаций для принятия обоснованных и перспективных решений по управлению объектом. Моделирование (имитацию) также широко применяют при обучении и тренировке персонала автоматизированных систем управления, вычислительных комплексов и сетей, информационных систем в различных сферах. В этом случае моделирование носит характер деловых игр. Модель, реализуемая обычно на ЭВМ, воспроизводит поведение управляемого объекта и внешней среды, а люди в определенные моменты времени принимают решения по управлению объектом.

АСОИУ являются системами, которые развиваются по мере эволюции объекта управления, появления новых средств управления и т. д. Поэтому при прогнозировании развития сложных систем роль моделирования очень высока, так как это единственная возможность ответить на многочисленные вопросы о путях дальнейшего эффективного развития системы и выбора из них наиболее оптимального.

В.3. ПЕРСПЕКТИВЫ РАЗВИТИЯ МЕТОДОВ И СРЕДСТВ МОДЕЛИРОВАНИЯ СИСТЕМ В СВЕТЕ НОВЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

В последние годы основные достижения в различных областях науки и техники неразрывно связаны с процессом совершенствования ЭВМ. Сфера эксплуатации ЭВМ — бурно развивающаяся

отрасль человеческой практики, стимулирующая развитие новых теоретических и прикладных направлений [35]. Ресурсы современной информационно-вычислительной техники дают возможность ставить и решать математические задачи такой сложности, которые в недавнем прошлом казались нереализуемыми, например моделирование больших систем.

Аналитические и имитационные методы. Исторически первым сложился *аналитический подход* к исследованию систем, когда ЭВМ использовалась в качестве вычислителя по аналитическим зависимостям. Анализ характеристик процессов функционирования больших систем с помощью только аналитических методов исследования наталкивается обычно на значительные трудности, приводящие к необходимости существенного упрощения моделей либо на этапе их построения, либо в процессе работы с моделью, что может привести к получению недостоверных результатов.

Поэтому в настоящее время наряду с построением аналитических моделей большое внимание уделяется задачам оценки характеристик больших систем на основе *имитационных моделей*, реализованных на современных ЭВМ с высоким быстродействием и большим объемом оперативной памяти. Причем перспективность имитационного моделирования как метода исследования характеристик процесса функционирования больших систем возрастает с повышением быстродействия и оперативной памяти ЭВМ, с развитием математического обеспечения, совершенствованием банков данных и периферийных устройств для организации диалоговых систем моделирования. Это, в свою очередь, способствует появлению новых «чисто машинных» методов решения задач исследования больших систем на основе организации имитационных экспериментов с их моделями. Причем ориентация на автоматизированные рабочие места на базе персональных ЭВМ для реализации экспериментов с имитационными моделями больших систем позволяет проводить не только анализ их характеристик, но и решать задачи структурного, алгоритмического и параметрического синтеза таких систем при заданных критериях оценки эффективности и ограничениях [4, 9, 18, 23].

Достигнутые успехи в использовании средств вычислительной техники для целей моделирования часто создают иллюзию, что применение современной ЭВМ гарантирует возможность исследования системы любой сложности. При этом игнорируется тот факт, что в основу любой модели положено трудоемкое по затратам времени и материальных ресурсов предварительное изучение явлений, имеющих место в объекте-оригинале. И от того, насколько детально изучены реальные явления, насколько правильно проведена их формализация и алгоритмизация, зависит в конечном итоге успех моделирования конкретного объекта.

Средства моделирования систем. Расширение возможностей моделирования различных классов больших систем неразрывно связа-

но с совершенствованием средств вычислительной техники и техники связи. Перспективным направлением является создание для целей моделирования иерархических многомашинных вычислительных систем и сетей [2, 7, 12, 25, 41].

При создании больших систем их компоненты разрабатываются различными коллективами, которые используют средства моделирования при анализе и синтезе отдельных подсистем. При этом разработчикам необходим оперативный доступ к программно-техническим средствам моделирования, а также оперативный обмен результатами моделирования отдельных взаимодействующих подсистем. Таким образом, появляется необходимость в создании диалоговых систем моделирования, для которых характерны следующие особенности: возможность одновременной работы многих пользователей, занятых разработкой одной или нескольких систем, доступ пользователей к программно-техническим ресурсам системы моделирования, включая, базы данных и знаний, пакеты прикладных программ моделирования, обеспечение диалогового режима работы с различными вычислительными машинами и устройствами, включая цифровые и аналоговые вычислительные машины, установки натурального и физического моделирования, элементы реальных систем и т. п., диспетчирование работ в *системе моделирования* и оказание различных услуг пользователям, включая обучение работе с диалоговой системой моделирования при обеспечении дружелюбного интерфейса.

В зависимости от специфики исследуемых объектов в ряде случаев эффективным оказывается моделирование на аналоговых вычислительных машинах (АВМ). При этом надо иметь в виду, что АВМ значительно уступают ЭВМ по точности и логическим возможностям, но по быстродействию, схемной простоте реализации, сопрягаемости с датчиками внешней информации АВМ превосходят ЭВМ или по крайней мере не уступают им.

Для сложных динамических объектов перспективным является моделирование на базе гибридных (аналого-цифровых) вычислительных комплексов. Такие комплексы реализуют преимущества цифрового и аналогового моделирования и позволяют наиболее эффективно использовать ресурсы ЭВМ и АВМ в составе единого комплекса. При использовании гибридных моделирующих комплексов упрощаются вопросы взаимодействия с датчиками, установленными на реальных объектах, что позволяет, в свою очередь, проводить комбинированное моделирование с использованием аналого-цифровой части модели и натурной части объекта [11, 19]. Такие гибридные моделирующие комплексы могут входить в состав многомашинного вычислительного комплекса, что еще больше расширяет его возможности с точки зрения моделируемых классов больших систем.

Информационные технологии в обществе XXI века. Конец XX столетия ознаменовался интенсивным развитием и внедрением во

все сферы жизни общества *информатики*. Это проявилось в интенсивном совершенствовании средств вычислительной техники и техники связи, в появлении новых и в дальнейшем развитии существующих информационных технологий, а также в реализации прикладных информационных систем. Достижения информатики заняли достойное место в организационном управлении, в промышленности, в проведении научных исследований и в автоматизированном проектировании. Информатизация охватила и социальную сферу: образование, науку, культуру, здравоохранение.

Переход страны к рыночной экономике потребовал развития соответствующего информационного обеспечения. Постепенно в России формировался рынок, в котором информация начинала выступать как ресурс, имеющий коммерческий характер. Наряду с производством систем и средств информатики большое место в настоящее время занимают и информационные услуги, на базе самокупаемости интенсивно развивается отрасль связи. Телефония, радиовещание, телевидение работают с использованием различных типов каналов связи. Компьютерная техника прочно вошла в быт и используется как в образовании, так и в воспитании подрастающего поколения.

Домашний компьютер стал естественным для многих семей. В образовании значительная доля нагрузки в учебном процессе переносится на самостоятельные задания, выполняемые на домашнем компьютере. По своему качеству домашний компьютер в настоящее время часто оказывается намного выше компьютера, используемого в школе или в ВУЗе. Характерно, что в последние годы покупая домашний компьютер, пользователь начал обращать внимание на место сборки, конфигурацию и перспективные возможности техники. Приобретается и значительное количество программных средств, в том числе записанных на CD-ROM, огромное число абонентов подключается к *Интернет*, значительное число пользователей работает с использованием сотовой и других сетей. Все это подтверждает, что процесс информатизации интенсифицируется, завершается этап неуправляемой информатизации. Управляемая составляющая, которая реализовывалась в основном в образовании, в промышленности и в административном управлении оказалась явно недостаточной из-за малых финансовых средств, но в целом современный уровень информатизации позволяет констатировать, что начало следующего века станет точкой перехода из века энергетики в век информатики, как это прогнозировал Норберт Винер [8, 35].

Информатизация как процесс перехода от индустриального общества к информационному характеризуется резким перераспределением трудовых ресурсов в материальное производство и в сферу информации. Это соотношение изменяется от 3:1 к 1:3. В ряде стран суммарные расходы на компьютерную технику, телекоммуникации, электронику превысили расходы на энергетику, а поэтому, рассмат-

ривая проблему перспектив развития образования, нам необходимо исходить из будущего, поскольку только логически разработанная картина будущего может помочь познать настоящее. Проблема становления *информационного общества* и составляющая ее проблема информатизации образования должна рассматриваться в тесной взаимосвязи с проблемой будущего устойчивого развития цивилизации.

Модель образовательной системы должна быть сформирована с учетом адаптации образования к модели устойчивого развития цивилизации, а отсюда вытекает проблема опережающего развития образовательной системы, которая должна удовлетворять потребностям будущего информационного общества.

Для информационного общества характерно полное удовлетворение информационных потребностей населения при завершении формирования единой информационной среды, определяющей новую культуру как общества в целом, так и каждого человека в отдельности. *Информационная культура* как составляющая и базис информационного общества должна закладываться уже в настоящее время. Переход от консервативной образовательной системы к опережающей мог бы базироваться на опережающем формировании информационного пространства Российского образования. Только *образование* может служить фундаментом новой информационной культуры.

Информационная культура конечно не ограничивается системой знаний в области информационных процессов, технологий и должна включать активно преобразовательный аспект отношения к миру. По сути информационная культура может рассматриваться как свод правил поведения в информационном обществе, в коммуникационной среде, в человеко-машинных системах, вписывающихся в мировую гуманистическую культуру человечества. Уже в настоящее время вхождение пользователя в мировую сеть позволяет получить огромные объемы информации, которая может быть предназначена и для идеологической обработки.

Необходимо воспитывать корректное отношение к получаемой информации. Информационное пространство Российского образования должно отвечать национальным интересам и базироваться на традициях отечественной культуры. Повсеместное использование зарубежной компьютерной техники сопровождается планомерным информационным идеологическим воздействием на пользователей. Можно пойти по пути защиты от чуждой нам информации, создавая соответствующие методы и средства, но особое внимание нужно обратить на информационную культуру педагога, воспитателя, учителя школы и преподавателя ВУЗа.

Уровень информационной подготовки учителя нередко отстает от уровня ученика, работающего на домашнем компьютере, подключенном к сети. Проблема развития интеллекта учащихся не может быть решена только средствами информатики, но проблема

развития науки об образовании должна разрешаться с учетом перехода в информационное общество, в котором будет сформирована *инфоноосфера* личности, а поэтому развитие интеллектуальных способностей личности даже в настоящее время тесно смыкается с проблемой информатизации образования. Возможности информатизации образования определяются современными достижениями информатики и методологией их использования в образовании.

Можно выделить три уровня информатики:

— *физический* — программно-аппаратные средства вычислительной техники и техники связи;

— *логический* — информационные технологии;

— *прикладной* — пользовательские информационные системы.

Для физического уровня характерно, что компьютерная техника и техника связи практически вся разработана за рубежом и в лучшем случае наблюдается лишь ее сборка на отечественном производстве.

Информатизация города, региона, области базируется на создании единой телекоммуникационной среды. Отличительными особенностями перспективных сетей являются интеграция услуг, предоставляемых пользователю, цифровизация, комплексное использование проводных, радио- и космических каналов связи, переход к цифровым сетям интегрального обслуживания [35]. Использование волоконно-оптических линий и сетей кабельного телевидения позволяет на одной и той же базе обеспечить передачу речи, видеосигнала, данных, служебной информации и тем самым обеспечить вхождение каждого пользователя как в Российское, так и в мировое информационное пространство.

Происходит формирование единой информационной среды на основе объединения банков данных и баз знаний, проектируются конкретные *информационные системы* в различных областях человеческой деятельности. Совершенствование технической базы сопровождается продвижением современных операционных систем пользовательскую среду, развиваются открытые системы.

Модели базовых информационных технологий в образовании. Для логического уровня информатики характерно совершенствование существующих, создание и развитие *новых информационных технологий*. Получили развитие как теория, так и практика информационных технологий [1, 8, 35]. Развивается методология, совершенствуются средства информационных технологий. Уже в настоящее время могут быть выделены базовые информационные процессы и *информационные технологии*.

В рамках базовых технологий получают развитие конкретные технологии, решающие задачи в выбранных предметных областях. Переход к информационному обществу заставляет задуматься о готовности выпускников учебных заведений к жизни и к труду в обществе XXI века. Учитывая, что уже в настоящее время скорости

преобразования технологий производства стали опережать темпы смены поколений, оказывается необходимым не только совершенствование и дополнительная подготовка, но и неоднократное освоение новых видов деятельности в течение трудовой жизни.

Поэтому в информационном обществе встает проблема обучения, и непрерывное образование становится составной частью жизни каждого человека. В этих условиях информатизация означает изменение всей образовательной системы с ее ориентацией на новую информационную культуру. Освоение новой информационной культуры может в значительной степени реализовываться за счет внедрения в учебный процесс, управление образованием и в повседневную жизнь перспективных информационных технологий.

Прежде всего следует обратить особое внимание на проблему обеспечения сферы образования теорией и методикой как разработки, так и эффективного применения новых средств информационных технологий. Теория информационных технологий должна определить модели базовых информационных процессов, связанных с получением, сбором, передачей, обработкой, хранением, накоплением и представлением информации. Особое место занимают *модели формализации и представления знаний*.

Весьма актуальным представляется выделение базовых информационных технологий, к которым уже в настоящее время можно отнести технологии распределенного хранения и обработки, офисные технологии, мультимедиа технологии, геоинформационные технологии, технологии защиты информации, CASE-технологии, телекоммуникационные технологии [15, 35]. На основе базовых разрабатываются *прикладные информационные технологии* по областям применения, позволяющие получать конкретные продукты соответствующего назначения в виде средств, систем, сред.

В рамках указанных технологий в образовании уже в настоящее время получили широкое применение:

1) компьютерные программы и обучающие системы, представляющие собой *электронные учебники*, учебные пособия, тренажеры, лабораторные практикумы, системы тестирования знаний и квалификации, выполненные на различных типах машинных носителей;

2) системы на базе *мультимедиа-технологии*, построенные с применением видеотехники, накопителей на CD-ROM и реализуемые на ПЭВМ;

3) *интеллектуальные обучающие экспертные системы*, которые специализируются по конкретным областям применения и имеют практическое значение как в процессе обучения, так и в учебных исследованиях;

4) информационные среды на основе *баз данных и знаний*, позволяющие осуществить как прямой, так и удаленный доступ к информационным ресурсам;

5) телекоммуникационные системы, реализующие электронную почту, телеконференции и т. д. и позволяющие осуществить выход в мировые коммуникационные сети;

6) *электронные настольные типографии*, позволяющие в индивидуальном режиме с высокой скоростью осуществить производство учебных пособий и документов на различных носителях;

7) *электронные библиотеки* как распределенного, так и централизованного характера, позволяющие по-новому реализовать доступ учащихся к мировым информационным ресурсам;

8) *геоинформационные системы*, которые базируются на технологиях объединения компьютерной картографии и систем управления базами данных. В итоге удается создать многослойные электронные карты, опорный слой которых описывает базовые явления или ситуации, а каждый последующий — задает один из аспектов, процессов или явлений;

9) *системы защиты информации* различной ориентации (от несанкционированного доступа при хранении информации, от искажений при передаче информации, от подслушивания и т. д.).

Перспективы применения информационных технологий. Методически новые информационные технологии в образовании должны быть проработаны с ориентацией на конкретное применение. Часть технологий может поддерживать учебный процесс (лекционные и практические занятия), другие технологии способны эффективно поддержать разработку новых учебников и учебных пособий. Информационные технологии помогут также эффективно организовать проведение экспериментально-исследовательских работ как в школе, так и в ВУЗе. Особую значимость информационные технологии приобретают при самостоятельной работе учащихся на домашнем компьютере с использованием *современных методов моделирования*.

Какие же новые возможности открываются при внедрении современных информационных технологий в образование? На основе мультимедиа технологии появляется возможность создавать учебники, учебные пособия и другие методические материалы на машинном носителе, которые могут быть разделены на некоторые группы:

1. *Учебники*, представляющие собой текстовое изложение материала с большим количеством иллюстраций, которые могут быть установлены на сервере и переданы через сеть на домашний компьютер. При ограниченном количестве материала такой учебник может быть реализован в прямом доступе пользователя к серверу.

2. *Учебники с высокой динамикой* иллюстративного материала, выполненные на CD-ROM. Наряду с основным материалом они содержат средства интерактивного доступа, средства анимации и мультипликации, а также видеозображения, в динамике демон-

стрирующие принципы и способы реализации отдельных процессов и явлений. Такие учебники могут иметь не только образовательное, но и художественное назначение. Огромный объем памяти носителя информации позволяет реализовывать на одном оптическом диске энциклопедию, справочник, путеводитель и т. д.

3. *Современные компьютерные обучающие системы* для проведения учебно-исследовательских работ. Они реализовывают моделирование как процессов, так и явлений, т. е. создают новую учебную компьютерную среду, в которой обучаемый является активным, и может сам вести учебный процесс.

4. *Системы виртуальной реальности*, в которых учащийся становится участником компьютерной модели, отображающей окружающий мир. Для грамотного использования мультимедиа продуктов этого типа крайне важно изучение их психологических особенностей и негативных воздействий на обучаемого.

5. *Системы дистанционного обучения*. В сложных социально-экономических условиях дистанционное образование становится особенно актуальным для отдаленных регионов, для людей с малой подвижностью, а также при самообразовании и самостоятельной работе учащихся. Эффективная реализация дистанционного обучения возможна лишь при целенаправленной программе создания высококачественных мультимедиа продуктов учебного назначения по фундаментальным, естественнонаучным, общепрофессиональным и специальным дисциплинам.

К сожалению, это требует значительных финансовых средств и пока не окупается на коммерческой основе, необходимы существенные бюджетные ассигнования в эту область. Реализация такой программы позволит по-новому организовать учебный процесс, увеличив нагрузку на самостоятельную работу обучаемого.

Формирование новой информационной культуры должно базироваться прежде всего на определенном уровне обучения в школе, а поэтому особое внимание следует уделить содержанию программы базового курса информатики, который, с одной стороны, должен быть согласован по содержанию с последующим обучением в ВУЗе, а с другой, должен поддерживать и остальные предметы школьного образования. В курс информатики уже в настоящее время закладываются сведения по моделированию процессов и явлений, по методологии формирования информационных моделей окружающего мира. У учащихся должна возникать в процессе познания информационная картина мира. Это невозможно без формирования информационной культуры населения. В основу создания информационной культуры нового общества должна быть положена идея компьютерной поддержки каждого изучаемого предмета, нельзя подменить это изучением единственного курса информатики.

Весьма важным является принцип непрерывности информационной подготовки учащихся, который должен соблюдаться как на

стадии школьного, так и при переходе от школьного к ВУЗовскому уровню. В структуре ВУЗовского образования информатика является фундаментальной дисциплиной. Наряду с информатикой в учебном плане специальностей может предусматриваться ряд курсов информационной подготовки даже для нетехнических ВУЗов, которые должны совершенствоваться, чтобы компьютер стал естественным орудием труда в любой предметной области деятельности выпускника ВУЗа. К информационной подготовке можно отнести обучение методологии и средствам моделирования. Создание опережающей информационной среды непрерывного Российского образования требует и решения ряда методических и организационных проблем, в том числе следующие:

1. Принятие единой системы программно и аппаратно совместимых *средств вычислительной техники и техники связи*, используемой в непрерывном учебном процессе. Это требует сертификации используемых средств учебного назначения и реализации программ по созданию сертификационных центров и эффективному их использованию.

2. Подключение образовательных организаций к единой цифровой сети в последующем выходом в *Интернет*. Решение этой задачи в значительной степени реализуется в настоящее время в высшем образовании и сдерживается в школьном образовании по финансовым причинам, а также и по сложностям выполнения для отдаленных районов.

3. Формирование *единой информационной среды* непрерывного образования с созданием баз данных по направлениям и специальностям подготовки, которые бы включали в себя методические документы, энциклопедии, справочники, учебники и учебные пособия, а также дополнительные средства, поддерживающие учебный процесс. Актуальным является представление в международной сети наших достижений и возможностей. Необходима организация обмена информационными ресурсами Российской образовательной системы с международной.

4. Необходимо совершенствование *инструментальных средств* непрерывного образования, ориентированных на ускоренное освоение материала и приобретение устойчивых навыков обучаемых, а также преследующих цели индивидуального обучения. Сюда можно отнести перспективные программные оболочки по разработке компьютерных учебников и методических материалов, программные и аппаратные средства создания компьютерных обучающих систем, средства технологии разработки мультимедиа продуктов, геоинформационных систем и т. д.

5. Необходима организация *инфраструктуры* информатизации образования как составной части информатизации общества в целом. Эта структура должна обеспечить создание новых, тиражирование и внедрение существующих информационных технологий в непрерывное образование.

Идеологически при информатизации образования необходимо учитывать ряд принципиальных позиций:

- Эволюционное развитие сложившейся методологии образования за счет явных преимуществ новых информационных технологий, а именно, возможность наглядного, динамичного представления информатизации с использованием видеоизображений и звука, применения удаленного доступа для ознакомления с внешним и внесения собственного информационного ресурса в образование.

- Непрерывность и преемственность компьютерного образования на всех уровнях обучения от дошкольного до послевузовского. Непрерывность может быть обеспечена компьютерной поддержкой всех предметов и дисциплин учебного процесса.

- Обеспечение свободы выбора методики, стиля и средств обучения с целью выявления творческих индивидуальных способностей обучаемого в сочетании с возможностью их коллективной деятельности на основе информационных технологий и телекоммуникационных систем.

- Создание научно и методически основанной системы базового образования на основе компьютерных технологий. Одним из реальных путей решения проблемы в целом является формирование и реализация региональных научно-технических программ с долевым федеральным и местным бюджетным финансированием при дополнительном использовании внебюджетных средств. Предметом специальных исследований коллективов Высшей школы должны стать содержание, методы и средства развития образования как опережающей системы в будущем информационном обществе. При этом фундаментальное место занимают *методы и средства моделирования*, на основе которых можно предсказать будущее. Только при устойчивом развитии цивилизации мы можем надеяться на последовательное становление ноосферы как сферы разума. Будущее развитие человечества должно быть управляемым и в этом аспекте, несомненно, управляемым должно быть и развитие образования.

Контрольные вопросы

- В.1. Что такое модель системы?
- В.2. Как определяется понятие «моделирование»?
- В.3. Что называется гипотезой и аналогией в исследовании систем?
- В.4. Чем отличается использование метода моделирования при внешнем и внутреннем проектировании систем?
- В.5. Какие современные средства вычислительной техники используются для моделирования систем?

ГЛАВА 1

ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ МОДЕЛИРОВАНИЯ СИСТЕМ

Моделирование начинается с формирования предмета исследований — системы понятий, отражающей существенные для моделирования характеристики объекта. Эта задача является достаточно сложной, что подтверждается различной интерпретацией в научно-технической литературе таких фундаментальных понятий, как система, модель, моделирование. Подобная неоднозначность не говорит об ошибочности одних и правильности других терминов, а отражает зависимость предмета исследований (моделирования) как от рассматриваемого объекта, так и от целей исследователя. Отличительной особенностью моделирования сложных систем является его многофункциональность и многообразие способов использования; оно становится неотъемлемой частью всего жизненного цикла системы. Объясняется это в первую очередь технологичностью моделей, реализованных на базе средств вычислительной техники: достаточно высокой скоростью получения результатов моделирования и их сравнительно невысокой себестоимостью.

1.1. ПРИНЦИПЫ СИСТЕМНОГО ПОДХОДА В МОДЕЛИРОВАНИИ СИСТЕМ

В настоящее время при анализе и синтезе сложных (больших) систем получил развитие *системный подход*, который отличается от классического (или индуктивного) подхода. Последний рассматривает систему путем перехода от частного к общему и синтезирует (конструирует) систему путем слияния ее компонент, разрабатываемых раздельно. В отличие от этого системный подход предполагает последовательный переход от общего к частному, когда в основе рассмотрения лежит цель, причем исследуемый объект выделяется из окружающей среды.

Объект моделирования. Специалисты по проектированию и эксплуатации сложных систем имеют дело с системами управления различных уровней, обладающими общим свойством — стремлением достичь некоторой цели. Эту особенность учтем в следующих определениях системы. *Система S* — целенаправленное множество взаимосвязанных элементов любой природы. *Внешняя среда E* — множество существующих вне системы элементов любой природы, оказывающих влияние на систему или находящихся под ее воздействием.

В зависимости от цели исследования могут рассматриваться разные соотношения между самим объектом S и внешней средой E . Таким образом, в зависимости от уровня, на котором находится наблюдатель, объект исследования может выделяться по-разному и могут иметь место различные взаимодействия этого объекта с внешней средой.

С развитием науки и техники сам объект непрерывно усложняется, и уже сейчас говорят об объекте исследования как о некоторой сложной системе, которая состоит из различных компонент, взаимосвязанных друг с другом. Поэтому, рассматривая системный подход как основу для построения больших систем и как базу создания методики их анализа и синтеза, прежде всего необходимо определить само понятие системного подхода.

Системный подход — это элемент учения об общих законах развития природы и одно из выражений диалектического учения. Можно привести разные определения системного подхода, но наиболее правильно то, которое позволяет оценить познавательную сущность этого подхода при таком методе исследования систем, как моделирование. Поэтому весьма важны выделение самой системы S и внешней среды E из объективно существующей реальности и описание системы исходя из общесистемных позиций.

При системном подходе к моделированию систем необходимо прежде всего четко определить цель моделирования. Поскольку невозможно полностью смоделировать реально функционирующую систему (систему-оригинал, или первую систему), создается модель (система-модель, или вторая система) под поставленную проблему. Таким образом, применительно к вопросам моделирования цель возникает из требуемых задач моделирования, что позволяет подойти к выбору критерия и оценить, какие элементы войдут в создаваемую модель M . Поэтому необходимо иметь критерий отбора отдельных элементов в создаваемую модель.

Подходы к исследованию систем. Важным для системного подхода является определение *структуры системы* — совокупности связей между элементами системы, отражающих их взаимодействие. Структура системы может изучаться извне с точки зрения состава отдельных подсистем и отношений между ними, а также изнутри, когда анализируются отдельные свойства, позволяющие системе достигать заданной цели, т. е. когда изучаются функции системы. В соответствии с этим наметился ряд подходов к исследованию структуры системы с ее свойствами, к которым следует прежде всего отнести структурный и функциональный.

При *структурном подходе* выявляются состав выделенных элементов системы S и связи между ними. Совокупность элементов и связей между ними позволяет судить о структуре системы. Последняя в зависимости от цели исследования может быть описана на

разных уровнях рассмотрения. Наиболее общее описание структуры — это топологическое описание, позволяющее определить в самых общих понятиях составные части системы и хорошо формализуемое на базе теории графов.

Менее общим является функциональное описание, когда рассматриваются отдельные функции, т. е. алгоритмы поведения системы, и реализуется *функциональный подход*, оценивающий функции, которые выполняет система, причем под функцией понимается свойство, приводящее к достижению цели. Поскольку функция отображает свойство, а свойство отображает взаимодействие системы S с внешней средой E , то свойства могут быть выражены в виде либо некоторых характеристик элементов S_{ij} и подсистем S_i системы, либо системы S в целом.

При наличии некоторого эталона сравнения можно ввести количественные и качественные характеристики систем. Для количественной характеристики вводятся числа, выражающие отношения между данной характеристикой и эталоном. Качественные характеристики системы находятся, например, с помощью метода экспертных оценок.

Проявление функций системы во времени $S(t)$, т. е. *функционирование системы*, означает переход системы из одного состояния в другое, т. е. движение в пространстве состояний Z . При эксплуатации системы S весьма важно качество ее функционирования, определяемое показателем эффективности и являющееся значением критерия оценки эффективности. Существуют различные подходы к выбору критериев оценки эффективности. Система S может оцениваться либо совокупностью частных критериев, либо некоторым общим интегральным критерием.

Следует отметить, что создаваемая модель M с точки зрения системного подхода также является системой, т. е. $S' = S'(M)$, и может рассматриваться по отношению к внешней среде E . Наиболее просты по представлению модели, в которых сохраняется прямая аналогия явления. Применяют также модели, в которых нет прямой аналогии, а сохраняются лишь законы и общие закономерности поведения элементов системы S . Правильное понимание взаимосвязей как внутри самой модели M , так и взаимодействия ее с внешней средой E в значительной степени определяется тем, на каком уровне находится наблюдатель.

Простой подход к изучению взаимосвязей между отдельными частями модели предусматривает рассмотрение их как отражение связей между отдельными подсистемами объекта. Такой классический подход может быть использован при создании достаточно простых моделей. Процесс синтеза модели M на основе классического (индуктивного) подхода представлен на рис. 1.1, а. Реальный объект, подлежащий моделированию, разбивается на отдельные подсистемы, т. е. выбираются исходные данные D для

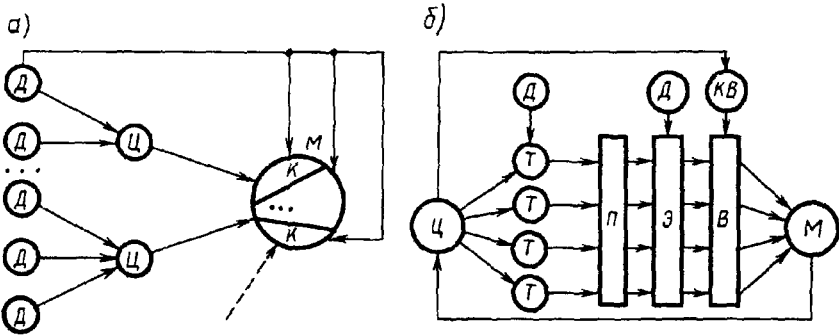


Рис. 1.1. Процесс синтеза модели на основе классического (а) и системного (б) подходов

моделирования и ставятся цели $Ц$, отображающие отдельные стороны процесса моделирования. По отдельной совокупности исходных данных $Д$ ставится цель моделирования отдельной стороны функционирования системы, на базе этой цели формируется некоторая компонента $К$ будущей модели. Совокупность компонент объединяется в модель $М$.

Таким образом, разработка модели $М$ на базе классического подхода означает суммирование отдельных компонентов в единую модель, причем каждая из компонент решает свои собственные задачи и изолирована от других частей модели. Поэтому классический подход может быть использован для реализации сравнительно простых моделей, в которых возможно разделение и взаимно независимое рассмотрение отдельных сторон функционирования реального объекта. Для модели сложного объекта такая разобщенность решаемых задач недопустима, так как приводит к значительным затратам ресурсов при реализации модели на базе конкретных программно-технических средств. Можно отметить две отличительные стороны классического подхода: наблюдается движение от частного к общему, создаваемая модель (система) образуется путем суммирования отдельных ее компонент и не учитывается возникновение нового системного эффекта.

С усложнением объектов моделирования возникла необходимость наблюдения их с более высокого уровня. В этом случае наблюдатель (разработчик) рассматривает данную систему S как некоторую подсистему какой-то метасистемы, т. е. системы более высокого ранга, и вынужден перейти на позиции нового системного подхода, который позволит ему построить не только исследуемую систему, решающую совокупность задач, но и создавать систему, являющуюся составной частью метасистемы. Например, если ставится задача проектирования АСУ предприятием, то с позиции системного подхода нельзя забывать

о том, что эта система является составной частью АСУ объединением.

Системный подход получил применение в системотехнике в связи с необходимостью исследования больших реальных систем, когда сказалась недостаточность, а иногда ошибочность принятия каких-либо частных решений. На возникновение системного подхода повлияли увеличивающееся количество исходных данных при разработке, необходимость учета сложных стохастических связей в системе и воздействий внешней среды E . Все это заставило исследователей изучать сложный объект не изолированно, а во взаимодействии с внешней средой, а также в совокупности с другими системами некоторой метасистемы.

Системный подход позволяет решить проблему построения сложной системы с учетом всех факторов и возможностей, пропорциональных их значимости, на всех этапах исследования системы S и построения модели M . Системный подход означает, что каждая система S является интегрированным целым даже тогда, когда она состоит из отдельных разобщенных подсистем. Таким образом, в основе системного подхода лежит рассмотрение системы как интегрированного целого, причем это рассмотрение при разработке начинается с главного — формулировки цели функционирования. Процесс синтеза модели M на базе системного подхода условно представлен на рис. 1.1, б. На основе исходных данных D , которые известны из анализа внешней системы, тех ограничений, которые накладываются на систему сверху либо исходя из возможности ее реализации, и на основе цели функционирования формулируются исходные требования T к модели системы S . На базе этих требований формируются ориентировочно некоторые подсистемы P , элементы \mathcal{E} и осуществляется наиболее сложный этап синтеза — выбор B составляющих системы, для чего используются специальные критерии выбора KB .

При моделировании необходимо обеспечить максимальную эффективность модели системы. Эффективность обычно определяется как некоторая разность между какими-то показателями ценности результатов, полученных в итоге эксплуатации модели, и теми затратами, которые были вложены в ее разработку и создание.

Стадии разработки моделей. На базе системного подхода может быть предложена и некоторая последовательность разработки моделей, когда выделяют две основные стадии проектирования: макропроектирование и микропроектирование.

На стадии макропроектирования на основе данных о реальной системе S и внешней среде E строится модель внешней среды, выявляются ресурсы и ограничения для построения модели системы, выбирается модель системы и критерии, позволяющие оценить адекватность модели M реальной системы S . Построив модель системы и модель внешней среды, на основе критерия эффективности функционирования системы в процессе модели-

рования выбирают оптимальную стратегию управления, что позволяет реализовать возможности модели по воспроизведению отдельных сторон функционирования реальной системы S .

Стадия микропроектирования в значительной степени зависит от конкретного типа выбранной модели. В случае имитационной модели необходимо обеспечить создание информационного, математического, технического и программного обеспечений системы моделирования. На этой стадии можно установить основные характеристики созданной модели, оценить время работы с ней и затраты ресурсов для получения заданного качества соответствия модели процессу функционирования системы S .

Независимо от типа используемой модели M при ее построении необходимо руководствоваться рядом принципов системного подхода: 1) пропорционально-последовательное продвижение по этапам и направлениям создания модели; 2) согласование информационных, ресурсных, надежностных и других характеристик; 3) правильное соотношение отдельных уровней иерархии в системе моделирования; 4) целостность отдельных обособленных стадий построения модели.

Модель M должна отвечать заданной цели ее создания, поэтому отдельные части должны компоноваться взаимно, исходя из единой системной задачи. Цель может быть сформулирована качественно, тогда она будет обладать большей содержательностью и длительное время может отображать объективные возможности данной системы моделирования. При количественной формулировке цели возникает целевая функция, которая точно отображает наиболее существенные факторы, влияющие на достижение цели.

Построение модели относится к числу системных задач, при решении которых синтезируют решения на базе огромного числа исходных данных, на основе предложений больших коллективов специалистов. Использование системного подхода в этих условиях позволяет не только построить модель реального объекта, но и на базе этой модели выбрать необходимое количество управляющей информации в реальной системе, оценить показатели ее функционирования и тем самым на базе моделирования найти наиболее эффективный вариант построения и выгодный режим функционирования реальной системы S .

1.2. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОБЛЕМЫ МОДЕЛИРОВАНИЯ СИСТЕМ

С развитием системных исследований, с расширением экспериментальных методов изучения реальных явлений все большее значение приобретают абстрактные методы, появляются новые научные дисциплины, автоматизируются элементы умственного труда. Важ-

ное значение при создании реальных систем S имеют математические методы анализа и синтеза, целый ряд открытий базируется на чисто теоретических изысканиях. Однако было бы неправильно забывать о том, что основным критерием любой теории является практика, и даже сугубо математические, отвлеченные науки базируются в своей основе на фундаменте практических знаний.

Экспериментальные исследования 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. Если раньше моделирование означало реальный физический эксперимент либо построение макета, имитирующего реальный процесс, то в настоящее время появились новые виды моделирования, в основе которых лежит постановка не только физических, но также и математических экспериментов.

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. В широком смысле под *экспериментом* можно понимать некоторую процедуру организации и наблюдения каких-то явлений, которые осуществляются в условиях, близких к естественным, либо имитируют их.

Различают *пассивный эксперимент*, когда исследователь наблюдает протекающий процесс, и *активный*, когда наблюдатель вмешивается и организует протекание процесса. В последнее время распространены активный эксперимент, поскольку именно на его основе удается выявить критические ситуации, получить наиболее интересные закономерности, обеспечить возможность повторения эксперимента в различных точках и т. д.

В основе любого вида моделирования лежит некоторая *модель*, имеющая соответствие, базирующееся на некотором общем качестве, которое характеризует реальный объект. Объективно реальный объект обладает некоторой формальной структурой, поэтому для любой модели характерно наличие некоторой структуры, соответствующей формальной структуре реального объекта, либо изучаемой стороне этого объекта.

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 [36,37].

Характеристики моделей систем. В качестве объекта моделирования выступают сложные организационно-технические системы, которые можно отнести к классу больших систем. Более того, по своему содержанию и созданная модель M также становится системой $S(M)$ и тоже может быть отнесена к классу больших систем, для которых характерно следующее [35, 37].

1. Цель функционирования, которая определяет степень целенаправленности поведения модели M . В этом случае модели могут быть разделены на одноцелевые, предназначенные для решения одной задачи, и многоцелевые, позволяющие разрешить или рассмотреть ряд сторон функционирования реального объекта.

2. Сложность, которую, учитывая, что модель M является совокупностью отдельных элементов и связей между ними, можно оценить по общему числу элементов в системе и связей между ними. По разнообразию элементов можно выделить ряд уровней иерархии, отдельные функциональные подсистемы в модели M , ряд входов и выходов и т. д., т. е. понятие сложности может быть идентифицировано по целому ряду признаков.

3. Целостность, указывающая на то, что создаваемая модель M является одной целостной системой $S(M)$, включает в себя большое количество составных частей (элементов), находящихся в сложной взаимосвязи друг с другом.

4. Неопределенность, которая проявляется в системе: по состоянию системы, возможности достижения поставленной цели, методам решения задач, достоверности исходной информации и т. д. Основной характеристикой неопределенности служит такая мера информации, как энтропия, позволяющая в ряде случаев оценить количество управляющей информации, необходимой для достижения заданного состояния системы. При моделировании основная цель — получение требуемого соответствия модели реальному объекту и в этом смысле количество управляющей информации в модели можно также оценить с помощью энтропии и найти то предельное минимальное количество, которое необходимо для

получения требуемого результата с заданной достоверностью. Таким образом, понятие неопределенности, характеризующее большую систему, применимо к модели M и является одним из ее основных признаков [35].

5. Поведенческая страта, которая позволяет оценить эффективность достижения системой поставленной цели. В зависимости от наличия случайных воздействий можно различать детерминированные и стохастические системы, по своему поведению — непрерывные и дискретные и т. д. Поведенческая страта рассмотрение системы S позволяет применительно к модели M оценить эффективность построенной модели, а также точность и достоверность полученных при этом результатов. Очевидно, что поведение модели M не обязательно совпадает с поведением реального объекта, причем часто моделирование может быть реализовано на базе иного материального носителя [44].

6. Адаптивность, которая является свойством высокоорганизованной системы. Благодаря адаптивности удается приспособиться к различным внешним возмущающим факторам в широком диапазоне изменения воздействий внешней среды. Применительно к модели существенна возможность ее адаптации в широком спектре возмущающих воздействий, а также изучение поведения модели в изменяющихся условиях, близких к реальным. Надо отметить, что существенным может оказаться вопрос устойчивости модели к различным возмущающим воздействиям. Поскольку модель M — сложная система, весьма важны вопросы, связанные с ее существованием, т. е. вопросы живучести, надежности и т. д. [50, 54].

7. Организационная структура системы моделирования, которая во многом зависит от сложности модели и степени совершенства средств моделирования. Одним из последних достижений в области моделирования можно считать возможность использования имитационных моделей для проведения машинных экспериментов. Необходимы оптимальная организационная структура комплекса технических средств, информационного, математического и программного обеспечений системы моделирования $S'(M)$, оптимальная организация процесса моделирования, поскольку следует обращать особое внимание на время моделирования и точность получаемых результатов.

8. Управляемость модели, вытекающая из необходимости обеспечивать управление со стороны экспериментаторов для получения возможности рассмотрения протекания процесса в различных условиях, имитирующих реальные. В этом смысле наличие многих управляемых параметров и переменных модели в реализованной системе моделирования дает возможность поставить широкий эксперимент и получить обширный спектр результатов [16, 45]. Управляемость системы тесно связана и со степенью автоматизации моделирования. В настоящее время получили применение системы моделирования, отличающиеся высокой степенью автоматизации

процесса моделирования, когда наряду с программными средствами управления машинным моделированием используется возможность мультимедийного общения исследователя с процессом моделирования.

9. Возможность развития модели, которая исходя из современного уровня науки и техники позволяет создавать мощные системы моделирования $S(M)$ для исследования многих сторон функционирования реального объекта. Однако нельзя при создании системы моделирования ограничиваться только задачами сегодняшнего дня. Необходимо предусматривать возможность развития системы моделирования как по горизонтали в смысле расширения спектра изучаемых функций, так и по вертикали в смысле расширения числа подсистем, т. е. созданная система моделирования должна позволять применять новые современные методы и средства. Естественно, что интеллектуальная система моделирования может функционировать только совместно с коллективом людей, поэтому к ней предъявляют эргономические требования [45, 50, 54].

Цели моделирования систем. Одним из наиболее важных аспектов построения систем моделирования является проблема цели. Любую модель строят в зависимости от цели, которую ставит перед ней исследователь, поэтому одна из основных проблем при моделировании — это проблема целевого назначения. Подобие процесса, протекающего в модели M , реальному процессу является не целью, а условием правильного функционирования модели, и поэтому в качестве цели должна быть поставлена задача изучения какой-либо стороны функционирования объекта.

Для упрощения модели M цели делят на подцели и создают более эффективные виды моделей в зависимости от полученных подцелей моделирования. Можно указать целый ряд примеров целей моделирования в области сложных систем. Например, для АСУ предприятием весьма существенно изучение процессов оперативного управления производством, оперативно-календарного планирования, перспективного планирования и здесь также могут быть успешно использованы методы моделирования [11, 35, 37].

Если цель моделирования ясна, то возникает следующая проблема, а именно проблема построения модели M . Построение модели оказывается возможным, если имеется информация или выдвинуты гипотезы относительно структуры, алгоритмов и параметров исследуемого объекта. На основании их изучения осуществляется идентификация объекта. В настоящее время широко применяют различные способы оценки параметров: по методу наименьших квадратов, по методу максимального правдоподобия, байесовские, марковские оценки [10, 13, 18, 22].

Если модель M построена, то следующей проблемой можно считать проблему работы с ней, т. е. реализацию модели, основные

задачи которой — минимизация времени получения конечных результатов и обеспечение их достоверности.

Для правильно построенной модели M характерным является то, что она выявляет лишь те закономерности, которые нужны исследователю, и не рассматривает свойства системы S , не существенные для данного исследования. Следует отметить, что оригинал и модель должны быть одновременно сходны по одним признакам и различны по другим, что позволяет выделить наиболее важные изучаемые свойства. В этом смысле модель выступает как некоторый «заместитель» оригинала, обеспечивающий фиксацию и изучение лишь некоторых свойств реального объекта.

Таким образом, характеризуя проблему моделирования в целом, необходимо учитывать, что от постановки задачи моделирования до интерпретации полученных результатов существует большая группа сложных научно-технических проблем, к основным из которых можно отнести следующие: идентификацию реальных объектов, выбор вида моделей, построение моделей и их машинную реализацию, взаимодействие исследователя с моделью в ходе машинного эксперимента, проверку правильности полученных в ходе моделирования результатов, выявление основных закономерностей, исследованных в процессе моделирования. В зависимости от объекта моделирования и вида используемой модели эти проблемы могут иметь разную значимость.

В одних случаях наиболее сложной оказывается идентификация, в других — проблема построения формальной структуры объекта. Возможны трудности и при реализации модели, особенно в случае имитационного моделирования больших систем. При этом следует подчеркнуть роль исследователя в процессе моделирования. Постановка задачи, построение содержательной модели реального объекта во многом представляют собой творческий процесс и базируются на эвристике. И в этом смысле нет формальных путей выбора оптимального вида модели. Часто отсутствуют формальные методы, позволяющие достаточно точно описать реальный процесс. Поэтому выбор той или иной аналогии, выбор того или иного математического аппарата моделирования полностью основывается на имеющемся опыте исследователя и ошибка исследователя может привести к ошибочным результатам моделирования [37, 46].

Средства вычислительной техники, которые в настоящее время широко используются либо для вычислений при аналитическом моделировании, либо для реализации имитационной модели системы, могут лишь помочь с точки зрения эффективности реализации сложной модели, но не позволяют подтвердить правильность той или иной модели. Только на основе обработанных данных, опыта исследователя можно с достоверностью оценить адекватность модели по отношению к реальному процессу.

Если в ходе моделирования существенное место занимает реальный физический эксперимент, то здесь весьма важна и надежность используемых инструментальных средств, поскольку сбои и отказы программно-технических средств могут приводить к искаженным значениям выходных данных, отображающих протекание процесса. И в этом смысле при проведении физических экспериментов необходимы специальная аппаратура, специально разработанное математическое и информационное обеспечение, которые позволяют реализовать диагностику средств моделирования, чтобы отсеять те ошибки в выходной информации, которые вызваны неисправностями функционирующей аппаратуры. В ходе машинного эксперимента могут иметь место и ошибочные действия человека-оператора. В этих условиях серьезные задачи стоят в области эргономического обеспечения процесса моделирования.

1.3. КЛАССИФИКАЦИЯ ВИДОВ МОДЕЛИРОВАНИЯ СИСТЕМ

В основе моделирования лежит *теория подобия*, которая утверждает, что абсолютное подобие может иметь место лишь при замене одного объекта другим точно таким же [5]. При моделировании абсолютное подобие не имеет места и стремятся к тому, чтобы модель достаточно хорошо отображала исследуемую сторону функционирования объекта.

Классификационные признаки. В качестве одного из первых признаков классификации видов моделирования можно выбрать степень полноты модели и разделить модели в соответствии с этим признаком на полные, неполные и приближенные. В основе полного моделирования лежит полное подобие, которое проявляется как во времени, так и в пространстве. Для неполного моделирования характерно неполное подобие модели изучаемому объекту. В основе приближенного моделирования лежит приближенное подобие, при котором некоторые стороны функционирования реального объекта не моделируются совсем [5, 36, 37]. Классификация видов моделирования систем S приведена на рис. 1.2.

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

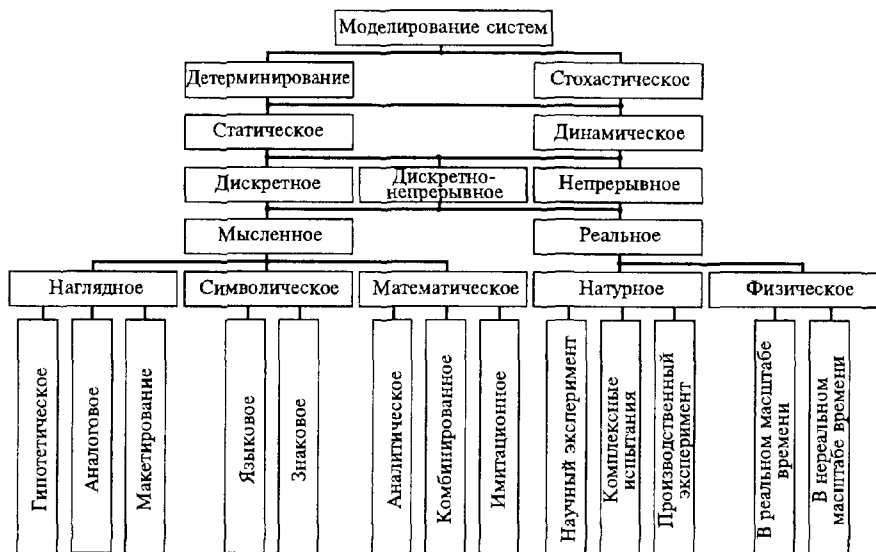


Рис. 1.2. Классификация видов моделирования систем

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 хотя бы выделить наличие как дискретных, так и непрерывных процессов.

В зависимости от формы представления объекта (системы S) можно выделить мысленное и реальное моделирование.

Мысленное моделирование часто является единственным способом моделирования объектов, которые либо практически нереализуемы в заданном интервале времени, либо существуют вне условий, возможных для их физического создания. Например, на базе мысленного моделирования могут быть проанализированы многие ситуации микромира, которые не поддаются физическому эксперименту. Мысленное моделирование может быть реализовано в виде наглядного, символического и математического.

При *наглядном моделировании* на базе представлений человека о реальных объектах создаются различные наглядные модели, отражающие явления и процессы, протекающие в объекте. В основу *гипотетического моделирования* исследователем закладывается некоторая гипотеза о закономерностях протекания процесса в реальном объекте, которая отражает уровень знаний исследователя об объекте и базируется на причинно-следственных связях между входом и выходом изучаемого объекта. Гипотетическое моделирова-

ние используется, когда знаний об объекте недостаточно для построения формальных моделей.

Аналоговое моделирование основывается на применении аналогий различных уровней. Наивысшим уровнем является полная аналогия, имеющая место только для достаточно простых объектов. С усложнением объекта используют аналогии последующих уровней, когда аналоговая модель отображает несколько либо только одну сторону функционирования объекта.

Существенное место при мысленном наглядном моделировании занимает *макетирование*. Мысленный макет может применяться в случаях, когда протекающие в реальном объекте процессы не поддаются физическому моделированию, либо может предшествовать проведению других видов моделирования. В основе построения мысленных макетов также лежат аналогии, однако обычно базирующиеся на причинно-следственных связях между явлениями и процессами в объекте. Если ввести условное обозначение отдельных понятий, т. е. знаки, а также определенные операции между этими знаками, то можно реализовать *знаковое моделирование* и с помощью знаков отображать набор понятий — составлять отдельные цепочки из слов и предложений. Используя операции объединения, пересечения и дополнения теории множеств, можно в отдельных символах дать описание какого-то реального объекта.

В основе *языкового моделирования* лежит некоторый тезаурус. Последний образуется из набора входящих понятий, причем этот набор должен быть фиксированным. Следует отметить, что между тезаурусом и обычным словарем имеются принципиальные различия. Тезаурус — словарь, который очищен от неоднозначности, т. е. в нем каждому слову может соответствовать лишь единственное понятие, хотя в обычном словаре одному слову могут соответствовать несколько понятий.

Символическое моделирование представляет собой искусственный процесс создания логического объекта, который замещает реальный и выражает основные свойства его отношений с помощью определенной системы знаков или символов.

Математическое моделирование. Для исследования характеристик процесса функционирования любой системы S 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, интегродифференциальных, 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).

Наиболее полное исследование процесса функционирования системы можно провести, если известны явные зависимости, связывающие искомые характеристики с начальными условиями, параметрами и переменными системы S . Однако такие зависимости удается получить только для сравнительно простых систем. При усложнении систем исследование их аналитическим методом наталкивается на значительные трудности, которые часто бывают непреодолимыми. Поэтому, желая использовать аналитический метод, в этом случае идут на существенное упрощение первоначальной модели, чтобы иметь возможность изучить хотя бы общие свойства системы. Такое исследование на упрощенной модели аналитическим методом помогает получить ориентировочные результаты для определения более точных оценок другими методами. 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 отдельных случаях исследования системы могут удовлетворить и те выводы, которые можно сделать при использовании качественного метода анализа математической модели. Такие качественные методы широко используются, например, в теории автоматического управления для оценки эффективности различных вариантов систем управления.

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 последователь-

ности 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. В настоящее время имитационное моделирование — наиболее эффективный метод исследования больших систем, а часто и единственный практически доступный метод получения информации о поведении системы, особенно на этапе ее проектирования [4, 11, 31, 37, 46].

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 и целесообразно в качестве метода машинной реализации имитационной модели использовать метод статистического моделирования. Первоначально был разработан метод статистических испытаний, представляющий собой численный метод, который применялся для моделирования случайных величин и функций, вероятностные характеристики которых совпадали с решениями аналитических задач (такая процедура получила название метода Монте-Карло). Затем этот прием стали применять и для машинной имитации с целью исследования характеристик процессов функционирования систем, подверженных случайным воздействиям, т. е. появился метод статистического моделирования [4, 10, 18, 29, 37]. Таким образом, методом статистического моделирования будем в дальнейшем называть метод машинной реализации имитационной модели, а методом статистических испытаний (Монте-Карло) — численный метод решения аналитической задачи.

Метод имитационного моделирования позволяет решать задачи анализа больших систем S , включая задачи оценки: вариантов структуры системы, эффективности различных алгоритмов управления системой, влияния изменения различных параметров системы. Имитационное моделирование может быть положено также в основу структурного, алгоритмического и параметрического синтеза больших систем, когда требуется создать систему, с заданными характеристиками при определенных ограничениях, которая является оптимальной по некоторым критериям оценки эффективности.

При решении задач машинного синтеза систем на основе их имитационных моделей помимо разработки моделирующих алгоритмов для анализа фиксированной системы необходимо также

разработать алгоритмы поиска оптимального варианта системы. Далее в методологии машинного моделирования будем различать два основных раздела: статику и динамику, — основным содержанием которых являются соответственно вопросы анализа и синтеза систем, заданных моделирующими алгоритмами [29, 37].

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.

Другие виды моделирования. При *реальном моделировании* используется возможность исследования различных характеристик либо на реальном объекте целиком, либо на его части. Такие исследования могут проводиться как на объектах, работающих в нормальных режимах, так и при организации специальных режимов для оценки интересующих исследователя характеристик (при других значениях переменных и параметров, в другом масштабе времени и т. д.). Реальное моделирование является наиболее адекватным, но при этом его возможности с учетом особенностей реальных объектов ограничены. Например, проведение реального моделирования АСУ предприятием потребует, во-первых, создания такой АСУ, а во-вторых, проведения экспериментов с управляемым объектом, т. е. предприятием, что в большинстве случаев невозможно. Рассмотрим разновидности реального моделирования.

Натурным моделированием называют проведение исследования на реальном объекте с последующей обработкой результатов эксперимента на основе теории подобия. При функционировании объекта в соответствии с поставленной целью удастся выявить закономерности протекания реального процесса. Надо отметить, что такие разновидности натурного эксперимента, как *производственный эксперимент* и *комплексные испытания*, обладают высокой степенью достоверности.

С развитием техники и проникновением в глубь процессов, протекающих в реальных системах, возрастает техническая оснащенность современного *научного эксперимента*. Он характеризуется широким использованием средств автоматизации проведения, применением весьма разнообразных средств обработки информации, возможностью вмешательства человека в процесс проведения эксперимента, и в соответствии с этим появилось новое научное направление — автоматизация научных экспериментов [12, 34, 40].

Отличие эксперимента от реального протекания процесса заключается в том, что в нем могут появиться отдельные критические

ситуации и определяться границы устойчивости процесса. В ходе эксперимента вводятся новые факторы и возмущающие воздействия в процессе функционирования объекта. Одна из разновидностей эксперимента — комплексные испытания, которые также можно отнести к натурному моделированию, когда вследствие повторения испытаний изделий выявляются общие закономерности о надежности этих изделий, о характеристиках качества и т. д. В этом случае моделирование осуществляется путем обработки и обобщения сведений, проходящих в группе однородных явлений. Наряду со специально организованными испытаниями возможна реализация натурального моделирования путем обобщения опыта, накопленного в ходе производственного процесса, т. е. можно говорить о производственном эксперименте. Здесь на базе теории подобия обрабатывают статистический материал по производственному процессу и получают его обобщенные характеристики.

Другим видом реального моделирования является *физическое*, отличающееся от натурального тем, что исследование проводится на установках, которые сохраняют природу явлений и обладают физическим подобием. В процессе физического моделирования задаются некоторые характеристики внешней среды и исследуется поведение либо реального объекта, либо его модели при заданных или создаваемых искусственно воздействиях внешней среды. Физическое моделирование может протекать в *реальном и нереальном* (псевдоральном) *масштабах времени*, а также может рассматриваться без учета времени. В последнем случае изучению подлежат так называемые «замороженные» процессы, которые фиксируются в некоторый момент времени. Наибольшие сложность и интерес с точки зрения верности получаемых результатов представляет физическое моделирование в реальном масштабе времени.

С точки зрения математического описания объекта и в зависимости от его характера модели можно разделить на модели аналоговые (непрерывные), цифровые (дискретные) и аналого-цифровые (комбинированные). Под *аналоговой* моделью понимается модель, которая описывается уравнениями, связывающими непрерывные величины. Под *цифровой* понимают модель, которая описывается уравнениями, связывающими дискретные величины, представленные в цифровом виде. Под *аналого-цифровой* понимается модель, которая может быть описана уравнениями, связывающими непрерывные и дискретные величины.

Особое место в моделировании занимает кибернетическое моделирование, в котором отсутствует непосредственное подобие физических процессов, происходящих в моделях, реальным процессам. В этом случае стремятся отобразить лишь некоторую функцию и рассматривают реальный объект как «черный ящик», имеющий ряд входов и выходов, и моделируют некоторые связи между выходами и входами. Чаще всего при использовании кибернетических моделей проводят анализ поведенческой стороны объ-

екта при различных воздействиях внешней среды [11, 25, 27, 44]. Таким образом, в основе кибернетических моделей лежит отражение некоторых информационных процессов управления, что позволяет оценить поведение реального объекта. Для построения имитационной модели в этом случае необходимо выделить исследуемую функцию реального объекта, попытаться формализовать эту функцию в виде некоторых операторов связи между входом и выходом и воспроизвести на имитационной модели данную функцию, причем на базе совершенно иных математических соотношений и, естественно, иной физической реализации процесса.

1.4. ВОЗМОЖНОСТИ И ЭФФЕКТИВНОСТЬ МОДЕЛИРОВАНИЯ СИСТЕМ НА ВЫЧИСЛИТЕЛЬНЫХ МАШИНАХ

Обеспечение требуемых показателей качества функционирования больших систем, связанное с необходимостью изучения протекания стохастических процессов в исследуемых и проектируемых системах S , позволяет проводить комплекс теоретических и экспериментальных исследований, взаимно дополняющих друг друга. Эффективность экспериментальных исследований сложных систем оказывается крайне низкой, поскольку проведение натурных экспериментов с реальной системой либо требует больших материальных затрат и значительного времени, либо вообще практически невозможно (например, на этапе проектирования, когда реальная система отсутствует). Эффективность теоретических исследований с практической точки зрения в полной мере проявляется лишь тогда, когда их результаты с требуемой степенью точности и достоверности могут быть представлены в виде аналитических соотношений или моделирующих алгоритмов, пригодных для получения соответствующих характеристик процесса функционирования исследуемых систем.

Средства моделирования систем. Появление современных ЭВМ было решающим условием широкого внедрения аналитических методов в исследование сложных систем. Стало казаться, что модели и методы, например математического программирования, станут практическим инструментом решения задач управления в больших системах. Действительно, были достигнуты значительные успехи в создании новых математических методов решения этих задач, однако математическое программирование так и не стало практическим инструментом исследования процесса функционирования сложных систем, так как модели математического программирования оказались слишком грубыми и несовершенными для их эффективного использования. Необходимость учета стохастических свойств системы, недетерминированности исходной информации, наличия корреляционных связей между большим числом переменных и параметров, характеризующих процессы в системах, приводят к построению сложных математических моделей, которые не

могут быть применены в инженерной практике при исследовании таких систем аналитическим методом. Пригодные для практических расчетов аналитические соотношения удается получить лишь при упрощающих предположениях, обычно существенно искажающих фактическую картину исследуемого процесса. Поэтому в последнее время все острее чувствуется потребность в разработке методов, которые дали бы возможность уже на этапе проектирования систем исследовать более адекватные модели. Указанные обстоятельства приводят к тому, что при исследовании больших систем все шире применяются методы имитационного моделирования [8, 11, 19, 25, 41, 54].

Наиболее конструктивным средством решения инженерных задач на базе моделирования в настоящее время стали ЭВМ. Современные ЭВМ можно разделить на две группы: универсальные, прежде всего предназначенные для выполнения расчетных работ, и управляющие, позволяющие проводить не только расчетные работы, но прежде всего приспособленные для управления объектами в реальном масштабе времени. Управляющие ЭВМ могут быть использованы как для управления технологическим процессом, экспериментом, так и для реализации различных имитационных моделей. В зависимости от того, удастся ли построить достаточно точную математическую модель реального процесса, или вследствие сложности объекта не удастся проникнуть в глубь функциональных связей реального объекта и описать их какими-то аналитическими соотношениями, можно рассматривать два основных пути использования ЭВМ: как средства расчета по полученным аналитическим моделям и как средства имитационного моделирования.

Для известной аналитической модели, полагая, что она достаточно точно отображает исследуемую сторону функционирования реального физического объекта, перед вычислительной машиной стоит задача расчета характеристик системы по каким-либо математическим соотношениям при подстановке числовых значений. В этом направлении вычислительные машины обладают возможностями, практически зависящими от порядка решаемого уравнения и от требований к скорости решения, причем могут быть использованы как ЭВМ, так и АВМ.

При использовании ЭВМ разрабатывается алгоритм расчета характеристик, в соответствии с которым составляются программы (либо генерируются с помощью пакета прикладных программ), дающие возможность осуществлять расчеты по требуемым аналитическим соотношениям. Основная задача исследователя заключается в том, чтобы попытаться описать поведение реального объекта одной из известных математических моделей.

Использование АВМ, с одной стороны, ускоряет для достаточно простых случаев процесс решения задачи, с другой стороны, могут возникать погрешности, обусловленные наличием дрейфа параметров отдельных блоков, входящих в АВМ, ограниченной точностью,

с которой могут быть заданы параметры, вводимые в машину, а также неисправностями технических средств и т. д.

Перспективно сочетание ЭВМ и АВМ, т. е. использование гибридных средств вычислительной техники — гибридных вычислительных комплексов (ГВК), что в ряде случаев значительно ускоряет процесс исследования [12, 20, 37, 49].

В ГВК удастся сочетать высокую скорость функционирования аналоговых средств и высокую точность расчетов на базе цифровых средств вычислительной техники. Одновременно удастся за счет наличия цифровых устройств обеспечить контроль проведения операций. Опыт использования вычислительной техники в задачах моделирования показывает, что с усложнением объекта большую эффективность по скорости решения и по стоимости выполнения операций дает использование гибридной техники.

Конкретным техническим средством воплощения имитационной модели могут быть ЭВМ, АВМ и ГВК. Если использование аналоговой техники ускоряет получение конечных результатов, сохраняя некоторую наглядность протекания реального процесса, то применение средств цифровой техники позволяет осуществить контроль за реализацией модели, создать программы по обработке и хранению результатов моделирования, обеспечить эффективный диалог исследователя с моделью.

Обычно модель строится по *иерархическому принципу*, когда последовательно анализируются отдельные стороны функционирования объекта и при перемещении центра внимания исследователя рассмотренные ранее подсистемы переходят во внешнюю среду. Иерархическая структура моделей может раскрывать и ту последовательность, в которой изучается реальный объект, а именно последовательность перехода от структурного (топологического) уровня к функциональному (алгоритмическому) и от функционального к параметрическому.

Результат моделирования в значительной степени зависит от адекватности исходной концептуальной (описательной) модели, от полученной степени подобия описания реального объекта, числа реализаций модели и многих других факторов. В ряде случаев сложность объекта не позволяет не только построить математическую модель объекта, но и дать достаточно близкое кибернетическое описание, и перспективным здесь является выделение наиболее трудно поддающейся математическому описанию части объекта и включение этой реальной части физического объекта в имитационную модель. Тогда модель реализуется, с одной стороны, на базе средств вычислительной техники, а с другой — имеется реальная часть объекта. Это значительно расширяет возможности и повышает достоверность результатов 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 удается 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.

Обеспечение моделирования. Эксперимент с имитационной моделью требует серьезной подготовки, поэтому имитационная система характеризуется наличием математического, программного, информационного, технического, эргономического и других видов обеспечения.

Математическое обеспечение имитационной системы включает в себя совокупность математических соотношений, описывающих поведение реального объекта, совокупность алгоритмов, обеспечивающих как подготовку, так и работу с моделью. Сюда могут быть отнесены алгоритмы ввода исходных данных, имитации, вывода, обработки.

Программное обеспечение по своему содержанию включает в себя совокупность программ: планирования эксперимента, имитационной модели, проведения эксперимента, обработки и интерпретации результатов. Кроме того, программное обеспечение имитационной системы должно обеспечивать синхронизацию процессов в модели, т. е. необходим блок, организующий псевдопараллельное выполнение процессов в модели. Машинные эксперименты с имитационными моделями не могут проходить без хорошо разработанного и реализованного информационного обеспечения.

Информационное обеспечение включает в себя средства и технологию организации и реорганизации базы данных моделирования, методы логической и физической организации массивов, формы документов, описывающих процесс моделирования и его результаты. Информационное обеспечение имитационной системы является наименее разработанной частью, поскольку только в настоящее время наблюдается переход к созданию сложных имитационных моделей и разрабатывается методология их использования

при анализе и синтезе сложных систем с использованием концепции базы данных и знаний.

Техническое обеспечение имитационной системы включает в себя прежде всего средства вычислительной техники, связи и обмена между оператором и сетью ЭВМ, ввода и вывода информации, управления проведением эксперимента. К техническому обеспечению предъявляются весьма серьезные требования по надежности функционирования, так как сбои и отказы технических средств, ошибки оператора ЭВМ могут резко увеличить время работы с имитационной моделью и даже привести к неверным конечным результатам.

Эргономическое обеспечение имитационной системы представляет собой совокупность научных и прикладных методик и методов, а также нормативно-технических и организационно-методических документов, используемых на всех этапах взаимодействия человека-экспериментатора с инструментальными средствами (ЭВМ, гибридными комплексами и т. д.). Эти документы, используемые на всех стадиях разработки и эксплуатации имитационных систем и их элементов, предназначены для формирования и поддержания эргономического качества путем обоснования и выбора организационно-проектных решений, которые создадут оптимальные условия для высокоэффективной деятельности человека во взаимодействии с моделирующим комплексом.

Таким образом, 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.

Возможности машинного моделирования. Несмотря на то что имитационное моделирование на ЭВМ является мощным инструментом исследования систем, его применение рационально не во всех случаях. Известно множество задач, решаемых более эффективно другими методами. Вместе с тем для большого класса задач исследования и проектирования систем метод имитационного моделирования наиболее приемлем. Правильное его употребление возможно лишь в случае четкого понимания сущности метода имитационного моделирования и условий его использования в практике исследования реальных систем при учете особенностей конкретных систем и возможностей их исследования различными методами.

В качестве основных критериев целесообразности применения метода имитационного моделирования на ЭВМ можно указать следующие: отсутствие или неприемлемость аналитических, численных и качественных методов решения поставленной задачи; наличие достаточного количества исходной информации о моделируемой системе S для обеспечения возможности построения адекватной

имитационной модели; необходимость проведения на базе других возможных методов решения очень большого количества вычислений, трудно реализуемых даже с использованием ЭВМ; возможность поиска оптимального варианта системы при ее моделировании на ЭВМ.

Имитационное моделирование на ЭВМ, как и любой метод исследований, имеет достоинства и недостатки, проявляющиеся в конкретных приложениях [37, 43, 46]. К числу основных достоинств метода имитационного моделирования при исследовании сложных систем можно отнести следующие: машинный эксперимент с имитационной моделью дает возможность исследовать особенности процесса функционирования системы S в любых условиях; применение ЭВМ в имитационном эксперименте существенно сокращает продолжительность испытаний по сравнению с натурным экспериментом; имитационная модель позволяет включать результаты натуральных испытаний реальной системы или ее частей для проведения дальнейших исследований; имитационная модель обладает известной гибкостью варьирования структуры, алгоритмов и параметров моделируемой системы, что важно с точки зрения поиска оптимального варианта системы; имитационное моделирование сложных систем часто является единственным практически реализуемым методом исследования процесса функционирования таких систем на этапе их проектирования.

Основным недостатком, проявляющимся при машинной реализации метода имитационного моделирования, является то, что решение, полученное при анализе имитационной модели M , всегда носит частный характер, так как оно соответствует фиксированным элементам структуры, алгоритмам поведения и значениям параметров системы S , начальных условий и воздействий внешней среды E . Поэтому для полного анализа характеристик процесса функционирования систем, а не получения только отдельной точки приходится многократно воспроизводить имитационный эксперимент, варьируя исходные данные задачи. При этом, как следствие, возникает увеличение затрат машинного времени на проведение эксперимента с имитационной моделью процесса функционирования исследуемой системы S .

Эффективность машинного моделирования. При имитационном моделировании, так же как и при любом другом методе анализа и синтеза системы S , весьма существенен вопрос его эффективности. Эффективность имитационного моделирования может оцениваться рядом критериев, в том числе точностью и достоверностью результатов моделирования, временем построения и работы с моделью M , затратами машинных ресурсов (времени и памяти), стоимостью разработки и эксплуатации модели. Очевидно, наилучшей оценкой эффективности является сравнение получаемых результатов с реальным исследованием, т. е. с моделированием на реальном объекте при проведении натурального эксперимента. Поскольку это не всегда

удается сделать, статистический подход позволяет с определенной степенью точности при повторяемости машинного эксперимента получить какие-то усредненные характеристики поведения системы. Существенное влияние на точность моделирования оказывает число реализаций, и в зависимости от требуемой достоверности можно оценить необходимое число реализаций воспроизводимого случайного процесса.

Существенным показателем эффективности являются затраты машинного времени. В связи с использованием ЭВМ различного типа суммарные затраты складываются из времени по вводу и выводу данных по каждому алгоритму моделирования, времени на проведение вычислительных операций, с учетом обращения к оперативной памяти и внешним устройствам, а также сложности каждого моделирующего алгоритма. Расчеты затрат машинного времени являются приближенными и могут уточняться по мере отладки программ и накопления опыта у исследователя при работе с имитационной моделью. Большое влияние на затраты машинного времени при проведении имитационных экспериментов оказывает рациональное планирование таких экспериментов. Определенное влияние на затраты машинного времени могут оказать процедуры обработки результатов моделирования, а также форма их представления.

Построение имитационных моделей больших систем и проведение машинных экспериментов с этими моделями представляют собой достаточно трудоемкий процесс, в котором в настоящее время много неизученного. Однако специалисты в области проектирования, исследования и эксплуатации больших систем должны в совершенстве знать методологию машинного моделирования, сложившуюся к настоящему времени, чтобы быть готовыми к появлению ЭВМ следующих поколений, которые позволят сделать еще один существенный шаг в автоматизации построения моделей и использования имитационного моделирования систем.

Контрольные вопросы

- 1.1. В чем сущность системного подхода к моделированию систем на ЭВМ?
- 1.2. Что такое процесс функционирования системы?
- 1.3. В каком соотношении находятся понятия «эксперимент» и «машинное моделирование»?
- 1.4. Каковы основные характерные черты машинной модели?
- 1.5. В чем заключается цель моделирования системы на ЭВМ?
- 1.6. Какие существуют классификационные признаки видов моделирования систем?
- 1.7. Что собой представляет математическое моделирование систем?
- 1.8. Какие особенности характеризуют имитационное моделирование систем?
- 1.9. В чем суть метода статистического моделирования на ЭВМ?
- 1.10. Чем определяется эффективность моделирования систем на ЭВМ?

ГЛАВА 2

МАТЕМАТИЧЕСКИЕ СХЕМЫ МОДЕЛИРОВАНИЯ СИСТЕМ

Наибольшие затруднения и наиболее серьезные ошибки при моделировании возникают при переходе от содержательного к формальному описанию объектов исследования, что объясняется участием в этом творческом процессе коллективов разных специальностей: специалистов в области систем, которые требуется моделировать (заказчиков), и специалистов в области машинного моделирования (исполнителей). Эффективным средством для нахождения взаимопонимания между этими группами специалистов является язык математических схем, позволяющий во главу угла поставить вопрос об адекватности перехода от содержательного описания системы к ее математической схеме, а лишь затем решать вопрос о конкретном методе получения результатов с использованием ЭВМ: аналитическом или имитационном, а возможно, и комбинированном, т. е. аналитико-имитационном. Применительно к конкретному объекту моделирования, т. е. к сложной системе, разработчику модели должны помочь конкретные, уже прошедшие апробацию для данного класса систем математические схемы, показавшие свою эффективность в прикладных исследованиях на ЭВМ и получившие название типовых математических схем.

2.1. ОСНОВНЫЕ ПОДХОДЫ К ПОСТРОЕНИЮ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ СИСТЕМ

Исходной информацией при построении математических моделей процессов функционирования систем служат данные о назначении и условиях работы исследуемой (проектируемой) системы S . Эта информация определяет основную цель моделирования системы S и позволяет сформулировать требования к разрабатываемой математической модели M . Причем уровень абстрагирования зависит от круга тех вопросов, на которые исследователь системы хочет получить ответ с помощью модели, и в какой-то степени определяет выбор математической схемы [4, 13, 29, 37, 42, 48].

Математические схемы. Введение понятия «математическая схема» позволяет рассматривать математику не как метод расчета, а как метод мышления, как средство формулирования понятий, что является наиболее важным при переходе от словесного описания системы к формальному представлению процесса ее функционирования в виде некоторой математической модели (аналитической или имитационной). При пользовании математической схемой исследователя системы S в первую очередь должен интересовать вопрос об адекватности отображения в виде конкретных схем реальных про-

цессов в исследуемой системе, а не возможность получения ответа (результата решения) на конкретный вопрос исследования. Например, представление процесса функционирования информационно-вычислительной системы коллективного пользования в виде сети схем массового обслуживания дает возможность хорошо описать процессы, происходящие в системе, но при сложных законах распределения входящих потоков и потоков обслуживания не дает возможности получения результатов в явном виде [13, 21, 30, 33, 37, 41].

Математическую схему можно определить как звено при переходе от содержательного к формальному описанию процесса функционирования системы с учетом воздействия внешней среды, т. е. имеет место цепочка «описательная модель — математическая схема — математическая [аналитическая или (и) имитационная] модель».

Каждая конкретная система S характеризуется набором свойств, под которыми понимаются величины, отражающие поведение моделируемого объекта (реальной системы) и учитывающие условия ее функционирования во взаимодействии с внешней средой (системой) E . При построении математической модели системы необходимо решить вопрос об ее полноте. Полнота модели регулируется в основном выбором границы «система S — среда E ». Также должна быть решена задача упрощения модели, которая помогает выделить основные свойства системы, отбросив второстепенные. Причем отнесение свойств системы к основным или второстепенным существенно зависит от цели моделирования системы (например, анализ вероятностно-временных характеристик процесса функционирования системы, синтез структуры системы и т. д.).

Формальная модель объекта. Модель объекта моделирования, т. е. системы S , можно представить в виде множества величин, описывающих процесс функционирования реальной системы и образующих в общем случае следующие подмножества: совокупность *входных воздействий* на систему

$$x_i \in X, i = \overline{1, n_X};$$

совокупность *воздействий внешней среды*

$$v_l \in V, l = \overline{1, n_V};$$

совокупность *внутренних (собственных) параметров* системы

$$h_k \in H, k = \overline{1, n_H};$$

совокупность *выходных характеристик* системы

$$y_j \in Y, j = \overline{1, n_Y}.$$

При этом в перечисленных подмножествах можно выделить управляемые и неуправляемые переменные. В общем случае $x_i, v_l, h_k,$

y_j являются элементами непересекающихся подмножеств и содержат как детерминированные, так и стохастические составляющие.

При моделировании системы S входные воздействия, воздействия внешней среды E и внутренние параметры системы являются *независимыми (экзогенными) переменными*, которые в векторной форме имеют соответственно вид $\vec{x}(t) = (x_1(t), x_2(t), \dots, x_{n_x}(t))$; $\vec{v}(t) = (v_1(t), v_2(t), \dots, v_{n_v}(t))$; $\vec{h}(t) = (h_1(t), h_2(t), \dots, h_{n_h}(t))$, а выходные характеристики системы являются *зависимыми (эндогенными) переменными* и в векторной форме имеют вид $\vec{y}(t) = (y_1(t), y_2(t), \dots, y_{n_y}(t))$.

Процесс функционирования системы S описывается во времени оператором F_S , который в общем случае преобразует экзогенные переменные в эндогенные в соответствии с соотношениями вида

$$\vec{y}(t) = F_S(\vec{x}, \vec{v}, \vec{h}, t). \quad (2.1)$$

Совокупность зависимостей выходных характеристик системы от времени $y_j(t)$ для всех видов $j = \overline{1, n_y}$ называется *выходной траекторией* $\vec{y}(t)$. Зависимость (2.1) называется *законом функционирования системы S* и обозначается F_S . В общем случае закон функционирования системы F_S может быть задан в виде функции, функционала, логических условий, в алгоритмической и табличной формах или в виде словесного правила соответствия.

Весьма важным для описания и исследования системы S является понятие *алгоритма функционирования A_S* , под которым понимается метод получения выходных характеристик с учетом входных воздействий $\vec{x}(t)$, воздействий внешней среды $\vec{v}(t)$ и собственных параметров системы $\vec{h}(t)$. Очевидно, что один и тот же закон функционирования F_S системы S может быть реализован различными способами, т. е. с помощью множества различных алгоритмов функционирования A_S .

Соотношения (2.1) являются математическим описанием поведения объекта (системы) моделирования во времени t , т. е. отражают его динамические свойства. Поэтому математические модели такого вида принято называть *динамическими моделями (системами)* [4, 11, 43, 44].

Для статических моделей математическая модель (2.1) представляет собой отображение между двумя подмножествами свойств моделируемого объекта Y и $\{X, V, H\}$, что в векторной форме может быть записано как

$$\vec{y} = f(\vec{x}, \vec{v}, \vec{h}). \quad (2.2)$$

Соотношения (2.1) и (2.2) могут быть заданы различными способами: аналитически (с помощью формул), графически, таблично и т. д. Такие соотношения в ряде случаев могут быть получены

через свойства системы S в конкретные моменты времени, называемые *состояниями*. Состояние системы S характеризуется векторами

$$\vec{z}' = (z'_1, z'_2, \dots, z'_k) \text{ и } z'' = (\vec{z}''_1, \vec{z}''_2, \dots, z''_k),$$

где $z'_1 = z_1(t')$, $z'_2 = z_2(t')$, ..., $z'_k = z_k(t')$ в момент $t' \in (t_0, T)$; $z''_1 = z_1(t'')$, $z''_2 = z_2(t'')$, ..., $z''_k = z_k(t'')$ в момент $t'' \in (t_0, T)$ и т. д., $k = \overline{1, n_Z}$.

Если рассматривать процесс функционирования системы S как последовательную смену состояний $z_1(t)$, $z_2(t)$, ..., $z_k(t)$, то они могут быть интерпретированы как координаты точки в k -мерном фазовом пространстве, причем каждой реализации процесса будет соответствовать некоторая фазовая траектория. Совокупность всех возможных значений состояний $\{\vec{z}\}$ называется *пространством состояний* объекта моделирования Z , причем $z_k \in Z$.

Состояния системы S в момент времени $t_0 < t^* \leq T$ полностью определяются начальными условиями $\vec{z}^0 = (z^0_1, z^0_2, \dots, z^0_k)$ [где $z^0_1 = z_1(t_0)$, $z^0_2 = z_2(t_0)$, ..., $z^0_k = z_k(t_0)$], входными воздействиями $\vec{x}(t)$, внутренними параметрами $\vec{h}(t)$ и воздействиями внешней среды $\vec{v}(t)$, которые имели место за промежуток времени $t^* - t_0$, с помощью двух векторных уравнений

$$z(t) = \Phi(\vec{z}^0, \vec{x}, \vec{v}, \vec{h}, t); \quad (2.3)$$

$$\vec{y}(t) = F(\vec{z}, t). \quad (2.4)$$

Первое уравнение по начальному состоянию \vec{z}^0 и экзогенным переменным \vec{x} , \vec{v} , \vec{h} определяет вектор-функцию $\vec{z}(t)$, а второе по полученному значению состояний $\vec{z}(t)$ — эндогенные переменные на выходе системы $\vec{y}(t)$. Таким образом, цепочка уравнений объекта «вход — состояния — выход» позволяет определить характеристики системы

$$\vec{y}(t) = F[\Phi(\vec{z}^0, \vec{x}, \vec{v}, \vec{h}, t)]. \quad (2.5)$$

В общем случае время в модели системы S может рассматриваться на интервале моделирования $(0, T)$ как непрерывное, так и дискретное, т. е. квантованное на отрезки длиной Δt временных единиц каждый, когда $T = m\Delta t$, где $m = \overline{1, m_T}$ — число интервалов дискретизации.

Таким образом, под *математической моделью объекта* (реальной системы) понимают конечное подмножество переменных $\{\vec{x}(t), \vec{v}(t), \vec{h}(t)\}$ вместе с математическими связями между ними и характеристиками $\vec{y}(t)$ [4, 9, 10, 35].

Если математическое описание объекта моделирования не содержит элементов случайности или они не учитываются, т. е. если

можно считать, что в этом случае стохастические воздействия внешней среды $\vec{v}(t)$ и стохастические внутренние параметры $\vec{h}(t)$ отсутствуют, то модель называется *детерминированной* в том смысле, что характеристики однозначно определяются детерминированными входными воздействиями

$$\vec{y}(t) = f(\vec{x}, t). \quad (2.6)$$

Очевидно, что детерминированная модель является частным случаем стохастической модели.

Типовые схемы. Приведенные математические соотношения представляют собой математические схемы общего вида и позволяют описать широкий класс систем. Однако в практике моделирования объектов в области системотехники и системного анализа на первоначальных этапах исследования системы рациональнее использовать *типовые математические схемы*: дифференциальные уравнения, конечные и вероятностные автоматы, системы массового обслуживания, сети Петри и т. д.

Не обладая такой степенью общности, как рассмотренные модели, типовые математические схемы имеют преимущества простоты и наглядности, но при существенном сужении возможностей применения. В качестве детерминированных моделей, когда при исследовании случайные факторы не учитываются, для представления систем, функционирующих в непрерывном времени, используются дифференциальные, интегральные, интегродифференциальные и другие уравнения, а для представления систем, функционирующих в дискретном времени, — конечные автоматы и конечно-разностные схемы. В качестве стохастических моделей (при учете случайных факторов) для представления систем с дискретным временем используются вероятностные автоматы, а для представления системы с непрерывным временем — системы массового обслуживания и т. д.

Перечисленные типовые математические схемы, естественно, не могут претендовать на возможность описания на их базе всех процессов, происходящих в больших информационно-управляющих системах. Для таких систем в ряде случаев более перспективным является применение агрегативных моделей [4, 37]. Агрегативные модели (системы) позволяют описать широкий круг объектов исследования с отображением системного характера этих объектов. Именно при агрегативном описании сложный объект (система) расчленяется на конечное число частей (подсистем), сохраняя при этом связи, обеспечивающие взаимодействие частей.

Таким образом, при построении математических моделей процессов функционирования систем можно выделить следующие основные подходы: непрерывно-детерминированный (например, дифференциальные уравнения); дискретно-детерминированный (ко-

нечные автоматы); дискретно-стохастический (вероятностные автоматы); непрерывно-стохастический (системы массового обслуживания); обобщенный, или универсальный (агрегативные системы).

Математические схемы, рассматриваемые в последующих параграфах данной главы, должны помочь оперировать различными подходами в практической работе при моделировании конкретных систем.

2.2. НЕПРЕРЫВНО-ДЕТЕРМИНИРОВАННЫЕ МОДЕЛИ (D-СХЕМЫ)

Рассмотрим особенности непрерывно-детерминированного подхода на примере использования в качестве математических моделей дифференциальных уравнений. *Дифференциальными уравнениями* называются такие уравнения, в которых неизвестными будут функции одной или нескольких переменных, причем в уравнение входят не только функции, но и их производные различных порядков. Если неизвестные — функции многих переменных, то уравнения называются *уравнениями в частных производных*, в противном случае при рассмотрении функции только одной независимой переменной уравнения называются *обыкновенными дифференциальными уравнениями*.

Основные соотношения. Обычно в таких математических моделях в качестве независимой переменной, от которой зависят неизвестные искомые функции, служит время t . Тогда математическое соотношение для детерминированных систем (2.6) в общем виде будет

$$\vec{y}' = \vec{f}(\vec{y}, t); \vec{y}(t_0) = \vec{y}_0, \quad (2.7)$$

где $\vec{y}' = d\vec{y}/dt$, $\vec{y} = (y_1, y_2, \dots, y_n)$ и $\vec{f} = (f_1, f_2, \dots, f_n)$ — n -мерные векторы; $f(\vec{y}, t)$ — вектор-функция, которая определена на некотором $(n+1)$ -мерном (\vec{y}, t) множестве и является непрерывной.

Так как математические схемы такого вида отражают динамику изучаемой системы, т. е. ее поведение во времени, то они называются *D-схемами* (англ. dynamic) [4, 37].

В простейшем случае обыкновенное дифференциальное уравнение имеет вид

$$y' = f(y, t). \quad (2.8)$$

Наиболее важно для системотехники приложение *D-схем* в качестве математического аппарата в теории автоматического управления. Для иллюстрации особенностей построения и применения *D-схем* рассмотрим простейший пример формализации процесса функционирования двух элементарных систем различной физической

природы: механической S_m (колебания маятника, рис. 2.1, а) и электрической S_x (колебательный контур, рис. 2.1, б).

Процесс малых колебаний маятника описывается обыкновенным дифференциальным уравнением

$$m_m l_m^2 [d^2 \theta(t)/dt^2] + m_m g l_m \theta(t) = 0,$$

где m_m , l_m — масса и длина подвеса маятника; g — ускорение свободного падения; $\theta(t)$ — угол отклонения маятника в момент времени t .

Из этого уравнения свободного колебания маятника можно найти оценки интересующих характеристик. Например, период колебания маятника

$$T_m = 2\pi \sqrt{l_m/g}.$$

Аналогично, процессы в электрическом колебательном контуре описываются обыкновенным дифференциальным уравнением

$$L_x [d^2 q(t)/dt^2] + [q(t)/C_x] = 0,$$

где L_x , C_x — индуктивность и емкость конденсатора; $q(t)$ — заряд конденсатора в момент времени t .

Из этого уравнения можно получить различные оценки характеристик процесса в колебательном контуре. Например, период характеристических колебаний

$$T_x = 2\pi \sqrt{L_x C_x}.$$

Очевидно, что, введя обозначения $h_0 = m_m l_m^2 = L_x$, $h_1 = 0$, $h_2 = m_m g l_m = 1/C_x$, $\theta(t) = q(t) = z(t)$, получим обыкновенное дифференциальное уравнение второго порядка, описывающее поведение этой замкнутой системы:

$$h_0 [d^2 z(t) dt^2] + h_1 [dz(t)/dt] + z_2 z(t) = 0, \quad (2.9)$$

где h_0 , h_1 , h_2 — параметры системы; $z(t)$ — состояние системы в момент времени t .

Таким образом, поведение этих двух объектов может быть исследовано на основе общей математической модели (2.9). Кроме того, необходимо отметить, что поведение одной из систем может быть проанализировано с помощью другой. Например, поведение маятника (системы S_m) может быть изучено с помощью электрического колебательного контура (системы S_x).

Если изучаемая система S , т. е. маятник или контур, взаимодействует с внешней средой E , то появляется входное воздействие $x(t)$ (внешняя сила для маятника и источник энергии для контура)

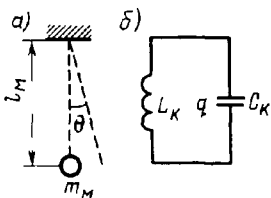


Рис. 2.1. Элементарные системы

и непрерывно-детерминированная модель такой системы будет иметь вид

$$h_0 [d^2 z(t)/dt^2] + h_1 [dz(t)dt] + h_2 z(t) = x(t).$$

С точки зрения общей схемы математической модели (см. § 2.1) $x(t)$ является входным (управляющим) воздействием, а состояние системы S в данном случае можно рассматривать как выходную характеристику, т. е. полагать, что выходная переменная совпадает с состоянием системы в данный момент времени $y = z$.

Возможные приложения. При решении задач системотехники важное значение имеют проблемы управления большими системами. Следует обратить внимание на системы автоматического управления — частный случай динамических систем, описываемых *D-схемами* и выделенных в отдельный класс моделей в силу их практической специфики [24, 43].

Описывая процессы автоматического управления, придерживаются обычно представления реального объекта в виде двух систем: управляющей и управляемой (объекта управления). Структура многомерной системы автоматического управления общего вида представлена на рис. 2.2, где обозначены эндогенные переменные: $\vec{x}(t)$ — вектор входных (задающих) воздействий; $\vec{v}(t)$ — вектор возмущающих воздействий; $\vec{h}'(t)$ — вектор сигналов ошибки; $\vec{h}''(t)$ — вектор управляющих воздействий; экзогенные переменные: $\vec{z}(t)$ — вектор состояний системы S ; $\vec{y}(t)$ — вектор выходных переменных, обычно $\vec{y}(t) = \vec{z}(t)$.

Современная управляющая система — это совокупность программно-технических средств, обеспечивающих достижение объектом управления определенной цели. Насколько точно объект управления достигает заданной цели, можно судить для одномерной системы по координате состояния $y(t)$. Разность между заданным $y_{зад}(t)$ и действительным $y(t)$ законами изменения управляемой величины есть ошибка управления $h'(t) = y_{зад}(t) - y(t)$. Если пред-

писанный закон изменения управляемой величины соответствует закону изменения входного (задающего) воздействия, т. е. $x(t) = y_{зад}(t)$, то $h'(t) = x(t) - y(t)$.

Системы, для которых ошибки управления $h'(t) = 0$ во все моменты времени, называются *идеальными*. На практике реализация идеальных си-

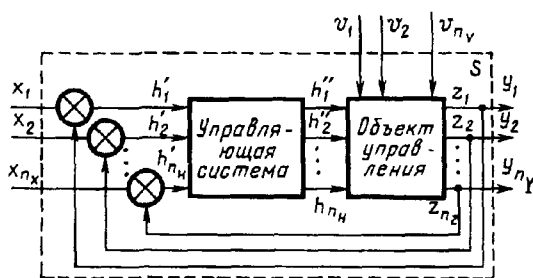


Рис 2.2. Структура системы автоматического управления

стем невозможна. Таким образом, ошибка $h'(t)$ — необходимый субстрат автоматического управления, основанного на принципе отрицательной обратной связи, так как для приведения в соответствие выходной переменной $y(t)$ ее заданному значению используется информация об отклонении между ними. Задачей системы автоматического управления является изменение переменной $y(t)$ согласно заданному закону с определенной точностью (с допустимой ошибкой). При проектировании и эксплуатации систем автоматического управления необходимо выбрать такие параметры системы S , которые обеспечили бы требуемую точность управления, а также устойчивость системы в переходном процессе.

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

Пример 2.1. Рассмотрим одноканальную систему автоматического управления S_A , которая описывается D -схемой общего вида

$$F(y^{n-1}, y^{n-2}, \dots, y, x^m, x^{m-1}, \dots, x) = 0, \quad (2.10)$$

где x^m и y^{n-1} — производные по времени m -го и n -го порядков от функций x и y соответственно. Пусть система S_A , описываемая уравнением (2.10), работает в некотором режиме, характеризуемом функциями $x_0(t)$ и $y_0(t)$. Обозначим малые отклонения $x(t)$ от $x_0(t)$ через $\Delta x(t)$, а $y(t)$ от $y_0(t)$ через $\Delta y(t)$, т. е. $x(t) = x_0(t) + \Delta x(t)$, $y(t) = y_0(t) + \Delta y(t)$.

Тогда уравнение (2.10) можно линеаризовать, разложив функцию $F(y^{n-1}, y^{n-2}, \dots, y, x^m, x^{m-1}, \dots, x)$ в ряд Тейлора и ограничившись его линейными членами относительно приращений Δx и Δy , т. е.

$$\begin{aligned} & \frac{\partial F}{\partial y_0^n} \Delta y^n + \frac{\partial F}{\partial y_0^{n-1}} \Delta y^{n-1} + \dots + \frac{\partial F}{\partial y_0} \Delta y + \Delta y = \\ & = \frac{\partial F}{\partial x_0^m} \Delta x^m + \frac{\partial F}{\partial x_0^{m-1}} \Delta x^{m-1} + \dots + \frac{\partial F}{\partial x_0} \Delta x + \Delta x. \end{aligned} \quad (2.11)$$

Так как полученное уравнение (2.11) приближенно описывает рассматриваемый процесс, то производные вычисляются при некоторых фиксированных значениях входящих в него переменных, т. е. получается система с постоянными коэффициентами. Кроме того, уравнения получаются линейными относительно Δx , Δy и их производных. Это весьма существенно, так как методы решения и исследования линейных систем значительно проще, чем систем общего вида, и более детально разработаны.

Таким образом, для линейных систем автоматического управления, т. е. для систем, описываемых линейными дифференциальными уравнениями, можно записать

$$a_0 \frac{d^n y}{dt^n} + a_1 \frac{d^{n-1} y}{dt^{n-1}} + \dots + a_n y = b_0 \frac{d^m x}{dt^m} + b_1 \frac{d^{m-1} x}{dt^{m-1}} + \dots + b_m x. \quad (2.12)$$

В уравнении (2.12) для простоты предполагается, что точки приложения возмущающих воздействий совпадают с входом системы. Для решения (2.12) можно воспользоваться, например, операторным методом, заменяя дифференциальное уравнение алгебраическим.

Таким образом, использование *D-схем* позволяет формализовать процесс функционирования непрерывно-детерминированных систем S и оценить их основные характеристики, применяя аналитический или имитационный подход, реализованный в виде соответствующего языка для моделирования непрерывных систем или использующий аналоговые и гибридные средства вычислительной техники.

2.3. 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. Теория автоматов — это раздел теоретической кибернетики, в котором изучаются математические модели — автоматы. На основе этой теории система представляется в виде автомата, перерабатывающего дискретную информацию и меняющего свои внутренние состояния лишь в допустимые моменты времени. Понятие «автомат» варьируется в зависимости от характера конкретно изучаемых систем, от принятого уровня абстракции и целесообразной степени общности.

Основные соотношения. Автомат можно представить как некоторое устройство (черный ящик), на которое подаются входные сигналы и снимаются выходные и которое может иметь некоторые внутренние состояния. *Конечным автоматом* называется автомат, у которого множество внутренних состояний и входных сигналов (а следовательно, и множество выходных сигналов) являются конечными множествами.

Абстрактно конечный автомат (англ. *finite automata*) можно представить как математическую схему (*F-схему*), характеризующуюся шестью элементами: конечным множеством X входных сигналов (входным алфавитом); конечным множеством Y выходных сигналов (выходным алфавитом); конечным множеством Z внутренних состояний (внутренним алфавитом или алфавитом состояний); начальным состоянием z_0 , $z_0 \in Z$; функцией переходов $\varphi(z, x)$; функцией выходов $\psi(z, x)$. Автомат, задаваемый *F-схемой*: $F = \langle Z, X, Y, \varphi, \psi, z_0 \rangle$, — 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 t -му такту при $t=0, 1, 2, \dots$, 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 $t=0, 1, 2, \dots$ 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 t -м такте на вход автомата, находящегося в состоянии $z(t)$, подается некоторый сигнал $x(t)$, на который он реагирует переходом в $(t+1)$ -м такте в новое состояние $z(t+1)$ и выдачей некоторого выходного сигнала. Сказанное выше можно описать следующими уравнениями: для F -автомата первого рода, называемого также *автоматом Мили*,

$$z(t+1)=\varphi[z(t), x(t)], t=0, 1, 2, \dots; \quad (2.13)$$

$$y(t)=\psi[z(t), x(t)], t=0, 1, 2, \dots; \quad (2.14)$$

для F -автомата второго рода

$$z(t+1)=\varphi[z(t), x(t)], t=0, 1, 2, \dots; \quad (2.15)$$

$$y(t)=\psi[z(t), x(t-1)], t=1, 2, 3, \dots \quad (2.16)$$

Автомат второго рода, для которого

$$y(t)=\psi[z(t)], t=0, 1, 2, \dots, \quad (2.17)$$

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

Таким образом, уравнения (2.13) — (2.17), полностью задающие F -автомат, являются частным случаем уравнений (2.3) и (2.4), когда система S детерминированная и на ее единственный вход поступает дискретный сигнал X .

По числу состояний различают конечные автоматы с памятью и без памяти. Автоматы с памятью имеют более одного состояния, а автоматы без памяти (комбинационные или логические схемы) обладают лишь одним состоянием. При этом, согласно (2.14), работа комбинационной схемы заключается в том, что она ставит в соответствие каждому входному сигналу $x(t)$ определенный выходной сигнал $y(t)$, т. е. реализует логическую функцию вида

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

Эта функция называется булевой, если алфавиты X и Y , которым принадлежат значения сигналов x и y , состоят из двух букв.

По характеру отсчета дискретного времени конечные автоматы делятся на синхронные и асинхронные. В синхронных *F-автоматах* моменты времени, в которые автомат «считывает» входные сигналы, определяются принудительно синхронизирующими сигналами. После очередного синхронизирующего сигнала с учетом «считанного» и в соответствии с уравнениями (2.13) — (2.17) происходит переход в новое состояние и выдача сигнала на выходе, после чего автомат может воспринимать следующее значение входного сигнала. Таким образом, реакция автомата на каждое значение входного сигнала заканчивается за один такт, длительность которого определяется интервалом между соседними синхронизирующими сигналами. Асинхронный *F-автомат* считывает входной сигнал непрерывно, и поэтому, реагируя на достаточно длинный входной сигнал постоянной величины x , он может, как следует из (2.13) — (2.17), несколько раз изменять состояние, выдавая соответствующее число выходных сигналов, пока не перейдет в устойчивое, которое уже не может быть изменено данным входным сигналом.

Возможные приложения. Чтобы задать конечный *F-автомат*, необходимо описать все элементы множества $F = \langle Z, X, Y, \varphi, \psi, z_0 \rangle$, т. е. входной, внутренней и выходной алфавиты, а также функции переходов и выходов, причем среди множества состояний необходимо выделить состояние z_0 , в котором автомат находился в момент времени $t = 0$. Существует несколько способов задания работы *F-автоматов*, но наиболее часто используются табличный, графический и матричный.

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

состоянием z_k автомата, обозначающим столбец таблицы, стоит соответствующий этому состоянию, согласно (2.17), выходной сигнал $\psi(z_i)$.

Описание работы *F-автомата* Мили таблицами переходов φ и выходов ψ иллюстрируется табл. 2.1, а описание *F-автомата* Мура — таблицей переходов (табл. 2.2).

Таблица 2.1

x_i	z_k			
	z_0	z_1	...	z_k
Переходы				
x_1	$\varphi(z_0, x_1)$	$\varphi(z_1, x_1)$...	$\varphi(z_k, x_1)$
x_2	$\varphi(z_0, x_2)$	$\varphi(z_1, x_2)$
...	$\varphi(z_k, x_2)$
x_l	$\varphi(z_0, x_l)$	$\varphi(z_1, x_l)$...	$\varphi(z_k, x_l)$
Выходы				
x_1	$\psi(z_0, x_1)$	$\psi(z_1, x_1)$...	$\psi(z_k, x_1)$
x_2	$\psi(z_0, x_2)$	$\psi(z_1, x_2)$...	$\psi(z_k, x_2)$
...
x_l	$\psi(z_0, x_l)$	$\psi(z_1, x_l)$...	$\psi(z_k, x_l)$

Таблица 2.2

x_i	$\psi(z_k)$			
	$\psi(z_0)$	$\psi(z_1)$...	$\psi(z_k)$
	z_0	z_1	...	z_k
x_1	$\varphi(z_0, x_1)$	$\varphi(z_1, x_1)$...	$\varphi(z_k, x_1)$
x_2	$\varphi(z_0, x_2)$	$\varphi(z_1, x_2)$...	$\varphi(z_k, x_2)$
...
x_l	$\varphi(z_0, x_l)$	$\varphi(z_1, x_l)$...	$\varphi(z_k, x_l)$

Примеры табличного способа задания *F-автомата* Мили *F1* с тремя состояниями, двумя входными и двумя выходными сигналами приведены в табл. 2.3, а для *F-автомата* Мура *F2* — в табл. 2.4.

При другом способе задания конечного автомата используется понятие направленного графа. Граф автомата представляет собой набор вершин, соответствующих различным состояниям автомата и соединяющих вершины дуг графа, соответствующих тем или иным переходам автомата. Если входной сигнал x_k вызывает переход из состояния z_i в состояние z_j , то на графе автомата дуга, соединяющая вершину z_i с вершиной z_j , обозначается x_k . Для того чтобы задать функцию выходов, дуги графа необходимо отметить соответствующими выходными сигналами. Для автоматов Мили эта разметка производится так: если входной сигнал x_k действует на состояние z_i , то, согласно сказанному, получается дуга, исходящая

из z_i и помеченная x_k ; эту дугу дополнительно отмечают выходным сигналом $y = \psi(z_i, x_k)$. Для автомата Мура аналогичная разметка графа такова: если входной сигнал x_k , действуя на некоторое состояние автомата, вызывает переход в состояние z_j , то дугу, направленную в z_j и помеченную x_k , дополнительно отмечают выходным сигналом $y = \psi(z_j, x_k)$.

Таблица 2.3

x_i	z_k		
	z_0	z_1	z_2
Переходы			
x_1	z_2	z_0	z_0
x_2	z_0	z_2	z_1
Выходы			
x_1	y_1	y_1	y_2
x_2	y_1	y_2	y_1

Таблица 2.4

x_i	y				
	y_1	y_1	y_3	y_2	y_3
	z_0	z_1	z_2	z_3	z_4
x_1	z_1	z_4	z_4	z_2	z_2
x_2	z_3	z_1	z_1	z_0	z_0

На рис. 2.3, а, б приведены заданные ранее таблицами *F*-автомата Мили *F1* и Мура *F2* соответственно.

При решении задач моделирования систем часто более удобной формой является матричное задание конечного автомата. При этом матрица соединений автомата есть квадратная матрица $S = \|c_{ij}\|$, строки которой соответствуют исходным состояниям, а столбцы — состояниям перехода. Элемент $c_{ij} = x_k/y_s$, стоящий на пересечении *i*-й строки и *j*-го столбца, в случае автомата Мили соответствует входному сигналу x_k , вызывающему переход из состояния z_i в состояние

z_j и выходному сигналу y_s , выдаваемому при этом переходе. Для автомата Мили *F1*, рассмотренного выше, матрица соединений имеет вид

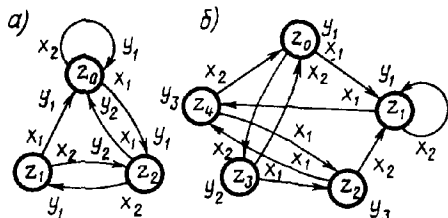


Рис. 2.3. Графы автоматов Мили (а) и Мура (б)

$$C_1 = \left\| \begin{array}{ccc} x_2/y_1 & - & x_1/y_1 \\ x_1/y_1 & - & x_2/y_2 \\ x_1/y_2 & x_2/y_1 & - \end{array} \right\|.$$

Если переход из состояния z_i в состояние z_j происходит под действием нескольких сигналов, элемент матрицы c_{ij} представляет собой множество пар «вход-выход» для этого перехода, соединенных знаком дизъюнкции.

Для *F-автомата* Мура элемент c_{ij} равен множеству входных сигналов на переходе (z_i, z_j) , а выход описывается вектором выходов

$$\vec{y} = \begin{pmatrix} \psi(z_0) \\ \psi(z_1) \\ \dots \\ \psi(z_k) \\ \dots \\ \psi(z_R) \end{pmatrix},$$

i -я компонента которого — выходной сигнал, отмечающий состояние z_i .

Пример 2.2. Для рассмотренного выше *F-автомата* Мура *F2* запишем матрицу соединений и вектор выходов:

$$C_2 = \begin{pmatrix} - & x_1 & - & x_2 & - \\ - & x_2 & - & - & x_1 \\ - & x_2 & - & - & x_1 \\ x_2 & - & x_1 & - & - \\ x_2 & - & x_1 & - & - \end{pmatrix}; \quad \vec{y} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{pmatrix}.$$

Для детерминированных автоматов выполняется условие однозначности переходов: автомат, находящийся в некотором состоянии, под действием любого входного сигнала не может перейти более чем в одно состояние. Применительно к графическому способу задания *F-автомата* это означает, что в графе автомата из любой вершины не могут выходить два ребра и более, отмеченные одним и тем же входным сигналом. Аналогично этому в матрице соединений автомата C в каждой строке любой входной сигнал не должен встречаться более одного раза.

Рассмотрим вид таблицы переходов и графа асинхронного конечного автомата. Для *F-автомата* состояние z_k называется *устойчивым*, если для любого входа $x_i \in X$, для которого $\varphi(z_k, x_i) = z_k$, имеет место $\psi(z_k, x_i) = y_k$. Таким образом, *F-автомат* называется *асинхронным*, если каждое его состояние $z_k \in Z$ устойчиво.

Необходимо отметить, что вообще на практике автоматы всегда являются асинхронными, а устойчивость их состояний обеспечивается тем или иным способом, например введением сигналов синхронизации. Однако на уровне абстрактной теории, когда конечный

автомат выступает в виде математической схемы для формализации конкретных объектов без учета ряда второстепенных особенностей, часто удобно оказывается оперировать с синхронными конечными автоматами.

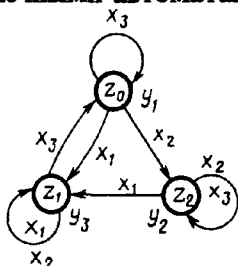


Рис. 2.4. Граф асинхронного автомата Мура

Таблица 2.5

x_i	y		
	y_1	y_2	y_3
	z_0	z_1	z_2
x_1	z_1	z_1	z_1
x_2	z_2	z_1	z_2
x_3	z_0	z_0	z_2

Пример 2.3. Рассмотрим асинхронный *F-автомат* Мура, который описан табл. 2.5 и приведен на рис. 2.4. Очевидно, что если в таблице переходов асинхронного автомата некоторое состояние z_k стоит на пересечении строки x_i и столбца z_s ($x \neq k$), то это состояние z_k обязательно должно встретиться в этой же строке в столбце z_k . В графе асинхронного автомата, если в некоторое состояние имеются переходы из других состояний под действием каких-то сигналов, то в вершине z_k должна быть петля, отмеченная символами тех же входных сигналов. Анализ табл. 2.3 и 2.4 или рис. 2.3, б и 2.4 показывает, что представленные там *F-автоматы* $F1$ и $F2$ являются синхронными.

Таким образом, понятие *F-автомата* в дискретно-детерминированном подходе к исследованию на моделях свойств объектов является математической абстракцией, удобной для описания широкого класса процессов функционирования реальных объектов в автоматизированных системах обработки информации и управления. В качестве таких объектов в первую очередь следует назвать элементы и узлы ЭВМ, устройства контроля, регулирования и управления, системы временной и пространственной коммутации в технике обмена информацией и т. д. Для всех перечисленных объектов характерно наличие дискретных состояний и дискретный характер работы во времени, т. е. их описание с помощью *F-схем* является эффективным. Но широта их применения не означает универсальности этих математических схем. Например, этот подход непригоден для описания процессов принятия решений, процессов в динамических системах с наличием переходных процессов и стохастических элементов.

2.4. ДИСКРЕТНО-СТОХАСТИЧЕСКИЕ МОДЕЛИ (P-СХЕМЫ)

Рассмотрим особенности построения математических схем при дискретно-стохастическом подходе к формализации процесса функционирования исследуемой системы S . Так как сущность дискрети-

зации времени при этом подходе остается аналогичной рассмотренным в § 2.3 конечным автоматам, то влияние фактора стохастичности проследим также на разновидности таких автоматов, а именно на вероятностных (стохастических) автоматах.

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

Применение схем вероятностных автоматов (*P-схем*) имеет важное значение для разработки методов проектирования дискретных систем, проявляющих статистически закономерное случайное поведение, для выяснения алгоритмических возможностей таких систем и обоснования границ целесообразности их использования, а также для решения задач синтеза по выбранному критерию дискретных стохастических систем, удовлетворяющих заданным ограничениям.

Введем математическое понятие *P-автомата*, используя понятия, введенные для *F-автомата*. Рассмотрим множество G , элементами которого являются всевозможные пары (x_i, z_i) , где x_i и z_i — элементы входного подмножества X и подмножества состояний Z соответственно. Если существуют две такие функции φ и ψ , то с их помощью осуществляются отображения $G \rightarrow Z$ и $G \rightarrow Y$, то говорят, что $F = \langle Z, X, Y, \varphi, \psi \rangle$ определяет автомат детерминированного типа.

Введем в рассмотрение более общую математическую схему. Пусть Φ — множество всевозможных пар вида (z_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:

$$\begin{array}{ccccccc} \text{Элементы из } \Phi & \dots & (z_1, y_1) \dots & (z_1, y_2) \dots & \dots & (z_k, y_{j-1}) & (z_k, y_j) \\ (x_i, z_k) & \dots & b_{11} & b_{12} & \dots & b_{k(j-1)} & b_{kj} \end{array}$$

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 , 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 $P = \langle Z, X, Y, B \rangle$ называется вероятностным автоматом (*P-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_k)	...	q_1	q_2	...	q_{j-1}	q_j
Элементы из Z	...	z_1	z_2	...	z_{k-1}	z_k
(x_i, z_k)	...	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 P -автомата в состояние z_k и появления выходного сигнала y_k при условии, что P -автомат находился в состоянии z_j и на его вход поступил входной сигнал x_i .

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

Пусть теперь определение выходного сигнала P -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_{l-1}	s_l

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

Возможные приложения. Если для всех k и i имеет место соотношение $z_k s_i = b_{ki}$, то такой P -автомат называется *вероятностным автоматом Мура*. Понятие P -автоматов Мили и Мура введено по аналогии с детерминированным F -автоматом, задаваемым $F = \langle Z, X, Y, \varphi, \psi \rangle$. Частным случаем P -автомата, задаваемого как $P = \langle Z, X, Y, B \rangle$, являются автоматы, у которых либо переход в новое состояние, либо выходной сигнал определяются детерминированно. Если выходной сигнал P -автомата определяется детерминированно, то такой автомат называется *Y -детерминированным вероятностным автоматом*. Аналогично, *Z -детерминированным вероятностным автоматом* называется P -автомат, у которого выбор нового состояния является детерминированным.

Пример 2.4. Рассмотрим Y -детерминированный P -автомат, который задан таблицей переходов (табл. 2.6) и таблицей выходов:

Z	...	z_1	z_2	...	z_{k-1}	z_k
Y	...	y_{11}	y_{12}	...	y_{1k-1}	y_{1k}

В этих таблицах p_{ij} — вероятность перехода P -автомата из состояния z_i в состояние z_j . При этом, как и ранее, $\sum_{j=1}^k p_{ij} = 1$.

Первую из этих таблиц можно представить в виде квадратной матрицы размерности $K \times K$, которую будем называть *матрицей переходных вероятностей* или просто *матрицей переходов P-автомата*. В общем случае такая матрица переходов имеет вид

$$P_P = \begin{pmatrix} p_{11} & p_{12} & \dots & p_{1K} \\ p_{21} & p_{22} & \dots & p_{2K} \\ \dots & \dots & \dots & \dots \\ p_{K1} & p_{K2} & \dots & p_{KK} \end{pmatrix}$$

Таблица 2.6

z_k	z_k				
	z_1	z_2	...	z_{K-1}	z_K
z_1	p_{11}	p_{12}	...	$p_{1(K-1)}$	p_{1K}
z_2	p_{21}	p_{22}	...	$p_{2(K-1)}$	p_{2K}
...
z_K	p_{K1}	p_{K2}	...	$p_{K(K-1)}$	p_{KK}

Для описания Y -детерминированного P -автомата необходимо задать начальное распределение вероятностей вида

$$\begin{matrix} Z & \dots & z_1 & z_2 & \dots & z_{K-1} & z_K \\ D & \dots & d_1 & d_2 & \dots & d_{K-1} & d_K \end{matrix}$$

Здесь d_k — вероятность того, что в начале работы P -автомат находится в состоянии k . При этом $\sum_{k=1}^K d_k = 1$.

Будем считать, что до начала работы (до нулевого такта времени) P -автомат всегда находится в состоянии z_0 и в нулевой такт времени меняет состояние в соответствии с распределением D . Дальнейшая смена состояний P -автомата определяется матрицей переходов P_P . Информацию о начальном состоянии P -автомата удобно внести в матрицу P_P , увеличив ее размерность до $(K+1) \times (K+1)$. При этом первая строка такой матрицы, сопоставляемая состоянию z_0 , будет иметь вид $(0, d_1, d_2, \dots, d_K)$, а первый столбец будет нулевым.

Описанный Y -детерминированный P -автомат можно задать в виде ориентированного графа, вершины которого сопоставляются состояниям автомата, а дуги — возможным переходам из одного состояния в другое. Дуги имеют веса, соответствующие вероятностям перехода p_{ij} , а около вершин графа пишутся значения выходных сигналов, индуцируемых этими состояниями.

Пример 2.5. Пусть задан Y -детерминированный P -автомат

$$P_P = \begin{pmatrix} 0 & 0,50 & 0 & 0 & 0,50 \\ 0 & 0 & 0 & 1,00 & 0 \\ 0 & 0 & 0,75 & 0 & 0,25 \\ 0 & 0 & 0,40 & 0 & 0,60 \\ 0 & 1,00 & 0 & 0 & 0 \end{pmatrix}; \quad \begin{matrix} Z & z_0 & z_1 & z_2 & z_3 & z_4 \\ Y & 0 & 0 & 1 & 1 & 0 \end{matrix}$$

На рис. 2.5 показан граф переходов этого автомата. Требуется оценить суммарные финальные вероятности пребывания этого *P*-автомата в состояниях z_2 и z_3 .

При использовании аналитического подхода можно записать известные соотношения из теории марковских цепей и получить систему уравнений для определения финальных вероятностей. При этом начальное состояние z_0 можно не учитывать, так как начальное распределение не оказывает влияния на значения финальных вероятностей. Тогда имеем

$$C = \vec{C} \begin{vmatrix} 0 & 0 & 1,00 & 0 \\ 0 & 0,75 & 0 & 0,25 \\ 0 & 0,40 & 0 & 0,60 \\ 1,00 & 0 & 0 & 0 \end{vmatrix}; \quad \vec{C} = (c_k) = (c_1, c_2, c_3, c_4),$$

где c_k — финальная вероятность пребывания *P*-автомата в состоянии z_k .
Получаем систему уравнений

$$\begin{cases} c_1 = c_4, \\ c_2 = 0,75c_2 + 0,40c_3, \\ c_3 = c_1, \\ c_4 = 0,25c_2 + 0,60c_3. \end{cases}$$

Добавим к этим уравнениям условие нормировки $c_1 + c_2 + c_3 + c_4 = 1$. Тогда, решая систему уравнений, получим $c_1 = 5/23$, $c_2 = 8/23$, $c_3 = 5/23$, $c_4 = 5/23$. Таким образом, $c_2 + c_3 = 13/23 = 0,5652$. Другими словами, при бесконечной работе заданного в этом примере Y -детерминированного *P*-автомата на его выходе формируется двоичная последовательность с вероятностью появления единицы, равной 0,5652.

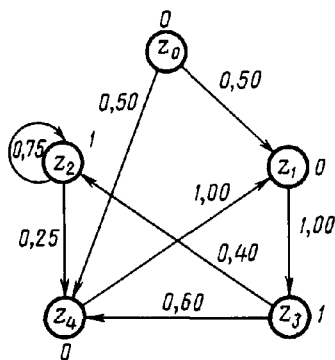


Рис. 2.5. Граф вероятностного автомата

Подобные *P*-автоматы могут использоваться как генераторы марковских последовательностей, которые необходимы при построении и реализации процессов функционирования систем *S* или воздействий внешней среды *E*.

Для оценки различных характеристик исследуемых систем, представляемых в виде *P*-схем, кроме рассмотренного случая аналитических моделей можно применять и имитационные модели, реализуемые, например, методом статистического моделирования.

2.5. НЕПРЕРЫВНО-СТОХАСТИЧЕСКИЕ МОДЕЛИ (Q-СХЕМЫ)

Особенности непрерывно-стохастического подхода рассмотрим на примере использования в качестве типовых математических схем систем массового обслуживания (англ. queueing system), которые

будем называть *Q-схемами*. Системы массового обслуживания представляют собой класс математических схем, разработанных в теории массового обслуживания и различных приложениях для формализации процессов функционирования систем, которые по своей сути являются процессами обслуживания [6, 13, 33, 37, 51].

Основные соотношения. В качестве процесса обслуживания могут быть представлены различные по своей физической природе процессы функционирования экономических, производственных, технических и других систем, например потоки поставок продукции некоторому предприятию, потоки деталей и комплектующих изделий на сборочном конвейере цеха, заявки на обработку информации ЭВМ от удаленных терминалов и т. д. При этом характерным для работы таких объектов является случайное появление заявок (требований) на обслуживание и завершение обслуживания в случайные моменты времени, т. е. стохастический характер процесса их функционирования. Остановимся на основных понятиях массового обслуживания, необходимых для использования *Q-схем*, как при аналитическом, так и при имитационном.

В любом элементарном акте обслуживания можно выделить две основные составляющие: ожидание обслуживания заявкой и собственно обслуживание заявки. Это можно изобразить в виде некоторого *i*-го прибора обслуживания Π_i (рис. 2.6), состоящего из накопителя заявок H_i , в котором может одновременно находиться $l_i = 0, \overline{L_i^H}$ заявок, где L_i^H — емкость *i*-го накопителя, и канала обслуживания заявок (или просто канала) K_i . На каждый элемент прибора обслуживания Π_i поступают потоки событий: в накопитель H_i — поток заявок w_i , на канал K_i — поток обслуживаний u_i .

Потоком событий называется последовательность событий, происходящих одно за другим в какие-то случайные моменты времени. Различают потоки однородных и неоднородных событий. Поток событий называется *однородным*, если он характеризуется только моментами поступления этих событий (вызывающими моментами) и задается последовательностью $\{t_n\} = \{0 \leq t_1 \leq t_2 \leq t_3 \leq \dots\}$, где t_n — момент наступления *n*-го события — неотрицательное вещественное число. Однородный поток событий также может быть задан в виде последовательности промежутков времени между *n*-м и (*n* - 1)-м событиями $\{\tau_n\}$, которая однозначно связана с последовательностью вызывающих моментов $\{t_n\}$, где $\tau_n = t_n - t_{n-1}$, $n \geq 1$, $t_0 = 0$, т. е. $\tau_1 = t_1$.

Потоком неоднородных событий называется последовательность $\{t_n, f_n\}$, где t_n — вызывающие моменты; f_n — набор признаков события. Например, применительно к процессу обслуживания для неоднородного потока заявок могут быть

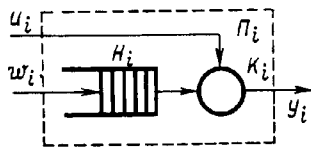


Рис. 2.6. Прибор обслуживания заявок

заданы принадлежность к тому или иному источнику заявок, наличие приоритета, возможность обслуживания тем или иным типом канала и т. п.

Рассмотрим поток, в котором события разделены интервалами времени τ_1, τ_2, \dots , которые вообще являются случайными величинами. Пусть интервалы τ_1, τ_2, \dots независимы между собой. Тогда поток событий называется *потоком с ограниченным последствием*.

Пример потока событий приведен на рис. 2.7, где обозначено T_j — интервал между событиями (случайная величина); T_n — время наблюдения, T_c — момент совершения события.

Интенсивность потока можно рассчитать экспериментально по формуле

$$\lambda = \frac{N}{T_n},$$

где N — число событий, произошедших за время наблюдения T_n . Если $T_j = \text{const}$ или определено какой-либо формулой $T_j = f(T_{j-1})$, то поток называется *детерминированным*. Иначе поток называется *случайным*.

Случайные потоки бывают:

- *ординарными*, когда вероятность одновременного появления 2-х и более событий равна нулю;
- *стационарными*, когда частота появления событий постоянная;
- *без последствия*, когда вероятность не зависит от момента совершения предыдущих событий.

Поток событий называется *ординарным*, если вероятность того, что на малый интервал времени Δt , примыкающий к моменту времени t , попадет больше одного события $P_{>1}(t, \Delta t)$, пренебрежительно мала по сравнению с вероятностью того, что на этот же интервал времени Δt попадает ровно одно событие $P_1(t, \Delta t)$, т. е. $P_1(t, \Delta t) \gg P_{>1}(t, \Delta t)$. Если для любого интервала Δt событие

$$P_0(t, \Delta t) + P_1(t, \Delta t) + P_{>1}(t, \Delta t) = 1$$

как сумма вероятностей событий, образующих полную группу и несовместных, то для ординарного потока событий

$$P_0(t, \Delta t) + P_1(t, \Delta t) \approx 1, \quad P_{>1}(t, \Delta t) = 0(\Delta t),$$

где $0(\Delta t)$ — величина, порядок малости которой выше, чем Δt , т. е.

$$\lim_{\Delta t \rightarrow 0} [0(\Delta t)/\Delta t] = 0.$$

Стационарным потоком событий называется поток, для которого вероятность появления того или иного числа событий на интервале времени t зависит лишь от длины этого участка и не зависит от того, где на оси времени $0t$ взят этот участок.

Рассмотрим на оси времени $0t$ ординарный поток событий и найдем среднее число событий, наступающих на интервале времени Δt , примыкающем к моменту времени t . Получим

$$0 \cdot P_0(t, \Delta t) + 1 \cdot P_1(t, \Delta t) = P_1(t, \Delta t).$$

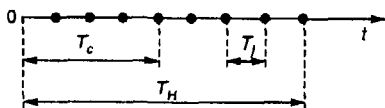


Рис. 2.7. Графическое изображение N -схемы

Тогда среднее число событий, наступающих на участке времени Δt в единицу времени, составит $[P_1(t, \Delta t)]/\Delta t$. Рассмотрим предел этого выражения при $\Delta t \rightarrow 0$. Если этот предел существует, то она называется *интенсивностью (плотностью)* ординарного потока событий $\lim_{\Delta t \rightarrow 0} [P_1(t, \Delta t)]/\Delta t$.

$\Delta t)/\Delta t] = \lambda(t)$. Интенсивность потока может быть любой неотрицательной функцией времени, имеющей размерность, обратную размерности времени. Для стационарного потока его интенсивность не зависит от времени и представляет собой постоянное значение, равное среднему числу событий, наступающих в единицу времени $\lambda(t) = \lambda = \text{const}$.

Возможные приложения. Обычно в приложениях при моделировании различных систем применительно к элементарному каналу обслуживания K_i можно считать, что поток заявок $w_i \in W$, т. е. интервалы времени между моментами появления заявок (вызывающие моменты) на входе K_i , образует подмножество неуправляемых переменных, а поток обслуживания $u_i \in U$, т. е. интервалы времени между началом и окончанием обслуживания заявки, образует подмножество управляемых переменных.

Заявки, обслуженные каналом K_i , и заявки, покинувшие прибор Π_i по различным причинам необслуженными (например, из-за переполнения накопителя H_i), образуют выходной поток $y_i \in Y$, т. е. интервалы времени между моментами выхода заявок образуют подмножество выходных переменных.

Процесс функционирования прибора обслуживания Π_i можно представить как процесс изменения состояний его элементов во времени $z_i(t)$. Переход в новое состояние для Π_i означает изменение количества заявок, которые в нем находятся (в канале K_i и в накопителе H_i). Таким образом, вектор состояний для Π_i имеет вид $\vec{z}_i = (z_i^H, z_i^K)$, где z_i^H — состояние накопителя H_i ($z_i^H = 0$ — накопитель пуст, $z_i^H = 1$ — в накопителе имеется одна заявка, ..., $z_i^H = L_i^H$ — накопитель полностью заполнен); L_i^H — емкость накопителя H_i , измеряемая числом заявок, которые в нем могут поместиться; z_i^K — состояние канала K_i ($z_i^K = 0$ — канал свободен, $z_i^K = 1$ — канал занят и т. д.).

В практике моделирования систем, имеющих более сложные структурные связи и алгоритмы поведения, для формализации используются не отдельные приборы обслуживания, а *Q-схемы*, образуемые композицией многих элементарных приборов обслуживания Π_i (сети массового обслуживания). Если каналы K_i различных приборов обслуживания соединены параллельно, то имеет место многоканальное обслуживание (многоканальная *Q-схема*), а если приборы Π_i и их параллельные композиции соединены последовательно, то имеет место многофазное обслуживание (многофазная *Q-схема*). Таким образом, для задания *Q-схемы* необходимо использовать оператор сопряжения R , отражающий взаимосвязь элементов структуры (каналов и накопителей) между собой.

Связи между элементами *Q-схемы* изображают в виде стрелок (линий потока, отражающих направление движения заявок). Различают разомкнутые и замкнутые *Q-схемы*. В разомкнутой *Q-схеме* выходной поток обслуженных заявок не может снова поступить на какой-либо элемент, т. е. обратная связь отсутствует, а в замкнутых

Q-схемах имеются обратные связи, по которым заявки двигаются в направлении, обратном движению вход-выход.

Собственными (внутренними) параметрами *Q*-схемы будут являться количество фаз L_ϕ , количество каналов в каждой фазе L_{kj} , $j=1, L^\phi$, количество накопителей каждой фазы L_{nk} , $k=1, L_\phi$, емкость i -го накопителя L_i^H . Следует отметить, что в теории массового обслуживания в зависимости от емкости накопителя применяются следующую терминологию для систем массового обслуживания: системы с потерями ($L_i^H=0$, т. е. накопитель в приборе Π_i отсутствует, а имеется только канал обслуживания K_i), системы с ожиданием ($L_i^H \rightarrow \infty$, т. е. накопитель H_i имеет бесконечную емкость и очередь заявок не ограничивается) и системы смешанного типа (с ограниченной емкостью накопителя H_i). Всю совокупность собственных параметров *Q*-схемы обозначим как подмножество H .

Для задания *Q*-схемы также необходимо описать алгоритмы ее функционирования, которые определяют набор правил поведения заявок в системе в различных неоднозначных ситуациях. В зависимости от места возникновения таких ситуаций различают алгоритмы (дисциплины) ожидания заявок в накопителе H_i и обслуживания заявок каналом K_i каждого элементарного обслуживающего прибора Π_i *Q*-схемы. Неоднородность заявок, отражающая процесс в той или иной реальной системе, учитывается с помощью введения классов приоритетов.

В зависимости от динамики приоритетов в *Q*-схемах различают статические и динамические приоритеты. Статические приоритеты назначаются заранее и не зависят от состояний *Q*-схемы, т. е. они являются фиксированными в пределах решения конкретной задачи моделирования. Динамические приоритеты возникают при моделировании в зависимости от возникающих ситуаций. Исходя из правил выбора заявок из накопителя H_i на обслуживание каналом K_i , можно выделить относительные и абсолютные приоритеты. *Относительный приоритет* означает, что заявка с более высоким приоритетом, поступившая в накопитель H_i , ожидает окончания обслуживания предшествующей заявки каналом K_i и только после этого занимает канал. *Абсолютный приоритет* означает, что заявка с более высоким приоритетом, поступившая в накопитель H_i , прерывает обслуживание каналом K_i заявки с более низким приоритетом и сама занимает канал (при этом вытесненная из K_i заявка может либо покинуть систему, либо может быть снова записана на какое-то место в H_i).

При рассмотрении алгоритмов функционирования приборов обслуживания Π_i (каналов K_i и накопителей H_i) необходимо также задать набор правил, по которым заявки покидают H_i и K_i ; для H_i — либо правила переполнения, по которым заявки в зависимости от заполнения H_i покидают систему, либо правила ухода,

связанные с истечением времени ожидания заявки в H_i , для K_i — правила выбора маршрутов или направлений ухода. Кроме того, для заявок необходимо задать правила, по которым они остаются в канале K_i или не допускаются до обслуживания каналом K_i , т. е. правила блокировок канала. При этом различают блокировки K_i по выходу и по входу. Такие блокировки отражают наличие управляющих связей в Q -схеме, регулирующих поток заявок в зависимости от состояний Q -схемы. Весь набор возможных алгоритмов поведения заявок в Q -схеме можно представить в виде некоторого оператора алгоритмов поведения заявок A .

Таким образом, Q -схема, описывающая процесс функционирования системы массового обслуживания любой сложности, однозначно задается в виде $Q = \langle W, U, H, Z, R, A \rangle$.

При ряде упрощающих предположений относительно подмножеств входящих потоков W и потоков обслуживания U (выполнение условий стационарности, ординарности и ограниченного последствия) оператора сопряжения элементов структуры R (однофазное одноканальное обслуживание в разомкнутой системе), подмножества собственных параметров H (обслуживание с бесконечной емкостью накопителя), оператора алгоритмов обслуживания заявок A (бесприоритетное обслуживание без прерываний и блокировок) для оценки вероятностно-временных характеристик можно использовать аналитический аппарат, разработанный в теории массового обслуживания. При принятых предположениях в обозначениях Д. Кендалла будет иметь место классическая система обслуживания типа М/М/1 (одноканальная система с марковским входящим потоком заявок и марковским потоком обслуживания). Рассмотрим на примере основные аналитические соотношения для такой элементарной Q -схемы [6, 24, 37].

Пример 2.6. Допустим, что процесс обслуживания начинается при отсутствии заявок в накопителе. Тогда состояния системы массового обслуживания описываются следующей системой уравнений:

$$\begin{cases} P_n(t + \Delta t) = P_n(t) [1 - (\lambda + \mu)\Delta t] + P_{n-1}(t)\lambda\Delta t + P_{n+1}(t)\mu\Delta t, & n \geq 1, \\ P_0(t + \Delta t) = P_0(t) (1 - \lambda\Delta t) + P_1(t)\mu\Delta t, \end{cases}$$

где $P_n(t)$ — вероятность нахождения системы в состоянии $z_n(t) \in Z$ в момент времени t , т. е. когда в ней имеется n заявок.

Эти уравнения следуют из того, что вероятность нахождения в системе n заявок в момент времени $(t + \Delta t)$ равна вероятности нахождения в системе n заявок в момент t , умноженной на вероятность того, что за время Δt в систему не поступит ни одной заявки и ни одна заявка не будет обслужена, плюс вероятность нахождения в системе $(n - 1)$ заявок в момент t , умноженная на вероятность того, что за время Δt поступит одна заявка и ни одна заявка не будет обслужена, плюс вероятность нахождения в системе $(n + 1)$ заявок в момент t , умноженная на вероятность того, что за время Δt одна заявка покинет систему и не поступит ни одной заявки. Вероятность того, что за время Δt не поступит ни одной заявки и ни одна заявка не покинет систему, равна $(1 - \lambda\Delta t)(1 - \mu\Delta t)$. Член, содержащий $(\Delta t)^2$, при составлении дифференциального уравнения опускается. Следовательно, можно записать $1 - (\lambda + \mu)\Delta t$. Относительно остальных двух членов первого уравнения заметим, что

$$\lambda \Delta t (1 - \mu \Delta t) \approx \lambda \Delta t, \quad \mu \Delta t (1 - \lambda \Delta t) \approx \mu \Delta t.$$

Перенеся $P_n(t)$ влево и устремив Δt к нулю, получим систему дифференциальных уравнений

$$\begin{cases} dP_n(t)/dt = -(\lambda + \mu)P_n(t) + \lambda P_{n-1}(t) + \mu P_{n+1}(t), & n \geq 1, \\ dP_0(t)/dt = -\lambda P_0(t) + \mu P_1(t). \end{cases}$$

Найдем выражение для математического ожидания числа заявок, находящихся в накопителе, и среднего времени ожидания заявок в накопителе для стационарного состояния $\rho = \lambda/\mu < 1$. Приравняв нулю производные по времени и исключив, таким образом, время t из уравнений, получим систему алгебраических уравнений

$$\begin{cases} (\lambda + \mu)p_n = \lambda p_{n-1} + \mu p_{n+1}, & n \geq 1, \\ \lambda p_0 = \mu p_1, \end{cases} \quad \begin{cases} (1 + \rho)p_n = p_{n+1} + \rho p_{n-1}, & n \geq 1, \\ p_1 = \rho p_0. \end{cases}$$

Пусть в первом уравнении $n=1$. Тогда $(1 + \rho)p_1 = p_2 + \rho p_0$. Подставив сюда значение p_1 из второго уравнения, находим $p_2 = \rho^2 p_0$. Повторяя эти операции,

получаем $p_n = \rho^n p_0$, причем $\sum_{n=0}^{\infty} p_n = 1$, так как это сумма вероятностей того, что

в системе нет ни одной заявки, имеется одна заявка, две заявки и т. д. Сумма этих вероятностей должна быть равна единице, так как рассматриваются все возможные состояния системы. Поэтому $\sum_{n=0}^{\infty} \rho^n p_0 = 1$,

или $\sum_{n=0}^{\infty} p_0 \rho^n = p_0 \sum_{n=0}^{\infty} \rho^n = p_0 / (1 - \rho) = 1$, откуда $p_0 = 1 - \rho$. Следовательно, $p_n = \rho^n (1 - \rho)$.

Полученное выражение представляет собой геометрическое распределение.

Математическое ожидание числа заявок, находящихся в системе (приборе),

$$\bar{l}_n = \sum_{n=0}^{\infty} n p_n = (1 - \rho) \sum_{n=0}^{\infty} n \rho^n = \rho (1 - \rho).$$

Отметим, что \bar{l}_n — среднее значение и возможны колебания числа заявок, ожидающих обслуживания, что можно оценить с помощью дисперсии:

$$D[l_n] = \sum_{n=0}^{\infty} (n - \bar{l}_n)^2 p_n = \sum_{n=0}^{\infty} n^2 p_n - [\rho / (1 - \rho)]^2.$$

При этом

$$\sum_{n=0}^{\infty} n^2 p_n = (1 - \rho) \sum_{n=0}^{\infty} n^2 \rho^n = \rho / (1 - \rho) + 2\rho^2 / (1 - \rho)^2.$$

Следовательно,

$$D[l_n] = \rho / (1 - \rho) + 2\rho^2 / (1 - \rho)^2.$$

Математическое ожидание числа заявок, находящихся в накопителе,

$$\bar{l}_n = \sum_{n=1}^{\infty} (n-1) p_n = \bar{l}_n - \rho = \rho / (1 - \rho) - \rho = \rho^2 / (1 - \rho).$$

Среднее время ожидания заявок в накопителе

$$\bar{l}_n = \bar{l}_n / \lambda = \rho^2 / [\lambda (1 - \rho)].$$

Возможности оценки характеристик с использованием аналитических моделей теории массового обслуживания являются весьма ограниченными по сравнению с требованиями практики исследования и проектирования систем, формализуемых в виде Q -схем. Несравненно большими возможностями обладают имитационные модели, позволяющие исследовать Q -схему, задаваемую $Q = \langle W, U, H, Z, Y, R, A \rangle$, без ограничений. На работу с Q -схемами при машинной реализации моделей ориентированы многие языки имитационного моделирования, например *SIMULA*, *SIMSCRIPT*, *GPSS* и др. Детально вопросы, связанные с имитационным моделированием Q -схем, будут рассмотрены далее.

2.6. 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 протекает несколько процессов. Самым распространенным в настоящее время формализмом, описывающим структуру и взаимодействие параллельных систем и процессов, являются сети Петри (англ. Petri Nets), предложенные К. Петри [28, 30].

Основные соотношения. Теория сетей Петри развивается в нескольких направлениях: разработка математических основ, структурная теория сетей, различные приложения (параллельное программирование, дискретные динамические системы и т. д.).

Формально сеть Петри (N -схема) 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, $D \neq \emptyset$, $B \cap D \neq \emptyset$; 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 в множество входных позиций $b_i \in I(d_j)$, а выходная функция O отображает переход d_j в 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

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

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

$$i = \overline{1, n}; j = \overline{1, m}, n = |B|, m = |D|.$$

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

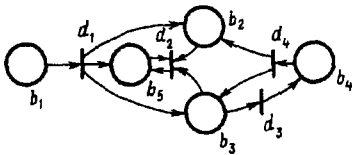


Рис. 2.8. Графическое изображение N -схемы

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

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

Графически N -схема изображается в виде двудольного ориентированного мультиграфа, представляющего собой совокупность позиций и переходов (рис. 2.8). 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.

Пример 2.7. Представим формально N -схему, показанную в виде графа на рис. 2.7:

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

$$B = \langle b_1, b_2, b_3, b_4, b_5 \rangle,$$

$$D = \langle d_1, d_2, d_3, d_4 \rangle,$$

$$I(d_1) = \{b_1\},$$

$$O(d_1) = \{b_2, b_3, b_5\},$$

$$I(d_2) = \{b_2, b_3, b_5\},$$

$$O(d_2) = \{b_5\},$$

$$I(d_3) = \{b_3\},$$

$$O(d_3) = \{b_4\},$$

$$I(d_4) = \{b_4\},$$

$$O(d_4) = \{b_2, b_3\}.$$

Возможные приложения. Приведенное представление N -схемы может использоваться только для отражения статике моделируемой системы (взаимосвязи событий и условий), но не позволяет отразить в модели динамику функционирования моделируемой системы. Для представления динамических свойств объекта вводится функция маркировки (разметки) $M : B \rightarrow \{0, 1, 2, \dots\}$. Маркировка M есть присвоение неких абстрактных объектов, называемых метками (фишками), позициям N -схемы, причем количество меток, соответствующее каждой позиции, может меняться. При графическом задании N -схемы разметка отображается помещением внутри вершин-позиций соответствующего числа точек (когда количество точек велико, ставят цифры).

Маркированная (размеченная) N -схема может быть описана в виде пятерки $N_M = \langle B, D, I, O, M \rangle$ и является совокупностью сети Петри и маркировки M [28, 30].

Функционирование N -схемы отражается путем перехода от разметки к разметке. Начальная разметка обозначается как $M_0 : B \rightarrow \{0, 1, 2, \dots\}$. Смена разметок происходит в результате срабатывания

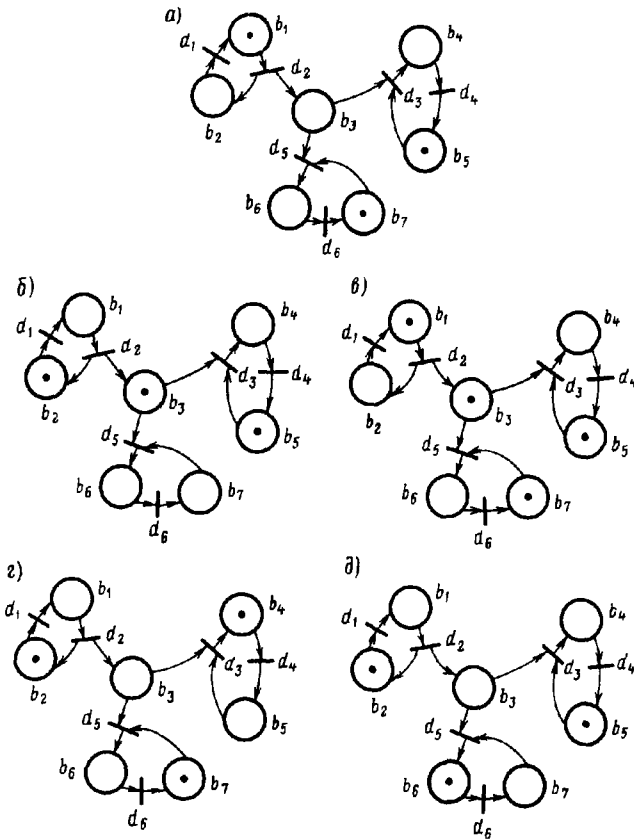


Рис. 2.9. Пример функционирования размеченной N -схемы

одного из переходов $d_j \in D$ сети. Необходимым условием срабатывания перехода d_j является $b_i \in I(d_j) \{M(b_i) \geq 1\}$, где $M\{b_i\}$ — разметка позиции b_i . Переход d_j , для которого выполняется указанное условие, определяется как находящийся в состоянии готовности к срабатыванию или как возбужденный переход.

Срабатывание перехода d_j изменяет разметку сети $M(b) = (M(b_1), M(b_2), \dots, M(b_n))^2$ на разметку $M'(b)$ по следующему правилу:

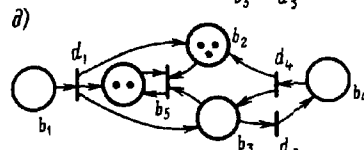
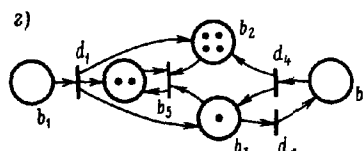
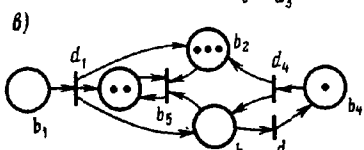
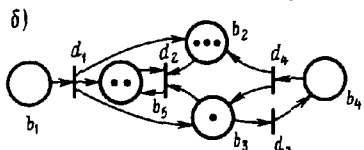
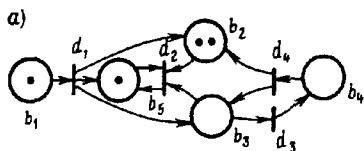
$$M'(b) = M(b) - I(d_j) + O(d_j),$$

т. е. переход d_j изымает по одной метке из каждой своей входной позиции и добавляет по одной метке в каждую из выходных позиций. Для изображения смены разметки M на M' применяют обозначение $M \stackrel{d_j}{\rightarrow} M'$.

Пример 2.8. Рассмотрим размеченную N -схему с начальной разметкой $M_0 = \{1, 0, 0, 0, 1, 0, 1\}$, которая приведена на рис. 2.9, а. При такой начальной разметке N -схемы единственным готовым к срабатыванию является переход d_2 , срабатывание которого ведет к смене разметки $M_0 \xrightarrow{d_2} M_1$, где $M_1 = \{0, 1, 1, 0, 1, 0, 1\}$ (рис. 2.9, б). При разметке M_1 возможно срабатывание переходов d_1 , d_3 и d_5 . В зависимости от того, какой переход сработал первым, получается одна из трех возможных новых маркировок (рис. 2.9, в, г, д). Функционирование N -схемы продолжается до тех пор, пока существует хотя бы один возможный переход.

Таким образом, N -схема выполняется путем запусков переходов под управлением количества меток и их распределения в сети. Переход запускается удалением меток из его входных позиций и образованием новых меток, помещаемых в выходные позиции. Переход может запускаться только тогда, когда он разрешен. Переход называется *разрешенным*, если каждая из его входных позиций имеет число меток, по крайней мере равное числу дуг из позиции в переход.

Пример 2.9. Для некоторой заданной размеченной N -схемы (рис. 2.8) с начальной маркировкой $M_0 = \{1, 2, 0, 0, 1\}$ (рис. 2.10, а)



разрешенным является только переход d_1 , а остальные переходы d_2 , d_3 и d_4 — запрещенные. В результате выполнения этого перехода получим новую размеченную N -схему (рис. 2.10, б). Теперь разрешены переходы d_1 и d_5 ; в результате их запуска получим новую размеченную N -схему. Переходы d_2 и d_3 накладываются в конфликт, так как запуском может быть только один из них. Например, при запуске d_3 получим сеть, показанную на рис. 2.10, в. Теперь разрешен только переход d_4 и получим новую размеченную сеть (рис. 2.10, г). Теперь разрешено два перехода: d_2 и d_3 (в конфликте). Запустим переход d_2 (рис. 2.10, д). Теперь ни один переход не может быть запущен и выполнение сети прекращается.

Важной особенностью моделей процесса функционирования систем с использованием типовых N -схем является простота построения иерархических конструкций модели. С одной стороны, каждая N -схема может рассматриваться как макропереход или макропозиция модели более высокого уровня. С другой стороны, переход, или позиция N -схемы, может детализироваться в форме отдельной подсети для более углубленного исследования процессов в моделируемой системе S . Отсюда вытекает возможность эффективного использования N -схем

Рис. 2.10. Пример функционирования размеченной заданной N -схемы

для моделирования параллельных и конкурирующих процессов в различных системах.

Типовые N -схемы на основе обычных размеченных сетей Петри пригодны для описания в моделируемой системе S событий произвольной длительности. В этом случае модель, построенная с использованием таких N -схем, отражает только порядок наступления событий в исследуемой системе S . Для отражения временных параметров процесса функционирования моделируемой системы S на базе N -схем используется расширение аппарата сетей Петри: временные сети, E -сети, сети Мерлина и т. д. [19]. Детально вопросы, связанные с имитационным моделированием с использованием N -схем, будут рассмотрены далее.

2.7. КОМБИНИРОВАННЫЕ МОДЕЛИ (A -СХЕМЫ)

Наиболее известным общим подходом к формальному описанию процессов функционирования систем является подход, предложенный *Н. П. Бусленко*. Этот подход позволяет описывать поведение непрерывных и дискретных, детерминированных и стохастических систем, т. е. по сравнению с рассмотренными является обобщенным (универсальным) и базируется на понятии *агрегативной системы* (от англ. aggregate system), представляющей собой формальную схему общего вида, которую будем называть *A -схемой* [4, 35].

Основные соотношения. Анализ существующих средств моделирования систем и задач, решаемых с помощью метода моделирования на ЭВМ, неизбежно приводит к выводу, что комплексное решение проблем, возникающих в процессе создания и машинной реализации модели, возможно лишь в случае, если моделирующие системы имеют в своей основе единую формальную математическую схему, т. е. *A -схему*. Такая схема должна одновременно выполнять несколько функций: являться адекватным математическим описанием объекта моделирования, т. е. системы S , служить основой для построения алгоритмов и программ при машинной реализации модели M , позволять в упрощенном варианте (для частных случаев) проводить аналитические исследования.

Приведенные требования в определенной степени противоречивы. Тем не менее в рамках обобщенного подхода на основе *A -схем* удается найти между ними некоторый компромисс.

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 [35].

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* ∈ *T* обозначается как *z(t)* ∈ *Z*, а входные и выходные сигналы — как *x(t)* ∈ *X* и *y(t)* ∈ *Y* соответственно [4].

Будем полагать, что переход агрегата из состояния *z(t₁)* в состояние *z(t₂)* ≠ *z(t₁)* происходит за малый интервал времени, т. е. имеет место скачок δz . Переходы агрегата из состояния *z(t₁)* в *z(t₂)* определяются собственными (внутренними) параметрами самого агрегата *h(t)* ∈ *H* и входными сигналами *x(t)* ∈ *X*.

В начальный момент времени *t₀* состояния *z* имеют значения, равные *z⁰*, т. е. *z⁰ = z(t₀)*, задаваемые законом распределения процесса *z(t)* в момент времени *t₀*, а именно *L[z(t₀)]*. Предположим, что процесс функционирования агрегата в случае воздействия входного сигнала *x_n* описывается случайным оператором *V*. Тогда в момент поступления в агрегат *t_n* ∈ *T* входного сигнала *x_n* можно определить состояние

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

Обозначим полуинтервал времени *t₁* < *t* ≤ *t₂* как [*t₁*, *t₂*], а полуинтервал *t₁* ≤ *t* < *t₂* — как [*t₁*, *t₂*). Если интервал времени (*t_n*, *t_{n+1}*) не содержит ни одного момента поступления сигналов, то для *t* ∈ (*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 накладываются 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_s*,

а состояния $z(t_s)$ — особыми состояниями A -схемы. Для описания скачков состояний dz в особые моменты времени t_s будем использовать случайный оператор W , представляющий собой частный случай оператора U , т. е.

$$z(t_s+0) = W[t_s, z(t_s)].$$

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

$$y = G[t_s, z(t_s)].$$

Таким образом, под *агрегатом* будем понимать любой объект, определяемый упорядоченной совокупностью рассмотренных множеств $T, X, Y, Z, Z^{(n)}, H$ и 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*.

Возможные приложения. Существует класс больших систем, которые ввиду их сложности не могут быть формализованы в виде математических схем одиночных агрегатов, поэтому их формализуют некоторой конструкцией из отдельных агрегатов $A_n, n = \overline{1, N_A}$, которую назовем агрегативной системой или A -схемой. Для описания некоторой реальной системы S в виде A -схемы необходимо иметь описание как отдельных агрегатов A_n , так и связей между ними.

Пример 2.10. Рассмотрим A -схему, структура которой приведена на рис. 2.11. Функционирование A -схемы связано с переработкой информации, передача последней на схеме показана стрелками. Вся информация, циркулирующая в A -схеме, делится на внешнюю и внутреннюю. Внешняя информация поступает от внешних объектов, не являющихся элементами рассматриваемой схемы, а внутренняя информация вырабатывается агрегатами самой A -схемы. Обмен информацией между A -схемой и внешней средой E происходит через агрегаты, которые называются *полосами A-схемы*. При этом различают входные полосы A -схемы, представляющие собой агрегаты, на которые поступают x -сообщения (агрегаты A_1, A_2, A_6), и выходные полосы A -схемы, выходящая информация которых является y -сообщениями (агрегаты A_1, A_3, A_4, A_5, A_6). Агрегаты, не являющиеся полосами, называются *внутренними*.

Каждый n -й агрегат A -схемы A_n имеет входные контакты, на которые поступает совокупность элементарных сигналов $x_i(t)$, $i = \overline{1, I_n}$, одновременно возникающих на входе элемента, и выходные контакты, с которых снимается совокупность элементарных сигналов $y_j(t)$, $j = \overline{1, J_n}$. Таким образом, каждый агрегат A -схемы A_n имеет I_n входных и J_n выходных контактов.

Описание отдельного агрегата уже рассмотрено, поэтому для построения формального понятия A -схемы остается выбрать

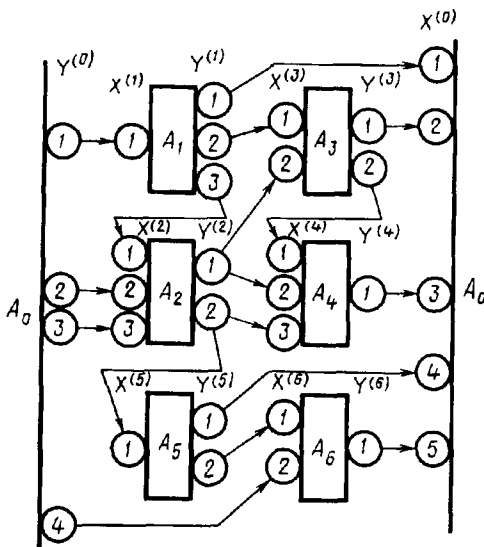


Рис. 2.11. Структура агрегативной системы

тарные сигналы мгновенно передаются в A -схеме независимо друг от друга по элементарным каналам; 4) к входному контакту любого элемента A -схемы подключается не более чем один элементарный канал, к выходному контакту — любое конечное число элементарных каналов при условии, что ко входу одного и того же элемента A -схемы направляется не более чем один из упомянутых элементарных каналов.

Взаимодействие A -схемы с внешней средой E рассматривается как обмен сигналами между внешней средой E и элементами A -схемы. В соответствии с этим внешнюю среду E можно представить в виде фиктивного элемента системы A_0 , вход которого содержит I_0 входных контактов $X_i^{(0)}$, $i = \overline{1, I_0}$, а выход — J_0 выходных контактов $Y_i^{(0)}$, $i = \overline{1, J_0}$. Сигнал, выдаваемый A -схемой во внешнюю среду E , принимается элементом A_0 как входной сигнал, состоящий из элементарных сигналов $x_1^{(0)}(t)$, $x_2^{(0)}(t)$, ..., $x_{I_0}^{(0)}(t)$. Сигнал, поступающий в A -схему из внешней среды E , является выходным сигналом элемента A_0 и состоит из элементарных сигналов $y_1^{(0)}(t)$, $y_2^{(0)}(t)$, ..., $y_{J_0}^{(0)}(t)$.

Таким образом, каждый A_n (в том числе и A_0) как элемент A -схемы в рамках принятых предположений о механизме обмена сигналами достаточно охарактеризовать множеством входных контактов $X_1^{(n)}$, $X_2^{(n)}$, ..., $X_{I_n}^{(n)}$, которое обозначим $\{X_i^{(n)}\}$, и множеством выходных контактов $Y_1^{(n)}$, $Y_2^{(n)}$, ..., $Y_{J_n}^{(n)}$, которое обозначим $\{Y_j^{(n)}\}$,

достаточно удобные способы математического описания взаимодействия между агрегатами. Для этого введем ряд предположений о закономерностях функционирования A -схем, хорошо согласующихся с опытом исследования реальных сложных систем [4]: 1) взаимодействие между A -схемой и внешней средой E , а также между отдельными агрегатами внутри системы S осуществляется при передаче сигналов, причем взаимные влияния, имеющие место вне механизма обмена сигналами, не учитываются; 2) для описания сигнала достаточно набора характеристик; 3) элементарные сигналы мгновенно передаются в A -схеме независимо друг от друга по элементарным каналам; 4) к входному контакту любого элемента A -схемы подключается не более чем один элементарный канал, к выходному контакту — любое конечное число элементарных каналов при условии, что ко входу одного и того же элемента A -схемы направляется не более чем один из упомянутых элементарных каналов.

где $n = \overline{0, N_A}$. Полученная пара множеств $\{X_i^{(n)}\}, \{Y_j^{(n)}\}$ является математической моделью элемента A_n , используемого для формального описания сопряжения его с прочими элементами A -схемы и внешней средой E .

В силу предположения о независимости передачи сигналов каждому входному контакту

$$X_i^{(n)} \in \bigcup_{n=0}^{N_A} \{X_i^{(n)}\}$$

соответствует не более чем один выходной контакт

$$Y_j^k \in \bigcup_{n=0}^{N_A} \{Y_j^{(n)}\},$$

где $\bigcup_{n=0}^{N_A} \{X_i^{(n)}\}$ — множество входных контактов всех элементов A -схемы и внешней среды E ; $\bigcup_{n=0}^{N_A} \{Y_j^{(n)}\}$ — множество выходных контактов всех элементов A -схемы и внешней среды E , с которыми она связана элементарным каналом; $k, n = \overline{0, N_A}$.

Поэтому можно ввести однозначный оператор $Y_j^k = R(X_i^{(n)})$ с областью определения в множестве $\bigcup_{n=0}^{N_A} \{X_i^{(n)}\}$ и областью значений

в множестве $\bigcup_{n=0}^{N_A} \{Y_j^{(n)}\}$, сопоставляющий входному контакту $X_i^{(n)}$ выходной контакт $Y_j^{(k)}$, связанный с ним элементарным каналом. Если в A -схеме к контакту $X_i^{(n)}$ не подключен никакой элементарный канал, то оператор R не определен на этом контакте $X_i^{(n)}$. Оператор R называется *оператором сопряжения* элементов (агрегатов) в A -схеме. Совокупность множеств $\{X_i^{(n)}\}, \{Y_j^{(k)}\}$ и оператор R образуют схему сопряжения элементов в систему S .

Рассмотрим оператор сопряжения для A -схемы, структура которой показана на рис. 2.11. Оператор сопряжения R можно задать в виде таблицы, в которой на пересечении строк с номерами элементов (агрегатов) n и столбцов с номерами контактов i располагаются пары чисел k, l , указывающие номер элемента k и номер контакта l , с которыми соединен контакт $X_i^{(n)}$ (табл. 2.7).

Если столбцы и строки такой таблицы пронумеровать двойными индексами n, i и k, l соответственно и на пересечении помещать 1 для контактов n, i и k, l , соединенных элементарным каналом и 0 в противном случае, то получим матрицу смежности ориентированного графа, вершинами которого являются контакты агрегатов, а дугами — элементарные каналы A -схемы.

Рассмотренная схема сопряжения агрегатов в A -схему, заданная совокупностью множеств $\{X_i^{(n)}\}$, $\{Y_j^{(n)}\}$ и оператором R , является одноуровневой схемой сопряжения. В более сложных случаях могут быть использованы многоуровневые иерархические схемы сопряжения. Схема сопряжения агрегата, определяемая оператором R , может быть использована для описания весьма широкого класса объектов. Однако взаимодействие элементов реальных систем даже в рамках механизма обмена сигналами не сводится к одному лишь сопряжению. Помимо сопряжения контактов серьезную роль играют также согласование совокупности элементарных сигналов, поступающих в элементарный канал от выходных контактов и воспринимаемых входными, а также влияние реальных средств передачи сигналов на их содержание. Кроме того, оказываются полезными некоторые дополнительные ограничения на структуру сопряжения агрегатов системы S с внешней средой E . Поэтому с практической точки зрения представляет интерес понятие A -схемы как типовой математической, отражающей наши представления о взаимодействии реальных объектов в рамках механизмов обмена сигналами.

Таблица 2.7

n	i				
	1	2	3	4	5
0	1,1	3,1	4,1	5,1	6,1
1	0,1				
2	1,3	0,2	0,3		
3	1,2	2,1			
4	3,2	2,1	2,2		
5	2,2				
6	5,2	0,4			

Упорядоченную совокупность конечного числа агрегатов A_n , $n = \overline{1, N_A}$ системы S , агрегата A_0 , характеризующего внешнюю среду E , и оператора R , реализующего отображение $\bigcup_{n=0}^{N_A} \{X_i^{(n)}\} \rightarrow \bigcup_{n=0}^{N_A} \{Y_j^{(n)}\}$, будем называть A -схемой при следующих условиях:

1) для любых $X_i^{(0)} \in \{X_i^{(0)}\}$ и $Y_j^{(0)} \in \{Y_j^{(0)}\}$ в данной A -схеме $Y_j^{(0)} \neq R(X_i^{(0)})$;

2) если $Y_j^{(0)} = R(X_i^{(n)})$, то

$$\dot{Y}_j^k \in \dot{X}_i^{(n)}, \quad (2.18)$$

где $\dot{Y}_j^{(k)}$ — соответствующие множества элементарных сигналов; для любого момента t' выдачи непустого элементарного сигнала

$$Y_j^k(t') \in \dot{Y}_j^k \quad (2.19)$$

имеет место

$$t' \in (T^k \cap T^{(n)}), \quad (2.20)$$

$$Y^{(k)}(t') = x_1^{(n)}(t'), \quad (2.21)$$

где $x_1^{(n)}(t') \in X_1^{(n)}$.

Ограничение (2.18) относится к структуре сопряжения агрегатов *A-схемы* системы *S* с внешней средой *E* и требует, чтобы каждый элементарный канал, передающий сигналы во внешнюю среду, начинался в одном из выходных контактов одного из агрегатов системы, каждый элементарный канал, передающий сигналы из внешней среды, заканчивался на одном из входных контактов *A-схемы*. Ограничение (2.19) предусматривает, что сигналы в *A-схеме* передаются непосредственно от одного агрегата к другому без устройств, способных отсеивать сигналы по каким-либо признакам. Ограничение (2.20) относится к согласованию функционирования агрегатов *A-схемы* во времени. Ограничение (2.21) предусматривает, что сигналы между агрегатами *A-схемы* передаются мгновенно, без искажений и перекодирования, изменяющего структуру сигнала. Для многих реальных систем ограничения (2.19) и (2.21) оказываются несправедливыми. Для того чтобы *A-схема* была адекватной моделью реального объекта, достаточно описать селективирующие устройства, реальные средства передачи сигналов и всевозможные вспомогательные устройства как самостоятельные агрегаты, связи между которыми удовлетворяют перечисленным ограничениям.

Пример 2.11. Рассмотрим представление некоторой системы в виде отдельного агрегата [4]. Для того чтобы упростить описание объекта моделирования и проследить связи с уже рассмотренными схемами, воспользуемся в качестве объекта такого моделирования схемой массового обслуживания (*Q-схемой*) и представим ее в виде агрегата (*A-схемы*). Для определенности полагаем, что имеется однофазная одноканальная система S_Q , показанная на рис. 2.6. В моменты времени t_j , образующие однородный поток случайных событий, в прибор (*П*) поступают заявки, каждая из которых характеризуется случайным параметром e_j . Если обслуживающий канал (*K*) занят, то заявка поступает в накопитель (*H*) и может ждать там не более чем $\gamma_j = \varphi(e_j, h)$, где h — параметр, характеризующий производительность системы обслуживания. Если к моменту $(t_j + \gamma_j)$ заявка не будет принята к обслуживанию, то она теряется. Время обслуживания заявки $\tau_j = \psi(e_j, h)$.

При представлении этой *Q-схемы* в виде *A-схемы* опишем ее состояния вектором $\vec{Z}(t) \in Z$ со следующими компонентами: $z_1(t)$ — время, оставшееся до окончания обслуживания заявки, которая находится в канале (*K*); $z_2(t)$ — количество заявок в приборе (*П*); $z_m(t) = e_k$, где e_k — параметр k -й заявки в накопителе (*H*); $z_2(t) \dots$ оставшееся время ожидания k -й заявки в накопителе (*H*) до момента, когда она получит отказ, $m = 1 + 2k$, $l = 2 + 2k$, $k = 1, z_2(t) - 1$.

Входные сигналы (заявки) поступают в *A-схему* в моменты t_j и принимают значения $x_j = e_j$. Рассмотрим случайные операторы V , U и G , описывающие такой агрегат. Пусть в момент t_j поступает новая заявка. Тогда оператор V можно записать следующим образом:

$$\left. \begin{aligned}
 z_1(t_j+0) &= z_1(t_j), \\
 z_2(t_j+0) &= z_2(t_j) + 1, \\
 z_m(t_j+0) &= z_m(t_j), \\
 z_l(t_j+0) &= z_l(t_j), \\
 z_{1+2k}(t_j+0) &= e_j, \\
 z_{2+2k}(t_j+0) &= \varphi(e_j, h), \\
 z_1(t_j+0) &= \psi(e_j, h) \\
 z_2(t_j+0) &= 1,
 \end{aligned} \right\} \begin{aligned}
 &k \leq z_2(t_j), \\
 &z_2(t_j) > 0; \\
 &z_2(t_j) = 0.
 \end{aligned}$$

Пусть $t = t_{\delta_1}$, т. е. обслуживание очередной заявки окончено. Этот момент является особым, так как в этот момент $z(t)$ достигает $Z^{(n)}$, т. е. $z_1(t_{\delta_1}) = 0$. Поэтому скачок состояний $z(t_{\delta_1})$ определяется оператором W^n вида

$$\left. \begin{aligned}
 z_1(t_{\delta_1}+0) &= \varphi(e'_{k+1}, h), \\
 z_2(t_{\delta_1}+0) &= z_2(t_{\delta_1}) - 1, \\
 z_m(t_{\delta_1}+0) &= z_{m+2}(t_{\delta_1}), \\
 z_l(t_{\delta_1}+0) &= z_{l+2}(t_{\delta_1}), \\
 z_1(t_{\delta_1}+0) &= z_1(t_{\delta_1}), \\
 z_2(t_{\delta_1}+0) &= 0,
 \end{aligned} \right\} \begin{aligned}
 &1 \leq k \leq z_3(t_{\delta_1}), \\
 &z_2(t_{\delta_1}) > 0; \\
 &z_2(t_{\delta_1}) = 0.
 \end{aligned}$$

Рассмотрим еще один особый момент времени t_{δ_2} , не являющийся моментом поступления входного сигнала. В момент t_{δ_2} , когда истекает время ожидания одной из заявок, например i -й, число заявок в системе уменьшается на 1. Состояние A -схемы $z_2(t_{\delta_2}+0)$ определяется оператором W^n вида

$$\left. \begin{aligned}
 z_1(t_{\delta_2}+0) &= z_1(t_{\delta_2}), \\
 z_2(t_{\delta_2}+0) &= z_2(t_{\delta_2}) - 1, \\
 z_m(t_{\delta_2}+0) &= z_m(t_{\delta_2}), \\
 z_l(t_{\delta_2}+0) &= z_l(t_{\delta_2}), \\
 z_m(t_{\delta_2}+0) &= z_{m+2}(t_{\delta_2}), \\
 z_l(t_{\delta_2}+0) &= z_{l+2}(t_{\delta_2}),
 \end{aligned} \right\} \begin{aligned}
 &k < i, \\
 &i \leq k \leq z_2(t_{\delta_2}).
 \end{aligned}$$

В полуинтервалах $(t_n, t_{n+1}]$ между особыми моментами времени t_n и t_{n+1} , к которым относятся моменты поступления в A -схему входных сигналов и выдачи выходных сигналов, состояния A -схемы изменяются по закону, задаваемому оператором U , который можно записать так:

$$\begin{aligned}
 z_1(t) &= z_1(t_n+0) - (t - t_n), \\
 z_2(t) &= z_2(t_n+0), \\
 z_m(t) &= z_m(t_n+0), \\
 z_l(t) &= z_l(t_n+0) - (t - t_n).
 \end{aligned}$$

Выходными сигналами A -схемы будем считать сведения о заявках, покидающих прибор (Π). Пусть $y(y^1, y^2)$, где y^1 — признак ($y^1 = 1$, если заявка обслужена; $y^1 = 0$, если заявка не обслужена); y^2 — совокупность сведений о заявке, например $y^2 = (e_p, h, t_{\delta})$, т. е. заявка поступила в систему обслуживания с параметром e_p , обслуживалась при значении параметра системы h , покинула систему в момент t_{δ} . Таким образом, действия оператора G сводятся к выбору признака y^1 и формированию сведений о заявке y^2 . Для моментов t_{δ_1} и t_{δ_2} выходной сигнал y определяется параметром G и может быть записан в следующем виде:

$$y = (1, e_j, h, t_{\delta_1}), y = (0, e_p, h, t_{\delta_2}),$$

где t_{δ_1} находят из $z_1(t_{\delta_1}) = 0$, а t_{δ_2} — из $z_1(t_{\delta_2}) = 0$.

На основании состояний системы S_Q можно оценить ее вероятностно-временные характеристики, например вероятность нахождения в обслуживающем приборе (Π) заданного числа заявок, среднее время ожидания заявок в накопителе (H) и т. д.

Таким образом, дальнейшее использование обобщенной типовой математической схемы моделирования, т. е. A -схемы, в принципе не отличается от рассмотренных ранее D -, F -, P -, N -, Q -схем. Для частного случая, а именно для кусочно-линейных агрегатов, результаты могут быть получены аналитическим методом. В более сложных случаях, когда применение аналитических методов неэффективно или невозможно, прибегают к имитационному методу, причем представление объекта моделирования в виде A -схемы может являться тем фундаментом, на котором базируется построение имитационной системы и ее внешнего и внутреннего математического обеспечения. Стандартная форма представления исследуемого объекта в виде A -схемы приводит к унификации не только алгоритмов имитации, но и к возможности применять стандартные методы обработки и анализа результатов моделирования системы S .

Рассмотренные примеры использования типовых математических схем (D -, F -, P -, Q -, N -, A -схем) позволяют формализовать достаточно широкий класс больших систем, с которыми приходится иметь дело в практике исследования и проектирования сложных систем. Особенности и возможности применения типовых схем при разработке машинных моделей систем рассмотрены в гл. 8 и 10.

Контрольные вопросы

- 2.1. Что называется математической схемой?
- 2.2. Что является экзогенными и эндогенными переменными в модели объекта?
- 2.3. Что называется законом функционирования системы?
- 2.4. Что понимается под алгоритмом функционирования?
- 2.5. Что называется статической и динамической моделями объекта?
- 2.6. Какие типовые схемы используются при моделировании сложных систем и их элементов?
- 2.7. Каковы условия и особенности использования при разработке моделей систем различных типовых схем?

ГЛАВА 3

ФОРМАЛИЗАЦИЯ И АЛГОРИТМИЗАЦИЯ ПРОЦЕССОВ ФУНКЦИОНИРОВАНИЯ СИСТЕМ

Несмотря на многообразие классов моделируемых систем и наличие широких возможностей реализации машинных моделей на современных ЭВМ, можно выделить основные закономерности перехода от построения концептуальной модели объекта моделирования до проведения машинного эксперимента с моделью системы, которые для целей эффективного решения пользователем практических задач моделирования рационально оформить в виде методики разработки и машинной реализации моделей. При этом наиболее существенным фактором, который следует учитывать уже при формализации и алгоритмизации моделей, является использование в качестве инструмента исследования аппаратно-программных средств вычислительной техники. В основе выделения этапов моделирования сложной системы лежит также необходимость привлечения для выполнения этой трудоемкой работы коллективов разработчиков различных специальностей (системщиков, алгоритмистов, программистов).

3.1. МЕТОДИКА РАЗРАБОТКИ И МАШИННОЙ РЕАЛИЗАЦИИ МОДЕЛЕЙ СИСТЕМ

С развитием вычислительной техники наиболее эффективным методом исследования больших систем стало машинное моделирование, без которого невозможно решение многих крупных народнохозяйственных проблем. Поэтому одной из актуальных задач подготовки специалистов является освоение теории и методов математического моделирования с учетом требований системности, позволяющих не только строить модели изучаемых объектов, анализировать их динамику и возможность управления машинным экспериментом с моделью, но и судить в известной мере об адекватности создаваемых моделей исследуемым системам, о границах применимости и правильно организовать моделирование систем на современных средствах вычислительной техники [35, 43, 46].

Методологические аспекты моделирования. Прежде чем рассматривать математические, алгоритмические, программные и прикладные аспекты машинного моделирования, необходимо изучить общие методологические аспекты для широкого класса математических моделей объектов, реализуемых на средствах вычислительной техники. Моделирование с использованием средств вычислительной техники (ЭВМ, АВМ, ГВК) позволяет исследовать механизм явле-

ний, протекающих в реальном объекте с большими или малыми скоростями, когда в натуральных экспериментах с объектом трудно (или невозможно) проследить за изменениями, происходящими в течение короткого времени, или когда получение достоверных результатов сопряжено с длительным экспериментом. При необходимости машинная модель дает возможность как бы «растягивать» или «сжимать» реальное время, так как машинное моделирование связано с понятием системного времени, отличного от реального. Кроме того, с помощью машинного моделирования в диалоговой системе можно обучать персонал АСОИУ принятию решений в управлении объектом, например при организации деловой игры, что позволяет выработать необходимые практические навыки реализации процесса управления [12, 29, 46, 53].

Сущность машинного моделирования системы состоит в проведении на вычислительной машине эксперимента с моделью, которая представляет собой некоторый программный комплекс, описывающий формально и (или) алгоритмически поведение элементов системы S в процессе ее функционирования, т. е. в их взаимодействии друг с другом и внешней средой E . Машинное моделирование с успехом применяют в тех случаях, когда трудно четко сформулировать критерий оценки качества функционирования системы и цель ее не поддается полной формализации, поскольку позволяет сочетать программно-технические возможности ЭВМ со способностями человека мыслить неформальными категориями. В дальнейшем основное внимание будет уделено моделированию систем на универсальных ЭВМ как наиболее эффективному инструменту исследования и разработки АСОИУ различных уровней, а случаи использования АВМ и ГВК будут специально оговариваться.

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

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

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

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

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

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

6. Программные и технические средства должны обеспечивать

эффективную (по быстродействию и памяти) машинную реализацию модели и удобное общение с ней пользователя.

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

С учетом этих требований рассмотрим основные положения, которые справедливы при моделировании на ЭВМ систем S , а также их подсистем и элементов. При машинном моделировании системы S характеристики процесса ее функционирования определяются на основе модели M , построенной исходя из имеющейся исходной информации об объекте моделирования. При получении новой информации об объекте его модель пересматривается и уточняется с учетом новой информации, т. е. процесс моделирования, включая разработку и машинную реализацию модели, является итерационным. Этот итерационный процесс продолжается до тех пор, пока не будет получена модель M , которую можно считать адекватной в рамках решения поставленной задачи исследования и проектирования системы S .

Моделирование систем с помощью ЭВМ можно использовать в следующих случаях [5, 29, 41]: а) для исследования системы S до того, как она спроектирована, с целью определения чувствительности характеристики к изменениям структуры, алгоритмов и параметров объекта моделирования и внешней среды; б) на этапе проектирования системы S для анализа и синтеза различных вариантов системы и выбора среди конкурирующих такого варианта, который удовлетворял бы заданному критерию оценки эффективности системы при принятых ограничениях; в) после завершения проектирования и внедрения системы, т. е. при ее эксплуатации, для получения информации, дополняющей результаты натуральных испытаний (эксплуатации) реальной системы, и для получения прогнозов эволюции (развития) системы во времени.

Существуют общие положения, применяемые ко всем перечисленным случаям машинного моделирования. Даже в тех случаях, когда конкретные способы моделирования отличаются друг от друга и имеются различные модификации моделей, например в области машинной реализации моделирующих алгоритмов с использованием конкретных программно-технических средств, в практике моделирования систем можно сформулировать общие принципы, которые могут быть положены в основу методологии машинного моделирования [29, 35, 46].

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

Взаимосвязь перечисленных этапов моделирования систем и их составляющих (подэтапов) может быть представлена в виде сетево-

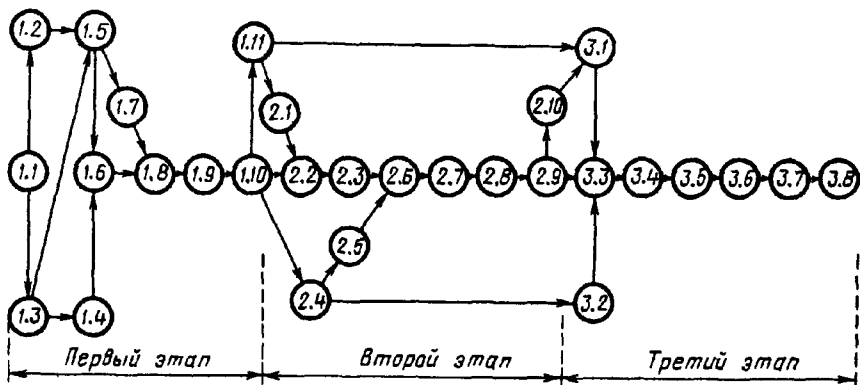


Рис. 3.1. Взаимосвязь этапов моделирования систем

го графика, показанного на рис. 3.1. Перечислим эти подэтапы: 1.1 — постановка задачи машинного моделирования системы; 1.2 — анализ задачи моделирования системы; 1.3 — определение требований к исходной информации об объекте моделирования и организация ее сбора; 1.4 — выдвижение гипотез и принятие предположений; 1.5 — определение параметров и переменных модели; 1.6 — установление основного содержания модели; 1.7 — обновление 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 — планирование машинного эксперимента с моделью системы; 3.2 — определение требований к вычислительным средствам; 3.3 — проведение рабочих расчетов; 3.4 — анализ результатов моделирования системы; 3.5 — представление результатов моделирования; 3.6 — интерпретация результатов моделирования; 3.7 — подведение итогов моделирования и выдача рекомендаций; 3.8 — составление технической документации по третьему этапу.

Таким образом, процесс моделирования системы S сводится к выполнению перечисленных подэтапов, сгруппированных в виде трех этапов. На этапе построения концептуальной модели M_c 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. Рассмотрим содержание каждого из этапов более подробно.

3.2. ПОСТРОЕНИЕ КОНЦЕПТУАЛЬНЫХ МОДЕЛЕЙ СИСТЕМ И ИХ ФОРМАЛИЗАЦИЯ

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

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

Рассмотрим механизм перехода от описания процесса функционирования некоторой гипотетической системы к модели этого процесса [29, 35]. Для наглядности введем представление об описа-

Подэтапы первого этапа моделирования. Рассмотрим более подробно основные подэтапы построения концептуальной модели M_s системы и ее формализации (см. рис. 3.1).

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

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

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

1.3. Определение требований к исходной информации об объекте моделирования и организация ее сбора. После постановки задачи моделирования системы S определяются требования к информации, из которой получают качественные и количественные исходные данные, необходимые для решения этой задачи. Эти данные помогают глубоко разобраться в сущности задачи, методах ее решения. Таким образом, на этом подэтапе проводится: а) выбор необходимой информации о системе S и внешней среде E ; б) подготовка априорных данных; в) анализ имеющихся экспериментальных данных; г) выбор методов и средств предварительной обработки информации о системе.

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

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

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

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

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

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

1.7. Обоснование критериев оценки эффективности системы. Для оценки качества процесса функционирования моделируемой системы S необходимо выбрать некоторую совокупность критериев оценки эффективности, т. е. в математической постановке задача сводится к получению соотношения для оценки эффективности как функции параметров и переменных системы. Эта функция представляет собой поверхность отклика в исследуемой области изменения параметров и переменных и позволяет определить реакцию системы. Эффективность системы S можно оценить с помощью интег-

ральных или частных критериев, выбор которых зависит от рассматриваемой задачи.

1.8. Определение процедур аппроксимации. Для аппроксимации реальных процессов, протекающих в системе S , обычно используются три вида процедур: а) детерминированную; б) вероятностную; в) определения средних значений.

При детерминированной процедуре результаты моделирования однозначно определяются по данной совокупности входных воздействий, параметров и переменных системы S . В этом случае отсутствуют случайные элементы, влияющие на результаты моделирования. Вероятностная (рандомизированная) процедура применяется в том случае, когда случайные элементы, включая воздействия внешней среды E , влияют на характеристики процесса функционирования системы S и когда необходимо получить информацию о законах распределения выходных переменных. Процедура определения средних значений используется тогда, когда при моделировании системы интерес представляют средние значения выходных переменных при наличии случайных элементов.

1.9. Описание концептуальной модели системы. На этом подэтапе построения модели системы: а) описывается концептуальная модель M_x в абстрактных терминах и понятиях; б) дается описание модели с использованием типовых математических схем; в) принимаются окончательно гипотезы и предположения; г) обосновывается выбор процедуры аппроксимации реальных процессов при построении модели. Таким образом, на этом подэтапе проводится подробный анализ задачи, рассматриваются возможные методы ее решения и дается детальное описание концептуальной модели M_x , которая затем используется на втором этапе моделирования.

1.10. Проверка достоверности концептуальной модели. После того как концептуальная модель M_x описана, необходимо проверить достоверность некоторых концепций модели перед тем, как перейти к следующему этапу моделирования системы S . Проверять достоверность концептуальной модели достаточно сложно, так как процесс ее построения является эвристическим и такая модель описывается в абстрактных терминах и понятиях. Один из методов проверки модели M_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.

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

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 .

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

3.3. АЛГОРИТМИЗАЦИЯ МОДЕЛЕЙ СИСТЕМ И ИХ МАШИННАЯ РЕАЛИЗАЦИЯ

На втором этапе моделирования — этапе алгоритмизации модели и ее машинной реализации — математическая модель, сформированная на первом этапе, воплощается в конкретную машинную модель. Этот этап представляет собой этап практической деятельности, направленной на реализацию идей и математических схем в виде машинной модели M_m процесса функционирования системы S . Прежде чем рассматривать подэтапы алгоритмизации и машинной реализации модели, остановимся на основных принципах построения моделирующих алгоритмов и формах их представления [4, 36, 37, 53].

Принципы построения моделирующих алгоритмов. Процесс функционирования системы S можно рассматривать как последовательную смену ее состояний $\vec{z} = z(z_1(t), z_2(t), \dots, z_k(t))$ в k -мерном пространстве. Очевидно, что задачей моделирования процесса функционирования исследуемой системы S является построение функций z , на основе которых можно провести вычисление интересующих характеристик процесса функционирования системы. Для этого должны иметься соотношения, связывающие функции z с переменными, параметрами и временем, а также начальные условия $\vec{z}^0 = z(z_1(t_0), z_2(t_0), \dots, z_k(t_0))$ в момент времени $t = t_0$.

Рассмотрим процесс функционирования некоторой детерминированной системы S_D , в которой отсутствуют случайные факторы, т. е. вектор состояний такой системы можно определить из (2.3) как $\vec{z} = \Phi(\vec{z}^0, \vec{x}, t)$. Тогда состояние процесса в момент времени $t_0 + j\Delta t$

может быть однозначно определено из соотношений математической модели по известным начальным условиям. Это позволяет строить моделирующий алгоритм процесса функционирования системы. Для этого преобразуем соотношения модели Z к такому виду, чтобы сделать удобным вычисление $z_1(t + \Delta t)$, $z_2(t + \Delta t)$, ..., $z_k(t + \Delta t)$ по значениям $z_i(\tau)$, $i = \overline{1, k}$, где $\tau \leq t$. Организуем счетчик системного времени, который в начальный момент показывает время t_0 . Для этого момента $z_i(t_0) = z_i^0$. Прибавим интервал времени Δt , тогда счетчик будет показывать $t_1 = t_0 + \Delta t$. Вычислим значения $z_i(t_0 + \Delta t)$. Затем перейдем к моменту времени $t_2 = t_1 + \Delta t$ и т. д. Если шаг Δt достаточно мал, то таким путем можно получить приближенные значения z .

Рассмотрим процесс функционирования стохастической системы S_R , т. е. системы, на которую оказывают воздействия случайные факторы, т. е. вектор состояний определяется соотношением (2.3). Для такой системы функция состояний процесса z в момент времени $\tau \leq t$ и соотношения модели определяют лишь распределение вероятностей для $z_i(t + \Delta t)$ в момент времени $t + \Delta t$. В общем случае и начальные условия z^0 могут быть случайными, задаваемыми соответствующим распределением вероятностей. При этом структура моделирующего алгоритма для стохастических систем в основном остается прежней. Только вместо состояния $z_i(t + \Delta t)$ теперь необходимо вычислить распределение вероятностей для возможных состояний. Пусть счетчик системного времени показывает время t_0 . В соответствии с заданным распределением вероятностей выбирается z_i^0 . Далее, исходя из распределения, получается состояние $z_i(t_0 + \Delta t)$ и т. д., пока не будет построена одна из возможных реализаций случайного многомерного процесса $z_i(t)$ в заданном интервале времени [9, 37].

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

При рассмотрении процессов функционирования некоторых систем можно обнаружить, что для них характерны два типа состояний: 1) особые, присущие процессу функционирования системы только в некоторые моменты времени (моменты поступления входных или управляющих воздействий, возмущений внешней среды и т. п.); 2) неособые, в которых процесс находится все остальное время. Особые состояния характерны еще и тем обстоятельством, что функции состояний $z_i(t)$ в эти моменты времени изменяются скачком, а между особыми состояниями изменение координат $z_i(t)$ происходит плавно и непрерывно или не происходит совсем. Таким образом, следя при моделировании системы S только за ее особыми

состояниями в те моменты времени, когда эти состояния имеют место, можно получить информацию, необходимую для построения функций $z_i(t)$. Очевидно, для описанного типа систем могут быть построены моделирующие алгоритмы по «принципу особых состояний». Обозначим скачкообразное (релейное) изменение состояния z как δz , а «принцип особых состояний» — как принцип δz .

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

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

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

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

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

Логическая схема моделирующего алгоритма представляет собой логическую структуру модели процесса функционирования системы S . Логическая схема указывает упорядоченную во времени последовательность логических операций, связанных с решением задачи моделирования.

Схема программы отображает порядок программной реализации моделирующего алгоритма с использованием конкретного математического обеспечения. Схема программы представляет собой интерпретацию логической схемы моделирующего алгоритма разработчиком программы на базе конкретного алгоритмического языка. Различие между этими схемами заключается в том, что логическая схема отражает логическую структуру модели процесса функционирования системы, а схема программы — логику машинной реализации модели с использованием конкретных программно-технических средств моделирования.

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

Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения». Некоторые наиболее употребительные в практике моделирования на ЭВМ символы показаны на рис. 3.3, где изображены основные, специфические и специальные символы процесса. К ним относятся: основной символ: *a* — процесс — символ отображает функцию обработки данных любого вида (выполнение определенной операции или группы операций, приводящее к изменению значения, формы или размещения информации или к определению, по которому из нескольких направлений потока следует двигаться); специфические символы процесса: *b* — решение — символ отображает решение или функцию переключательного типа, имеющую один вход и ряд альтернативных выходов, один и только один из которых может быть активизирован после вычисления условий, определенных внутри этого символа (соответствующие результаты вычисления могут быть записаны по соседству с линиями, отображающими эти пути); *v* — подготовка — символ отображает модификацию команды или группы команд с целью воздействия на некоторую последующую функцию (установка переключателя, модификация индексного регистра или инициализация программы); *z* — предопределенный процесс — символ отображает предопределенный процесс, состоящий из одной или нескольких операций или шагов программы, которые определены в другом месте (в подпрограмме, модуле); *d* — ручная операция — символ отображает любой процесс, выполняемый человеком; специальные символы: *e* — соединитель — символ отображает выход в часть схемы и вход из другой части этой схемы и используется для обрыва линии и продолжения ее в другом месте (соответствующие символы-соединители должны содержать одно и то же уникальное обозначение); *ж* — терминатор — символ отображает выход во внешнюю среду и вход из внешней среды (начало или конец схемы алгоритма, внешнее использование или пункт назначения данных).

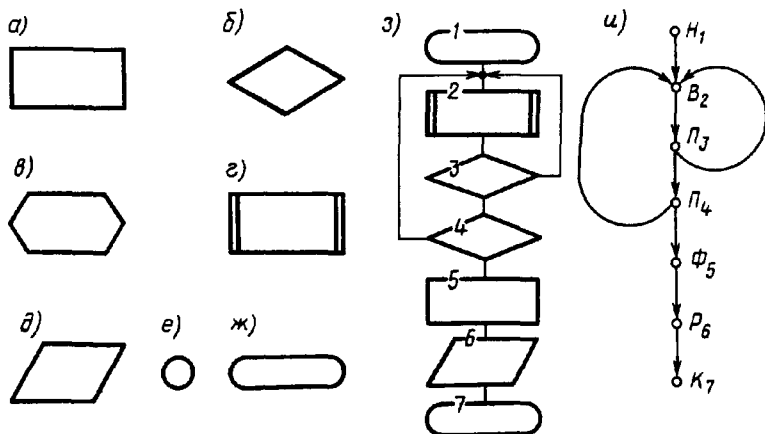


Рис. 3.3. Символы и схемы моделирующих алгоритмов

Пример изображения схемы моделирующего алгоритма показан на рис. 3.3, з.

Обычно схема является наиболее удобной формой представления структуры моделирующих алгоритмов. В ряде случаев используются и другие формы представления моделирующих алгоритмов, например форма *граф-схем* (рис. 3.3, и). Здесь H_i — начало, K_i — конец, B_i — вычисление, Φ_i — формирование, Π_i — проверка условия, C_i — счетчик, P_i — выдача результата, $i = \overline{1, g}$, где g — общее число операторов моделирующего алгоритма. В качестве пояснения к граф-схеме алгоритма в тексте дается раскрытие содержания операторов, что позволяет упростить представление алгоритма, но усложняет работу с ним.

Моделирующие алгоритмы могут быть также представлены в виде операторных схем [4]. Обозначения операторов на такой схеме соответствуют обозначениям для граф-схем. Для рассмотренного примера операторная схема алгоритма имеет вид

$$H_1 \overset{3}{\rightarrow} B_2 \overset{4}{\rightarrow} \Pi_3 \overset{4}{\rightarrow} \Pi_4 \overset{5}{\rightarrow} \Phi_5 \overset{6}{\rightarrow} P_6 \overset{7}{\rightarrow} K_7.$$

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

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

2.1. Построение логической схемы модели. Рекомендуется строить модель по блочному принципу, т. е. в виде некоторой совокупности стандартных блоков. Построение модели систем S из таких

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

2.2. Получение математических соотношений. Одновременно с выполнением подэтапа построения логической схемы модели необходимо получить, если это возможно, математические соотношения в виде явных функций, т. е. построить аналитические модели. Этот подэтап соответствует неявному заданию возможных математических соотношений на этапе построения концептуальной модели. При выполнении первого этапа еще не может иметься информации о конкретном виде таких математических соотношений, а на втором этапе уже необходимо получить эти соотношения. Схема машинной модели M_m 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 процесса функционирования системы; г) задание математических соотношений в явном виде.

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

2.3. Проверка достоверности модели системы. Эта проверка является первой из проверок, выполняемых на этапе реализации модели. Так как модель представляет собой приближенное описание процесса функционирования реальной системы S , то до тех пор, пока не доказана достоверность модели M_m , нельзя утверждать, что с ее помощью будут получены результаты, совпадающие с теми, которые могли бы быть получены при проведении натурального эксперимента с реальной системой S . Поэтому определение достоверности модели можно считать наиболее важной проблемой при моделировании систем. От решения этой проблемы зависит степень доверия к результатам, полученным методом моделирования. Проверка модели на рассматриваемом подэтапе должна дать ответ на вопрос, насколько логическая схема модели системы и используемые математические соотношения отражают замысел модели, сформированный на первом этапе. При этом проверяются: а) возможность

решения поставленной задачи; б) точность отражения замысла в логической схеме; в) полнота логической схемы модели; г) правильность используемых математических соотношений.

Только после того, как разработчик убеждается путем соответствующей проверки в правильности всех этих положений, можно считать, что имеется логическая схема модели системы S , пригодная для дальнейшей работы по реализации модели на ЭВМ.

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

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

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

Построение схемы программы представляет собой одну из основных задач на этапе машинной реализации модели. При этом особое внимание должно быть уделено особенностям выбранного для реализации модели языка: алгоритмического языка общего

назначения или языка моделирования (например, *SIMULA*, *SIMSCRIPT*, *GPSS*).

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

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

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

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

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

Таким образом, на этом этапе разрабатывается схема модели системы *S*, проводится ее алгоритмизация и программирование

с использованием конкретных программно-технических средств, т. е. строится машинная модель M_m , с которой предстоит работать для получения необходимых результатов моделирования по оценке характеристик процесса функционирования системы S (задача анализа) или для поиска оптимальных структур, алгоритмов и параметров системы S (задача синтеза).

3.4. ПОЛУЧЕНИЕ И ИНТЕРПРЕТАЦИЯ РЕЗУЛЬТАТОВ МОДЕЛИРОВАНИЯ СИСТЕМ

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

Особенности получения результатов моделирования. При реализации моделирующих алгоритмов на ЭВМ вырабатывается информация о состояниях процесса функционирования исследуемых систем $z(t) \in Z$. Эта информация является исходным материалом для определения приближенных оценок искомых характеристик, получаемых в результате машинного эксперимента, т. е. критериев оценки. *Критерием оценки* будем называть любой количественный показатель, по которому можно судить о результатах моделирования системы. Критериями оценки могут служить показатели, получаемые на основе процессов, действительно протекающих в системе, или получаемых на основе специально сформированных функций этих процессов [4, 29, 35].

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

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

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

Рассмотрим общую схему фиксации и обработки результатов моделирования системы, которая приведена на рис. 3.4. Будем рассматривать гипотетическую модель M , предназначенную для исследования поведения системы S на интервале времени $[0, T]$. В общем случае критерием интерпретации результатов моделирова-

ния является нестационарный случайный n -мерный процесс $\vec{q}(t)$, $0 \leq t \leq T$. Полагаем для определенности, что состояние моделируемой системы S проверяется каждые Δt временных единиц, т. е. используется «принцип Δt ». При этом вычисляют значения $\vec{q}(j\Delta t)$, $j=0, \overline{k}$, критерия $\vec{q}(t)$. Таким образом, о свойствах случайного процесса $\vec{q}(t)$ судят по свойствам случайной последовательности $\vec{q}(j\Delta t)$, $j=0, \overline{k}$, или, иначе говоря, по свойствам m -мерного вектора вида

$$\vec{q} = (\vec{q}(0), \vec{q}(\Delta t), \dots, \vec{q}[(k-1)\Delta t], \vec{q}(T)), m = n(k+1), T = k\Delta t.$$

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

На схеме, изображенной на рис. 3.4, обозначено: $I \equiv i$; $J \equiv j$; $K \equiv k$; $N \equiv N$; $T \equiv t$; $DT \equiv \Delta t$; $Q \equiv q$.

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

Внутренний цикл (блоки 5 — 8) позволяет получить последовательность $\vec{q}_i(t) = \vec{q}(j\Delta t)$, $j=0, \overline{k}$ в моменты времени $t=0, \Delta t, 2\Delta t, \dots, k\Delta t = T$. Основной блок 7 реализует процедуру вычисления последовательности $\vec{q}_i(t)$: ВВЧ $[QI(T)]$. Именно в этом блоке имитируется процесс функционирования моделируемой системы S на интервале времени $[0, T]$.

Промежуточный цикл (блоки 3 — 10), в котором организуется N -кратное повторение прогона модели, позволяющее после соответствующей статистической обработки результатов судить об оценках характеристик моделируемого варианта системы. Окончание моделирования варианта системы S может определяться не только заданным числом реализаций (блок 10), как это показано на схеме, но и заданной точностью результатов моделирования. В этом цикле содержится блок 9, реализующий процедуру фиксации

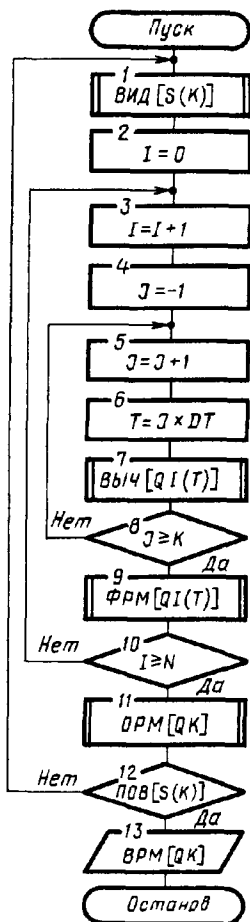


Рис. 3.4. Алгоритм фиксации и обработки результатов моделирования системы

результатов моделирования по i -му прогону модели $\vec{q}_i(t)$: ФРМ $[QI(T)]$.

Внешний цикл (блоки 1 — 12) охватывает оба предшествующих цикла и дополнительно включает блоки 1, 2, 11, 12, управляющие последовательностью моделирования вариантов системы S . Здесь организуется поиск оптимальных структур, алгоритмов и параметров системы S , т. е. блок 11 обрабатывает результаты моделирования исследуемого k -го варианта системы ОРМ $[QK]$, блок 12 проверяет удовлетворительность полученных оценок характеристик процесса функционирования системы $\vec{q}_i^{(k)}(t)$ требуемым (ведет поиск оптимального варианта системы ПОВ $[S(K)]$), блок 1 изменяет структуру, алгоритмы и параметры системы S на уровне ввода исходных данных для очередного k -го варианта системы ВИД $[S(K)]$. Блок 13 реализует функцию выдачи результатов моделирования по каждому k -му варианту модели системы S_k , т. е. ВРМ $[QK]$.

Рассмотренная схема позволяет вести статистическую обработку результатов моделирования в наиболее общем случае при нестационарном критерии $\vec{q}(t)$. В частных случаях можно ограничиться более простыми схемами [22, 29, 37].

Если свойства моделируемой системы S определяются значением критерия $\vec{q}(t)$ в некоторый заданный момент времени, например в конце периода функционирования модели $t = k\Delta t = T$, то обработка сводится к оценке распределения n -мерного вектора $\vec{q}(t)$ по независимым реализациям $\vec{q}_i(t)$, $i = \overline{1, N}$, полученным в результате N прогонов модели.

Если в моделируемой системе S по истечению некоторого времени с начала работы $t_0 = k_0\Delta t$ установится стационарный режим, то о нем можно судить по одной, достаточно длинной реализации $\vec{q}_1(t)$ критерия $\vec{q}(t)$, стационарного и эргодического на интервале $[t_0, T]$. Для рассмотренной схемы это означает, что исключается средний цикл ($n = 1$) и добавляется оператор, позволяющий начать обработку значений $q_1(j\Delta t)$ при $j \geq k_0$.

Другая особенность применяемых на практике методов статистической обработки результатов моделирования связана с исследованием процесса функционирования систем с помощью моделей блочной конструкции. В этом случае часто приходится применять раздельное моделирование отдельных блоков модели, когда имитация входных воздействий для одного блока проводится на основе оценок критериев, полученных предварительно на другом блоке модели. При раздельном моделировании может иметь место либо непосредственная запись в накопителе реализаций критериев, либо их аппроксимация, полученная на основе статистической обработки

результатов моделирования с последующим использованием генераторов случайных чисел для имитации этих воздействий.

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

3.1. Планирование машинного эксперимента с моделью системы. Перед выполнением рабочих расчетов на ЭВМ должен быть составлен план проведения эксперимента с указанием комбинаций переменных и параметров, для которых должно проводиться моделирование системы S . Планирование машинного эксперимента призвано дать в итоге максимальный объем необходимой информации об объекте моделирования при минимальных затратах машинных ресурсов. При этом различают стратегическое и тактическое планирование машинного эксперимента. При стратегическом планировании эксперимента ставится задача построения оптимального плана эксперимента для достижения цели, поставленной перед моделированием (например, оптимизация структуры, алгоритмов и параметров системы S , исследуемой методом моделирования на ЭВМ). Тактическое планирование машинного эксперимента преследует частные цели оптимальной реализации каждого конкретного эксперимента из множества необходимых, заданных при стратегическом планировании (например, решение задачи выбора оптимальных правил остановки при статистическом моделировании системы S на ЭВМ). Для получения наиболее эффективного плана машинного эксперимента необходимо использовать статистические методы [10, 18, 21].

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_n и определения чувствительности результатов к изменению исходных данных.

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

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

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

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

3.8. Составление технической документации по третьему этапу. Эта документация должна включать в себя: а) план проведения

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

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

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

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

Контрольные вопросы

- 3.1. В чем суть методики машинного моделирования систем?
- 3.2. Какие требования пользователь предъявляет к машинной модели системы?
- 3.3. Что называется концептуальной моделью системы?
- 3.4. Какие группы блоков выделяются при построении блочной конструкции модели системы?
- 3.5. Каковы основные принципы построения моделирующих алгоритмов процессов функционирования систем?
- 3.6. Какие схемы используются при разработке алгоритмического и программного обеспечения машинного моделирования?
- 3.7. Какие циклы можно выделить в моделирующем алгоритме?
- 3.8. Что называется прогоном модели?
- 3.9. Какая техническая документация оформляется по каждому этапу моделирования системы?

ГЛАВА 4

СТАТИСТИЧЕСКОЕ МОДЕЛИРОВАНИЕ СИСТЕМ НА ЭВМ

В практике моделирования систем информатики наиболее часто приходится иметь дело с объектами, которые в процессе своего функционирования содержат элементы стохастичности или подвергаются стохастическим воздействиям внешней среды. Поэтому основным методом получения результатов с помощью имитационных моделей таких стохастических систем является метод статистического моделирования на ЭВМ, использующий в качестве теоретической базы предельные теоремы теории вероятностей. Возможность получения пользователем модели результатов статистического моделирования сложных систем в условиях ограниченности машинных ресурсов существенно зависит от эффективности процедур генерации псевдослучайных последовательностей на ЭВМ, положенных в основу имитации воздействий на элементы моделируемой системы.

4.1. ОБЩАЯ ХАРАКТЕРИСТИКА МЕТОДА СТАТИСТИЧЕСКОГО МОДЕЛИРОВАНИЯ

На этапе исследования и проектирования систем при построении и реализации машинных моделей (аналитических и имитационных) широко используется метод статистических испытаний (Монте-Карло), который базируется на использовании случайных чисел, т. е. возможных значений некоторой случайной величины с заданным распределением вероятностей. Статистическое моделирование представляет собой метод получения с помощью ЭВМ статистических данных о процессах, происходящих в моделируемой системе. Для получения представляющих интерес оценок характеристик моделируемой системы S с учетом воздействий внешней среды E статистические данные обрабатываются и классифицируются с использованием методов математической статистики [10, 13, 18].

Сущность метода статистического моделирования. Таким образом, 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.

Различают две области применения метода статистического моделирования: 1) для изучения стохастических систем; 2) для решения детерминированных задач. Основной идеей, которая используется для решения детерминированных задач методом статистического моделирования, является замена детерминированной задачи эквивалентной схемой некоторой стохастической системы, выходные характеристики последней совпадают с результатом решения детерминированной задачи. Естественно, что при такой замене вместо точного решения задачи получается приближенное решение и погрешность уменьшается с увеличением числа испытаний (реализаций моделирующего алгоритма) N .

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 .

Теоретической основой метода статистического моделирования систем на ЭВМ являются *предельные теоремы теории вероятностей* [2, 13]. Множества случайных явлений (событий, величин) подчиняются определенным закономерностям, позволяющим не только прогнозировать их поведение, но и количественно оценить некоторые средние их характеристики, проявляющие определенную устойчивость. Характерные закономерности наблюдаются также в распределениях случайных величин, которые образуются при сложении множества воздействий. Выражением этих закономерностей и устойчивости средних показателей являются так называемые предельные теоремы теории вероятностей, часть из которых приводится ниже в пригодной для практического использования при статистическом моделировании формулировке. Принципиальное значение предельных теорем состоит в том, что они гарантируют высокое качество статистических оценок при весьма большом числе испытаний (реализаций) N . Практически приемлемые при статистическом моделировании количественные оценки характеристик систем часто могут быть получены уже при сравнительно небольших (при использовании ЭВМ) N .

Неравенство Чебышева. Для неотрицательной функции $g(\xi)$ случайной величины ξ и любого $K > 0$ выполняется неравенство

$$P\{g(\xi) \geq K\} \leq M[g(\xi)]/K. \quad (4.1)$$

В частности, если $g(\xi) = (\xi - \bar{x})^2$ и $K = k^2 \sigma^2$ (где \bar{x} — среднее арифметическое; σ — среднее квадратическое отклонение), то

$$P\{|\xi - \bar{x}| \geq k\sigma\} \leq 1/k^2. \quad (4.2)$$

Теорема Бернулли. Если проводится N независимых испытаний, в каждом из которых некоторое событие A осуществляется с вероятностью p , то относительная частота появления события m/N при $N \rightarrow \infty$ сходится по вероятности к p , т. е. при любом $\varepsilon > 0$

$$\lim_{N \rightarrow \infty} P\{|m/N - p| \geq \varepsilon\} = 0, \quad (4.3)$$

где m — число положительных исходов испытания.

Теорема Пуассона. Если проводится N независимых испытаний и вероятность осуществления события A в i -м испытании равна p_i , то относительная частота появления события m/N при $N \rightarrow \infty$ сходится по вероятности к среднему из вероятностей p_i , т. е. при любом $\varepsilon > 0$

$$\lim_{N \rightarrow \infty} P\left\{\left|m/N - \frac{1}{N} \sum_{i=1}^N p_i\right| \geq \varepsilon\right\} = 0. \quad (4.4)$$

Теорема Чебышева. Если в N независимых испытаниях наблюдаются значения x_1, \dots, x_N случайной величины ξ , то при $N \rightarrow \infty$ среднее арифметическое значений случайной величины сходится по вероятности к ее математическому ожиданию a , т. е. при любом $\varepsilon > 0$

$$\lim_{N \rightarrow \infty} P\left\{\left|\frac{1}{N} \sum_{i=1}^N x_i - a\right| \geq \varepsilon\right\} = 0. \quad (4.5)$$

Обобщенная теорема Чебышева. Если ξ_1, \dots, ξ_N — независимые случайные величины с математическими ожиданиями a_1, \dots, a_N и дисперсиями $\sigma_1^2, \dots, \sigma_N^2$, ограниченными сверху одним и тем же числом, то при $N \rightarrow \infty$ среднее арифметическое значений случайной величины сходится по вероятности к среднему арифметическому их математических ожиданий:

$$\lim_{N \rightarrow \infty} P\left\{\left|\frac{1}{N} \sum_{i=1}^N x_i - \frac{1}{N} \sum_{i=1}^N a_i\right| \geq \varepsilon\right\} = 0. \quad (4.6)$$

Теорема Маркова. Выражение (4.6) справедливо и для зависимых случайных величин ξ_1, \dots, ξ_N , если только

$$\lim_{N \rightarrow \infty} \frac{1}{N^2} D\left[\sum_{i=1}^N x_i\right] = 0.$$

Совокупность теорем, устанавливающих устойчивость средних показателей, принято называть *законом больших чисел*.

Центральная предельная теорема. Если ξ_1, \dots, ξ_N — независимые одинаково распределенные случайные величины, имеющие математическое ожидание a и дисперсию σ^2 , то при $N \rightarrow \infty$ закон распределения суммы $\sum_{i=1}^N x_i$ неограниченно приближается к нормальному:

$$\lim_{N \rightarrow \infty} P\left\{\alpha < \left(\sum_{i=1}^N x_i - Na\right) / \sqrt{N} \sigma < \beta\right\} = \frac{1}{\sqrt{2\pi}} \int_{\alpha}^{\beta} e^{-t^2/2} dt = \Phi_0(\beta) - \Phi_0(\alpha). \quad (4.7)$$

Здесь интеграл вероятностей

$$\Phi_0(\gamma) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\gamma} e^{-t^2/2} dt$$

Теорема Лапласа. Если в каждом из N независимых испытаний событие A появляется с вероятностью p , то

$$\lim_{N \rightarrow \infty} P\{\alpha < (m - Np) / \sqrt{Np(1-p)} < \beta\} = \Phi_0(\beta) - \Phi_0(\alpha), \quad (4.8)$$

где m — число появлений события A в N испытаниях. Теорема Лапласа является частным случаем центральной предельной теоремы.

Примеры статистического моделирования. Статистическое моделирование систем на ЭВМ требует формирования значений случайных величин, что реализуется с помощью датчиков (генераторов) случайных чисел. Не останавливаясь пока на способах их реализации для целей моделирования на ЭВМ, поясним сущность метода статистического моделирования следующими примерами.

Пример 4.1. Необходимо методом статистического моделирования найти оценки выходных характеристик некоторой стохастической системы S_R , функционирование которой описывается следующими соотношениями: $x = 1 - e^{-\lambda}$ — входное воздействие, $v = 1 - e^{-\varphi}$ — воздействие внешней среды, где λ и φ — случайные величины, для которых известны их функции распределения. Целью моделирования является оценка математического ожидания $M[y]$ величины y . Зависимость последней от входного воздействия x и воздействия внешней среды

v имеет вид $y = \sqrt{x^2 + v^2}$.

В качестве оценки математического ожидания $M[y]$, как следует из приведенных теорем теории вероятностей, может выступать среднее арифметическое, вычисленное по формуле

$$\bar{y} = \frac{1}{N} \sum_{i=1}^N y_i,$$

где y_i — случайное значение величины y ; N — число реализаций, необходимое для статистической устойчивости результатов.

Структурная схема системы S_R показана на рис. 4.1.

Здесь элементы выполняют следующие функции:

вычисление V_1 :

$$x_i = 1 - e^{-\lambda_i} \text{ и } V_2: v_i = 1 - e^{-\varphi_i},$$

возведение в квадрат K_1 :

$$h'_i = (1 - e^{-\lambda_i})^2; K_2: h''_i = (1 - e^{-\varphi_i})^2;$$

суммирование C :

$$h_i = (1 - e^{-\lambda_i})^2 + (1 - e^{-\varphi_i})^2;$$

извлечение квадратного корня I :

$$y_i = \sqrt{(1 - e^{-\lambda_i})^2 + (1 - e^{-\varphi_i})^2}.$$

Схема алгоритма, реализующего метод статистического моделирования для оценки $M[y]$ системы S_R , приведена на рис. 4.2. Здесь LA и FI — функции распределе-

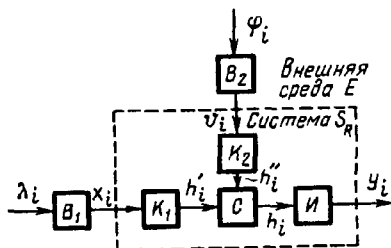


Рис. 4.1. Структурная схема системы S_R

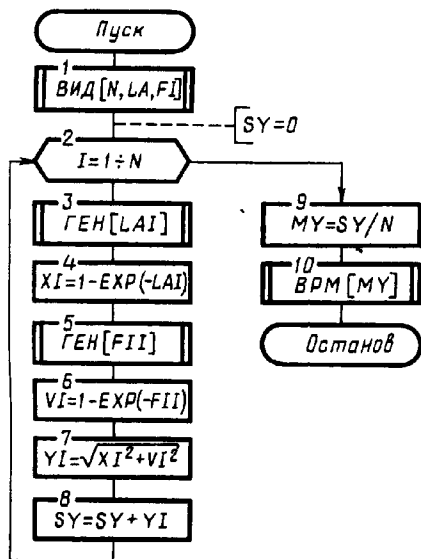


Рис. 4.2. Схема моделирующего алгоритма системы S_R

Таким образом, данная задача является чисто детерминированной и ее аналитическое решение сводится к вычислению определенного интеграла, т. е. искомая площадь фигуры

$$S_{\Phi} = \int_0^{\alpha_1} f(\alpha) d\alpha.$$

Для решения этой детерминированной задачи методом статистического моделирования необходимо предварительно построить адекватную по выходным характеристикам стохастическую систему S_D , оценки характеристик которой будут совпадать с искомыми в данной детерминированной задаче. Вариант структурной схемы такой системы S_D показан на рис. 4.4, где элементы выполняют следующие функции:

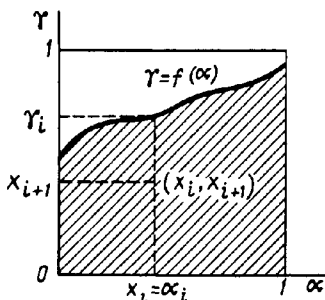


Рис. 4.3 Геометрическая интерпретация оценки площади фигуры

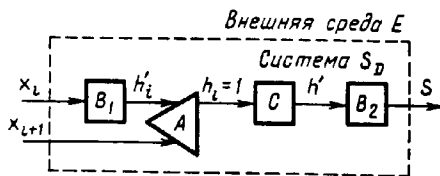


Рис. 4.4 Структурная схема системы S_D

ния случайных величин λ и φ ; N — заданное число реализаций; $I=i$ — номер текущей реализации; $LAI \equiv \lambda$; $FII \equiv \varphi$;

$EXP \equiv e$; $MY \equiv M[y]$, $SY \equiv \sum_{i=1}^N y_i$ — суммирующая ячейка; $ВИД[...]$, $ГЕН[...]$,

$ВРМ[...]$ — процедуры ввода исходных данных, генерации псевдослучайных последовательностей и выдачи результатов моделирования соответственно.

Таким образом, данная модель позволяет получить методом статистического моделирования на ЭВМ статистическую оценку математического ожидания выходной характеристики системы S_R . Точность и достоверность результатов взаимодействия в основном будут определяться числом реализаций N .

Пример 4.2. Необходимо методом статистического моделирования найти оценку площади фигуры (рис. 4.3), ограниченной осями координат, ординатой $\alpha=1$ и кривой $y=f(\alpha)$; при этом для определенности предполагается, что $0 \leq f(\alpha) \leq 1$ для всех α , $0 \leq \alpha \leq 1$.

вычисление B_1 : $h'_i = f(x_i)$;

анализ А: $h = \begin{cases} 1, & \text{если } x_{i+1} \leq f(x_i), \\ 0 & \text{в противном случае,} \end{cases}$

суммирование С: $h' = \sum_{i=1}^N h_i$;

вычисление B_2 : $S = h'/N$.

Система S_D функционирует следующим образом: получается пара независимых случайных чисел интервала (0, 1), определяется координата точки (x_i, x_{i+1}) , показанной на рис. 4.3, вычисляется ордината $y_i = f(x_i)$ и проводится сравнение величин y_i и x_{i+1} ; причем если точка (x_i, x_{i+1}) попала в площадь фигуры (в том числе и на кривую $f(x)$), то исход испытания считается положительным $h_i = 1$ и в итоге можно получить статистическую оценку площади фигуры S_D по заданному числу реализаций N .

Логическая схема моделирующего алгоритма вероятностной системы S_D представлена на рис. 4.5. Здесь $Y \equiv y = f(x)$ — заданная функция (табличная кривая); N — заданное число реализаций; $I \equiv i$ — номер текущей реализации; $XI \equiv x_i$;

$XI \equiv x_{i+1}$; $HI \equiv h_i$; $S \equiv s$; $SH \equiv h' = \sum_{i=1}^N h_i$ —

суммирующая ячейка.

Таким образом, построение некоторой стохастической системы S_D позволяет методом статистического моделирования получать оценки для детерминированной задачи.

Пример 4.3. Необходимо методом статистического моделирования решить следующую задачу. Проводится $s = 10$ независимых выстрелов по мишени, причем вероятность попадания при одном выстреле задана и равна p . Требуется оценить вероятность того, что число попаданий в мишень будет четным, т. е. 0, 2, 4, 6, 8, 10.

Данная задача является вероятностной, причем существует ее аналитическое решение:

$$P = \sum_{k=0}^5 C_{10}^{2k} p^{2k} (1-p)^{10-2k}.$$

В качестве объекта статистического моделирования можно рассмотреть следующую вероятностную систему S_P , структура которой представлена на рис. 4.6, где элементы выполняют такие функции:

анализ A_1 :

$h_i = \begin{cases} 1, & \text{если } x_i < p, \\ 0 & \text{в противном случае;} \end{cases}$

суммирование С: $h_j = \sum_{i=1}^{10} h_i, j = \overline{1, N}$;

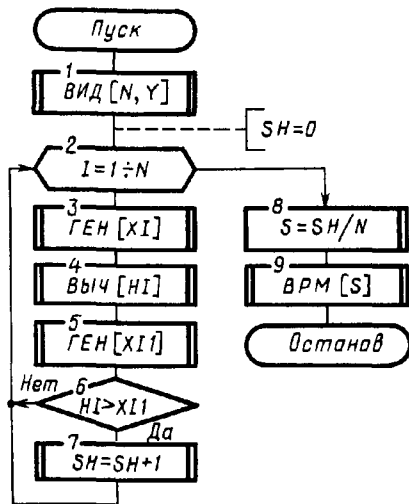


Рис. 4.5. Схема моделирующего алгоритма системы S_D

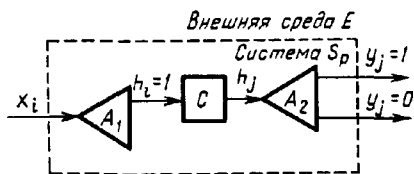


Рис. 4.6. Структурная схема системы S_P

анализ A_2 : $y_j = \begin{cases} 1, & \text{если } h_j \text{ — четное;} \\ 0 & \text{в противном случае.} \end{cases}$

Выходным воздействием в данной системе S_p является событие четного числа попаданий в мишень в серии из десяти выстрелов. В качестве оценки выходной характеристики необходимо при числе испытаний (серий выстрелов), равном N , найти вероятность четного числа попаданий:

$$P(y) = \frac{1}{N} \sum_{j=1}^N y_j$$

Логическая схема алгоритма статистического моделирования для оценки искомой характеристики такой системы $P(y)$ приведена на рис. 4.7. Здесь $P \equiv p$ — заданная вероятность попадания в мишень при одном выстреле; N — заданное число реализаций;

$XI \equiv x_i$; $HJ \equiv h_j = \sum_{i=1}^{10} h_i$; $PY \equiv P(y)$;
 $SY \equiv \sum_{j=1}^{10} y_j$ — суммирующая ячейка.

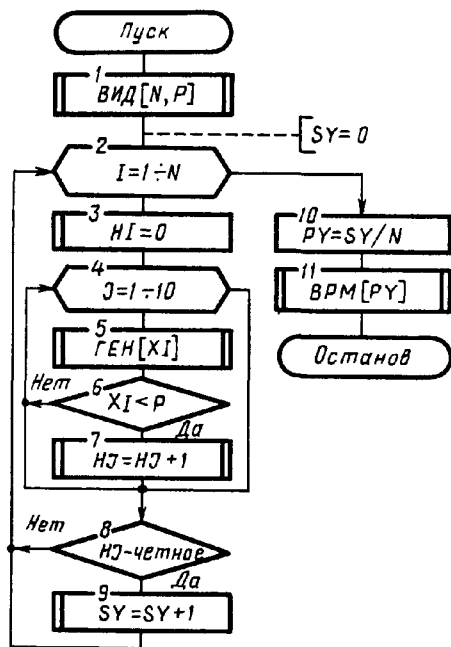


Рис. 4.7. Схема моделирующего алгоритма системы S_p

падание в мишень», а в противном случае — «промах». Далее моделируются серии из десяти испытаний каждая, подсчитывается четное число «попаданий» в каждой серии и находится статистическая оценка искомой характеристики $P(y)$.

Таким образом, подход при использовании статистического моделирования независим от природы объекта исследования (будет ли он детерминированным или стохастическим) является общим, причем при статистическом моделировании детерминированных систем (система S_D в примере 4.2) необходимо предварительно построить стохастическую систему, выходные характеристики которой позволяют оценить искомые.

Отметим, что во всех рассмотренных примерах не требуется запоминания всего множества генерируемых случайных чисел, используемых при статистическом моделировании системы S . Запоминается только накопленная сумма исходов и общее число ре-

ализаций. Это немаловажное обстоятельство вообще является характерным при реализации имитационных моделей методом статистического моделирования на ЭВМ.

4.2. ПСЕВДОСЛУЧАЙНЫЕ ПОСЛЕДОВАТЕЛЬНОСТИ 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 расходуются 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 [31, 37, 46].

Рассмотрим возможности и особенности получения последовательностей случайных чисел при статистическом моделировании систем на ЭВМ. На практике используются три основных способа генерации случайных чисел: аппаратный (физический), табличный (файловый) и алгоритмический (программный).

Аппаратный способ. При этом способе генерации случайные числа вырабатываются специальной электронной приставкой — генератором (датчиком) случайных чисел, — служащей в качестве одного из внешних устройств ЭВМ. Таким образом, реализация этого способа генерации не требует дополнительных вычислительных операций ЭВМ по выработке случайных чисел, а необходима только операция обращения к внешнему устройству (датчику). В качестве физического эффекта, лежащего в основе таких генераторов чисел, чаще всего используются шумы в электронных и полупроводниковых приборах, явления распада радиоактивных элементов и т. д. Рассмотрим принцип получения случайных чисел от приставки, основанный, например, на эффекте шума в полупроводниковых приборах.

Структурная схема аппаратного генератора случайных чисел приведена на рис. 4.8, а. Здесь *ИШ* — источник шума; *КС* — ключевая схема; *ФИ* — формирователь импульсов; *ПС* — пересчетная схема. При усилении шумов на выходе *ИШ* получается напряжение

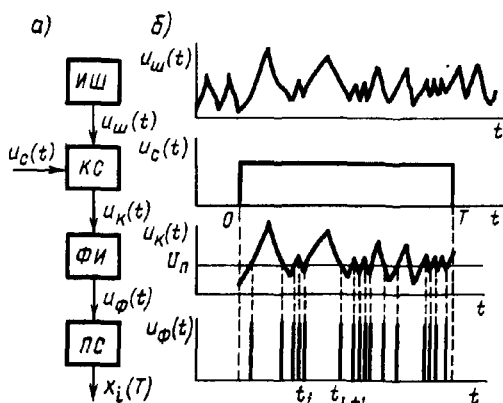


Рис. 4.8. Аппаратный способ получения случайных чисел

если провести масштабирование и принять длину интервала $(0, T)$ за единицу, то значения интервалов времени $\Delta t_i = t_{i+1} - t_i$ между соседними импульсами $u_\phi(t)$ будут случайными числами $x_i \in (0, 1)$. Возможны и другие схемные решения аппаратных генераторов случайных чисел [29, 37]. Однако аппаратный способ получения случайных чисел не позволяет гарантировать качество последовательности непосредственно во время моделирования системы S на ЭВМ, а также повторно получать при моделировании одинаковые последовательности чисел.

Табличный способ. Если случайные числа, оформленные в виде таблицы, помещать во внешнюю или оперативную память ЭВМ, предварительно сформировав из них соответствующий файл (массив чисел), то такой способ будет называться табличным. Однако этот способ получения случайных чисел при моделировании систем на ЭВМ обычно рационально использовать при сравнительно небольшом объеме таблицы и соответственно файла чисел, когда для хранения можно применять оперативную память. Хранение файла во внешней памяти при частом обращении в процессе статистического моделирования не рационально, так как вызывает увеличение затрат машинного времени при моделировании системы S из-за необходимости обращения к внешнему накопителю. Возможны промежуточные способы организации файла, когда он переписывается в оперативную память периодически по частям. Это уменьшает время на обращение к внешней памяти, но сокращает объем оперативной памяти, который можно использовать для моделирования процесса функционирования системы S .

Алгоритмический способ. Способ получения последовательностей случайных чисел основан на формировании случайных чисел в ЭВМ с помощью специальных алгоритмов и реализующих их программ.

$u_w(t)$, которое является случайным процессом, показанным на временной диаграмме рис. 4.8, б. Причем отрезок шумовой реализации $u_x(t)$, сформированный на интервале времени $(0, T)$ с помощью КС, содержит случайное число выбросов. Сравнение напряжения $u_x(t)$ с пороговым U_n позволяет сформировать на выходе ФИ серию импульсов $u_\phi(t)$. Тогда на выходе ПС может быть получена последовательность случайных чисел $x_i(t)$. Например,

Каждое случайное число вычисляется с помощью соответствующей программы по мере возникновения потребностей при моделировании системы на ЭВМ.

Таблица 4.1

Способ	Достоинства	Недостатки
Аппаратный	Запас чисел не ограничен Расходуется мало операций вычислительной машины Не занимает место в памяти машины	Требуется периодическая проверка Нельзя воспроизводить последовательности Используется специальное устройство Необходимы меры по обеспечению стабильности
Табличный	Требуется однократная проверка Можно воспроизводить последовательности	Запас чисел ограничен Занимает много места в оперативной памяти или необходимо время на обращение к внешней памяти
Алгоритмический	Требуется однократная проверка Можно многократно воспроизводить последовательности чисел Занимает мало места в памяти машины Не используются внешние устройства	Запас чисел последовательно ограничен ее периодом Существенные затраты машинного времени

Достоинства и недостатки трех перечисленных способов получения случайных чисел для сравнения представлены в табл. 4.1. Из этой таблицы видно, что алгоритмический способ получения случайных чисел наиболее рационален на практике при моделировании систем на универсальных ЭВМ.

Генерация базовой последовательности. При моделировании систем на 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 [29, 37, 46].

Непрерывная случайная величина ξ имеет равномерное распределение в интервале (a, b) , если ее функции плотности

(рис. 4.9, а) и распределения (рис. 4.9, б) соответственно примут вид

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

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

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

$$M|\xi| = \int_a^b x f(x) dx = \int_a^b x dx / (b-a) = (a+b)/2;$$

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

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

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

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

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 *п*-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*.

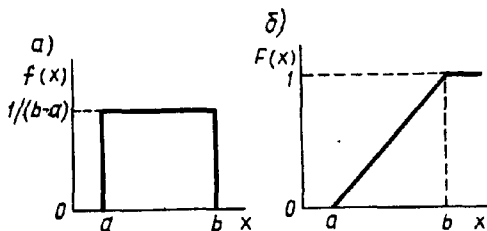


Рис. 4.9. Равномерное распределение случайной величины

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

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

$$D[\xi] = \sum_{i=0}^{2^n-1} \frac{1}{2^n} \left[\frac{i}{2^n - 1} - \frac{1}{2} \right]^2 = \frac{1}{2^n} \sum_{i=0}^{2^n-1} \left(\frac{i^2}{(2^n - 1)^2} - \frac{i}{2^n - 1} + \frac{1}{4} \right) =$$

$$= \frac{1}{2^n} \left(\frac{(2^n - 1)2^n(2^{n+1} - 1)}{6(2^n - 1)^2} - \frac{(2^n - 1)2^n}{2(2^n - 1)} + \frac{1}{4} \right) = \frac{1}{12} \frac{2^n + 1}{2^n - 1}.$$

Таким образом, математическое ожидание квазиравномерной случайной величины совпадает с математическим ожиданием равномерной случайной последовательности интервала $(0, 1)$, а дисперсия отличается только множителем $(2^n + 1)/(2^n - 1)$, 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* [4, 29].

Требования к генератору случайных чисел. Прежде чем перейти к описанию конкретных алгоритмов получения на ЭВМ последовательностей псевдослучайных чисел, сформулируем набор требований, которым должен удовлетворять идеальный генератор. Полученные с помощью идеального генератора псевдослучайные последовательности чисел должны состоять из квазиравномерно распределенных чисел, содержать статистически независимые числа, быть воспроизводимыми, иметь неповторяющиеся числа, получаться с минимальными затратами машинного времени, занимать минимальный объем машинной памяти.

Наибольшее применение в практике моделирования на ЭВМ для генерации последовательностей псевдослучайных чисел находят алгоритмы вида

$$x_{i+1} = \Phi(x_i), \quad (4.9)$$

представляющие собой *рекуррентные соотношения* первого порядка, для которых начальное число x_0 и постоянные параметры заданы [36, 37].

Определим качественно требования к виду функции Φ . Например, легко показать, что функция вида (4.9), изображенная на рис. 4.10, а, не может породить хорошую последовательность псевдослучайных чисел x_1, x_2, \dots . Действительно, если построить точки с координатами $(x_1, x_2), (x_3, x_4)$ по случайным числам, полученным, например, из таблицы случайных чисел (табл. 4.2), то они будут равномерно распределены в единичном квадрате $0 \leq x_i \leq 1, 0 \leq x_{i+1} \leq 1$. Соответствующие же точки, построенные по числам $(x_1, \Phi(x_2)), (x_3, \Phi(x_4)), \dots$, располагаются в площади, ограниченной кривой $x_{i+1} = \Phi(x_i)$.

Хорошую последовательность случайных чисел может породить только такая функция $x_{i+1} = \Phi(x_i)$, график которой достаточно плотно заполняет единичный квадрат. Примером такой функции может служить $x_{i+1} = D(Ax_i)$ при больших целых положительных A , где $D(u) = u - C(u)$ — дробная часть числа u ; $C(u)$ — целая часть числа u , т. е. наибольшее целое число, не превосходящее u . Пусть для примера $A=10$, тогда функция $x_{i+1} = \Phi(x_i)$ будет иметь вид, показанный на рис. 4.10, б. Приведенные условия являются только необходимыми, но не достаточными для того, чтобы соотношение (4.9) порождало хорошие последовательности псевдослучайных чисел.

Рассмотрим некоторые процедуры получения последовательностей псевдослучайных квазирегулярно распределенных чисел, которые нашли применение в практике статистического моделирования систем на ЭВМ.

Одной из исторически первых процедур получения псевдослучайных чисел была процедура, получившая название метода серединных квадратов. Пусть имеется $2n$ -разрядное число, меньшее 1: $x_i = 0, a_1 a_2 \dots a_{2n}$. Возведем его в квадрат: $x_i^2 = 0, b_1 b_2 \dots b_{4n}$, а затем отберем средние $2n$ разрядов $x_{i+1} = 0, b_{n+1} b_{n+2} \dots b_{3n}$, которые и будут являться очередным числом псевдослучайной последовательности. Например, если начальное число $x_0 = 0,2152$, то $(x_0)^2 = 0,04631104$, т. е. $x_1 = 0,6311$, затем $(x_1)^2 = 0,39828721$, т. е. $x_2 = 0,8287$, и т. д.

Недостаток этого метода — наличие корреляции между числами последовательности, а в ряде случаев случайность вообще может отсутствовать. Например, если $x_0 = 0,4500$, то $(x_0)^2 = 0,20250000$, $x_1 = 0,2500$, $(x_1)^2 = 0,06250000$, $x_2 = 0,2500$, $(x_2)^2 = 0,06250000$, $x_3 = 0,2500$ и т. д. Кроме того, при некоторых i^* вообще может наблюдаться вырождение последовательности, т. е. $x_i = 0$ при $i \geq i^*$. Это существенно ограни-

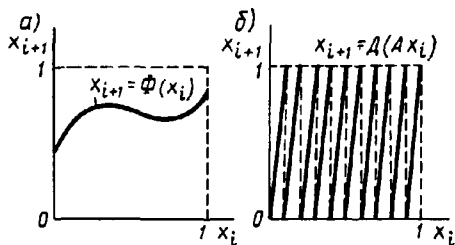


Рис. 4.10. Вид функции Φ

чивает возможности использования метода серединных квадратов.

Конгруэнтные процедуры генерации. Широкое применение при моделировании систем на ЭВМ получили *конгруэнтные процедуры* генерации псевдослучайных последовательностей, представляющие собой арифметические операции, в основе которых лежит фундаментальное понятие конгруэнтности. Два целых числа α и β конгруэнтны (сравнимы) по модулю m , где m — целое число, тогда и только тогда, когда существует такое целое число k , что $\alpha - \beta = km$, т. е. если разность $\alpha - \beta$ делится на m и если числа α и β дают одинаковые остатки от деления на абсолютную величину числа m . Например, $1984 \equiv 4 \pmod{10}$, $5008 \equiv 8 \pmod{10^3}$ и т. д.

Таблица 4.2

x_i					
86515	90795	66155	66434	56558	12332
69186	03393	42502	99224	88955	53758
41686	42163	85181	38967	33181	72664
86522	47171	88059	89342	67248	09082
72587	93000	89688	78416	27589	99528
94377	57802	52452	42499	33346	83935
91641	18867	76773	97526	27256	66447
53807	00607	04825	82134	80317	75120
12311	90316	87113	84778	45863	24520
14480	50961	84754	57616	38132	64294
79130	90410	45420	77757	75593	51435
25731	37525	16287	66181	73244	61870
45904	75601	70492	10274	23974	14783
19976	04925	07824	76044	32373	05312
15218	49286	89571	42903	59598	56774
73189	64448	31276	70795	33071	96929
28709	38238	76208	76575	53163	58481
17932	66686	64254	57598	26623	91730
94590	22561	70177	03569	21302	17381
08749	43448	28484	16325	62786	31466
91682	12804	29142	65877	64517	31466
87653	98088	75162	97496	59297	79636
79429	66186	59157	95114	16021	30890
85444	39453	67981	49687	36801	38666
85739	44326	91641	40837	93030	03675
02555	52905	84637	76154	14150	07876
74364	16796	59575	32764	91090	66515
21656	93662	81305	58846	69558	41675
50055	11244	29835	35801	23472	22700
18788	91332	32795	54313	39072	16809
41899	69207	66785	87225	05498	51512
16107	52141	88898	23775	30649	86545
39976	21279	36694	85970	22148	60102
18465	87650	65509	66065	91016	50324

Конгруэнтные процедуры являются чисто детерминированными, так как описываются в виде рекуррентного соотношения, когда функция (4.9) имеет вид

$$X_{i+1} = \lambda X_i + \mu \pmod{M}, \quad (4.10)$$

где X_i, λ, μ, M — неотрицательные целые числа.

Раскроем рекуррентное соотношение (4.10):

$$X_1 = \lambda X_0 + \mu \pmod{M};$$

$$X_2 = \lambda X_1 + \mu = \lambda^2 X_0 + (\lambda + 1)\mu \pmod{M};$$

$$X_3 = \lambda X_2 + \mu = \lambda^3 X_0 + (\lambda^2 + \lambda + 1)\mu = \lambda^3 X_0 + (\lambda^3 - 1)\mu / (\lambda - 1) \pmod{M};$$

$$\dots \dots \dots$$

$$X_i = \lambda^i X_0 + (\lambda^i - 1)\mu / (\lambda - 1) \pmod{M}.$$

$$(4.11)$$

Если заданы начальное значение X_0 , множитель λ и аддитивная константа μ , то (4.11) однозначно определяет последовательность целых чисел $\{X_i\}$, составленную из остатков от деления на M членов последовательности $\{\lambda^i X_0 + \mu(\lambda^i - 1)/(\lambda - 1)\}$. Таким образом, для любого $i \geq 1$ справедливо неравенство $X_i < M$. По целым числам последовательности $\{X_i\}$ можно построить последовательность $\{x_i\} = \{X_i/M\}$ рациональных чисел из единичного интервала $(0, 1)$.

Конгруэнтная процедура получения последовательностей псевдослучайных квазиравномерно распределенных чисел может быть реализована мультипликативным либо смешанным методом.

Мультипликативный метод. Задаст последовательность неотрицательных целых чисел $\{X_i\}$, не превосходящих M , по формуле

$$X_{i+1} = \lambda X_i \pmod{M}, \quad (4.12)$$

т. е. это частный случай соотношения (4.10) при $\mu = 0$.

В силу детерминированности метода получаются воспроизводимые последовательности. Требуемый объем машинной памяти при этом минимален, а с вычислительной точки зрения необходим последовательный подсчет произведения двух целых чисел, т. е. выполнение операции, которая быстро реализуется современными ЭВМ.

Для машинной реализации наиболее удобна версия $M = p^g$, где p — число цифр в системе счисления, принятой в ЭВМ ($p = 2$ для двоичной и $p = 10$ для десятичной машины); g — число битов в машинном слове. Тогда вычисление остатка от деления на M сводится к выделению g младших разрядов делимого, а преобразование целого числа X_i в рациональную дробь из интервала $x_i \in (0, 1)$ осуществляется подстановкой слева от X_i двоичной или десятичной запятой.

Алгоритм построения последовательности для двоичной машины $M = 2^g$ сводится к выполнению таких операций [31, 36, 46]:

1. Выбрать в качестве X_0 произвольное нечетное число.
2. Вычислить коэффициент $\lambda = 8t \pm 3$, где t — любое целое положительное число.
3. Найти произведение λX_0 , содержащее не более $2g$ значащих разрядов.
4. Взять g младших разрядов в качестве первого члена последовательности X_1 , а остальные отбросить.
5. Определить дробь $x_1 = X_1/2^g$ из интервала $(0, 1)$.
6. Присвоить $X_0 = X_1$.
7. Вернуться к п. 3.

Пример 4.4. Необходимо получить числа последовательности для случая $g=4$, используя приведенный алгоритм мультипликативного метода. Для этого выполняем следующие действия: 1. Выбираем $X_{010} = 7$ (в десятичной системе счисления) или $X_0 = 0111$ (в двоичной системе счисления). 2. Найдем $t=1$, тогда $\lambda_{10} = 11$ или 5; пусть $\lambda_{10} = 5$, $\lambda = 0101$. 3. Рассчитываем произведение λX_0 , берем g младших разрядов, вычисляем X_1 и присваиваем $X_0 = X_1$, т. е. выполняем п. 3 — 7 алгоритма:

- а) $\lambda X_0 = (0101)(0111) = 00100011$, $X_1 = 0011$, $x_1 = 3/16 = 0,1875$;
- б) $\lambda X_1 = (0101)(0011) = 00001111$, $X_2 = 1111$, $x_2 = 15/16 = 0,9375$;
- в) $\lambda X_2 = (0101)(1111) = 01001011$, $X_3 = 1011$, $x_3 = 11/16 = 0,6875$;
- г) $\lambda X_3 = (0101)(1011) = 00110111$, $X_4 = 0111$, $x_4 = 7/16 = 0,4375$.

Смешанный метод. Позволяет вычислить последовательность неотрицательных целых чисел $\{X_i\}$, не превосходящих M , по формуле

$$X_{i+1} = \lambda X_i + \mu \pmod{M},$$

т. е. в отличие от мультипликативного метода $\mu \neq 0$. С вычислительной точки зрения смешанный метод генерации сложнее мультипликативного на одну операцию сложения, но при этом возможность выбора дополнительного параметра позволяет уменьшить возможную корреляцию получаемых чисел.

Качество конкретной версии такого генератора можно оценить только с помощью соответствующего машинного эксперимента.

В настоящее время почти все пакеты прикладных программ универсальных ЭВМ для вычисления последовательностей равномерно распределенных случайных чисел основаны на конгруэнтной процедуре [17, 38, 40].

4.3. ПРОВЕРКА И УЛУЧШЕНИЕ КАЧЕСТВА ПОСЛЕДОВАТЕЛЬНОСТЕЙ ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ

Эффективность статистического моделирования систем на ЭВМ и достоверность получаемых результатов существенным образом зависят от качества исходных (базовых) последовательностей псевдослучайных чисел, которые являются основой для получения стохастических воздействий на элементы моделируемой системы. Поэтому, прежде чем приступать к реализации моделирующих алгоритмов на ЭВМ, необходимо убедиться в том, что исходная

последовательность псевдослучайных чисел удовлетворяет предъявляемым к ней требованиям, так как в противном случае даже при наличии абсолютно правильного алгоритма моделирования процесса функционирования системы S по результатам моделирования нельзя достоверно судить о характеристиках системы [29, 37].

Проверка качества последовательностей. 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. Рассмотрим возможные методы проведения таких проверок, наиболее часто используемые в практике статистического моделирования систем.

Проверка равномерности последовательностей псевдослучайных квазиравномерно распределенных чисел $\{x_i\}$ может быть выполнена по гистограмме с использованием косвенных признаков [4, 26]. Суть проверки по гистограмме сводится к следующему. Выдвигается гипотеза о равномерности распределения чисел в интервале $(0, 1)$. Затем интервал $(0, 1)$ разбивается на m равных частей, тогда при генерации последовательности $\{x_i\}$ каждое из чисел x с вероятностью $p_j = 1/m$, $j = \overline{1, m}$, попадает в один из подынтервалов. Всего в каждый j -й подынтервал попадает N_j чисел последовательности $\{x_i\}$, $i = \overline{1, N}$, причем $N = \sum_{j=1}^m N_j$. Относительная частота попадания случайных чисел последовательности $\{x_i\}$ в каждый из подынтервалов будет равна N_j/N . Вид соответствующей гистограммы для примера показан на рис. 4.11, а, где пунктирная линия соответствует теоретическому значению p_j , а сплошная — экспериментальному N_j/N . Очевидно, что если числа x_i принадлежат псевдослучайной квазиравномерно распределенной последовательности, то при достаточно больших N экспериментальная гистограмма (ломаная линия на рис. 4.11, а) приблизится к теоретической прямой $1/m$.

Оценка степени приближения, т. е. равномерности последовательности $\{x_i\}$, может быть проведена с использованием критериев согласия. На практике обычно принимается $m = 20 \div 50$, $N = (10^2 \div 10^3)m$.

Суть проверки равномерности по косвенным признакам сводится к следующему. Генерируемая последовательность чисел $\{x_i\}$ разбивается на две последовательности:

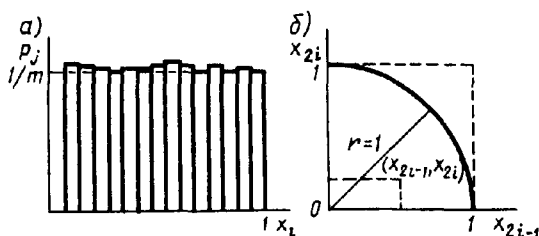


Рис. 4.11. Проверка равномерности последовательности

$$x_1, x_3, x_5, \dots, x_{2i-1};$$

$$x_2, x_4, x_6, \dots, x_{2i}; \quad i = \overline{1, N}.$$

Затем проводится следующий эксперимент. Если выполняется условие

$$x_{2i-1}^2 + x_{2i}^2 < 1, \quad i = \overline{1, N}, \quad (4.13)$$

то фиксируется наступление некоторого события и в счетчик событий добавляется единица. После $N/2$ опытов, когда генерировано N число, в счетчике будет некоторое число $k \leq N/2$.

Геометрически условие (4.13) означает, что точка (x_{2i-1}, x_{2i}) , $i = \overline{1, N}$, находится внутри четверти круга радиусом $r=1$, что иллюстрируется рис. 4.11, б. В общем случае точка (x_{2i-1}, x_{2i}) всегда попадает внутрь единичного квадрата. Тогда теоретически вероятность попадания этой точки в четверть круга

$$p_x = S_{1/4 \text{ круга}} / S_{\text{квадрата}} = (\pi^2/4)/(1 \cdot 1) = \pi/4.$$

Если числа последовательности $\{x_i\}$ равномерны, то в силу закона больших чисел теории вероятностей при больших N относительная частота $2k/N \rightarrow \pi/4$.

Проверка стохастичности последовательностей псевдослучайных чисел $\{x_i\}$ наиболее часто проводится методами комбинаций и серий [7, 11, 25]. Сущность метода комбинаций сводится к определению закона распределения длин участков между единицами (нулями) или закона распределения (появления) числа единиц (нулей) в n -разрядном двоичном числе X_i . На практике длину последовательности N берут достаточно большой и проверяют все n разрядов или только l старших разрядов числа X_i .

Теоретически закон появления j единиц в l разрядах двоичного числа X_i описывается исходя из независимости отдельных разрядов биномиальным законом распределения:

$$P(j, l) = C_l^j p^j (1-p)^{l-j} = C_l^j p^l,$$

где $P(j, l)$ — вероятность появления j единиц в l разрядах числа X_i ; $p(1) = p(0) = 0,5$ — вероятность появления единицы (нуля) в любом разряде числа X_i ; $C_l^j = l! / [j!(l-j)!]$.

Тогда при фиксированной длине выборки N теоретически ожидаемое число появления случайных чисел X_i с j единицами в проверяемых l разрядах будет равно

$$n_j = N C_l^j p^l.$$

После нахождения теоретических и экспериментальных вероятностей $P(j, l)$ или чисел n_j при различных значениях $l \leq n$ гипотеза о стохастичности проверяется с использованием критериев согласия [7, 11, 18, 21].

При анализе стохастичности последовательности чисел $\{x_i\}$ методом серий последовательность разбивается на элементы первого и второго рода (a и b), т. е.

$$x_i = \begin{cases} a, & \text{если } x_i < p; \\ b & \text{в противном случае,} \end{cases}$$

где $0 < p < 1$.

Серией называется любой отрезок последовательности, состоящий из идущих друг за другом элементов одного и того же рода, причем число элементов в отрезке (a или b) называется *длиной серии*.

После разбиения последовательности $\{x_i\}$ на серии первого и второго рода будем иметь, например, последовательность вида

$$\dots aabbbbbaabaabaabbbab \dots$$

Так как случайные числа a и b в данной последовательности независимы и принадлежат последовательности $\{x_i\}$, равномерно распределенной на интервале $(0, 1)$,

то теоретическая вероятность появления серии длиной j в последовательности длиной l в N опытах (под опытом здесь понимается генерация числа x_i , и проверка условия $x_i < p$) определяется формулой Бернулли:

$$P(j, l) = C_p^j (1-p)^{l-j}, \quad j = \overline{0, l}, \quad l = \overline{1, n}.$$

В случае экспериментальной проверки оцениваются частоты появления серий длиной j . В результате получаются теоретическая и экспериментальная зависимости $P(j, l)$, сходимость которых проверяется по известным критериям согласия, причем проверку целесообразно проводить при различных значениях p , $0 < p < 1$ и l .

Проверка независимости элементов последовательности псевдослучайных квазиравномерно распределенных чисел проводится на основе вычисления корреляционного момента [4].

Случайные величины ξ и η называются независимыми, если закон распределения каждой из них не зависит от того, какое значение приняла другая. Таким образом, независимость элементов последовательности $\{x_i\}$ может быть проверена путем введения в рассмотрение последовательности $\{y_j\} = \{x_{i+\tau}\}$, где τ — величина сдвига последовательностей.

В общем случае корреляционный момент дискретных случайных величин ξ и η с возможными значениями x_i и y_j определяется по формуле

$$K_{\xi\eta} = \sum_i \sum_j (x_i - M\{\xi\})(y_j - M\{\eta\}) p_{ij},$$

где p_{ij} — вероятность того, что (ξ, η) примет значение (x_i, y_j) .

Корреляционный момент характеризует рассеивание случайных величин ξ и η и их зависимость. Если случайные числа независимы, то $K_{\xi\eta} = 0$. Коэффициент корреляции

$$\bar{\rho}_{\xi\eta} = K_{\xi\eta} / (\sigma_x \sigma_y),$$

где $\sigma_x - \sigma_y$ — средние квадратические отклонения величин ξ и η .

При проведении оценок коэффициента корреляции на ЭВМ удобно для вычисления использовать следующее выражение:

$$\bar{\rho}_{\xi\eta}(\tau) = \frac{\frac{1}{N-\tau} \sum_{i=1}^{N-\tau} x_i x_{i+\tau} - \frac{1}{(N-\tau)^2} \sum_{i=1}^{N-\tau} x_i \sum_{i=1}^{N-1} x_{i+\tau}}{\sqrt{D[x_i] D[x_{i+\tau}]}} ,$$

где

$$D[x_i] = \frac{1}{N-\tau} \sum_{i=1}^{N-\tau} x_i^2 - \frac{1}{(N-\tau)^2} \left(\sum_{i=1}^{N-\tau} x_i \right)^2,$$

$$D[x_{i+\tau}] = \frac{1}{N-\tau} \sum_{i=1}^{N-\tau} x_{i+\tau}^2 - \frac{1}{(N-\tau)^2} \left(\sum_{i=1}^{N-\tau} x_{i+\tau} \right)^2.$$

При вычислениях сначала рационально определить суммы:

$$\sum_i x_i, \quad \sum_i x_{i+\tau}, \quad \sum_i x_i x_{i+\tau}, \quad \sum_i x_i^2, \quad \sum_i x_{i+\tau}^2.$$

При любом $\tau \neq 0$ для достаточно больших N с доверительной вероятностью β справедливо соотношение

$$|\bar{\rho}_{i\tau}(\tau)| \leq \beta \sqrt{1/N}.$$

Если найденное эмпирическое значение $\bar{\rho}_{i\tau}(\tau)$ находится в указанных пределах, то с вероятностью β можно утверждать, что полученная последовательность чисел $\{x_i\}$ удовлетворяет гипотезе корреляционной независимости.

Характеристики качества генераторов. При статистическом моделировании системы S с использованием программных генераторов псевдослучайных квазиравномерных последовательностей важными характеристиками качества генератора является *длина периода* P и *длина отрезка аперiodичности* L . Длина отрезка аперiodичности L псевдослучайной последовательности $\{x_i\}$, заданной уравнением

$$X_{i+1} \equiv \lambda X_i + \mu \pmod{M}, \quad x_i = X_i/M,$$

есть наибольшее целое число, такое, что при $0 \leq j < i \leq L$ событие $P\{x_i = x_j\}$ не имеет места. Это означает, что все числа x_i в пределах отрезка аперiodичности не повторяются.

Очевидно, что использование при моделировании систем последовательности чисел $\{x_i\}$, длина которой больше отрезка аперiodичности L , может привести к повторению испытаний в тех же условиях, что и раньше, т. е. увеличение числа реализаций не дает новых статистических результатов.

Способ экспериментального определения длины периода P и длины отрезка аперiodичности L сводится к следующему [29]. Запускается программа генерации последовательности $\{x_i\}$ с начальным значением x_0 и генерируется V чисел x_i . В большинстве практических случаев можно полагать $V = (1 \div 5) 10^6$. Генерируются числа последовательности $\{x_i\}$ и фиксируется число x_V .

Затем программа запускается повторно с начальным числом x_0 и при генерации очередного числа проверяется истинность события $P'\{x_i = x_V\}$. Если это событие истинно: $i = i_1$ и $i = i_2$ ($i_1 < i_2 < V$), то вычисляется длина периода последовательности $P = i_2 - i_1$. Проводится запуск программы генерации с начальными числами x_0 и x_P . При этом фиксиру-

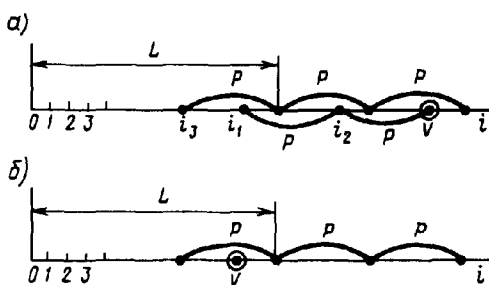


Рис. 4.12. Экспериментальное определение длины периода и длины отрезка аперiodичности:

а — вариант 1, б — вариант 2

ется минимальный номер $i=i_3$, при котором истинно событие $P'' \{x_i = x_{P+i}\}$, и вычисляется длина отрезка аперiodичности $L=i_3+P$ (рис. 4.12, а). Если P' оказывается истинным лишь для $i=V$, то $L>V$ (рис. 4.12, б).

В некоторых случаях достаточно громоздкий эксперимент по определению длин периода и отрезка аперiodичности можно заменить аналитическим расчетом, как это показано в следующем примере.

Пример 4.5. Необходимо показать, что в последовательности чисел $\{x_i\}$, описываемой уравнением

$$X_{i+1} \equiv \lambda X_i \pmod{M}, \quad x_i = \frac{X_i}{M},$$

при простом модуле M можно так выбрать коэффициент λ , что при любом X_0 , взаимно простом с M , длина отрезка аперiodичности, совпадающая в этом случае с длиной периода P , будет $L=P=M-1$. Иначе говоря, надо найти, при каких условиях равенство

$$X_s \equiv X_0 \pmod{M} \quad (4.14)$$

справедливо при минимальном значении $s=M-1$.

Можно записать [см. (4.11)], что $X_{i+s} \equiv \lambda^s X_i \pmod{M}$, поэтому (4.14) имеет место при

$$\lambda^s \equiv 1 \pmod{M} \quad (4.15)$$

(здесь существенно, что X_0 взаимно просто с M).

По условию требуется, что наименьший показатель степени $s=P_M(A)$, удовлетворяющий (4.15) и называемый показателем A по модулю M , был равен $M-1$. Для любого простого модуля M существует $\varphi(M-1)$ значений λ (первообразных корней), удовлетворяющих уравнению (4.15) при $P_M(A)=\varphi(M)$, где $\varphi(M)$ — функция Эйлера, определяемая как число натуральных чисел $m \leq M$, взаимно простых с M . Для простого модуля M имеем $\varphi(M)=M-1$.

Таким образом, доказано существование многих λ , при которых повторение элементов последовательности $\{x_i\}$ наступит на $(M-1)$ -м числе x_{M-1} , т. е. $L=P=M-1$, что и требовалось доказать.

Для алгоритмов получения последовательностей чисел $\{x_i\}$ общего вида (4.10) экспериментальная проверка является сложной (из-за наличия больших P и L), а расчетные соотношения в явном виде не получены. Поэтому в таких случаях целесообразно провести теоретическую оценку длины отрезка аперiodичности последовательности L . Для этого воспользуемся элементарной вероятностной моделью, рассмотренной в следующем примере [4, 36, 37].

Пример 4.6. Пусть имеется конечное множество, содержащее N различных чисел. Проведем последовательность независимых опытов, в каждом из которых из этого множества извлекается и записывается одно число. Вероятность извлечения любого числа в каждом из опытов равна $1/N$, так как выборка чисел проводится с возвратом. Обозначим через L случайную величину — номер опыта, в котором впервые будет снова извлечено уже записанное ранее число. Можно доказать, что в данной вероятностной модели для любого $x > 0$ имеем

$$\lim_{N \rightarrow \infty} P\{L/\sqrt{N} < x\} = 1 - e^{-x^2/2}.$$

Так как математическое ожидание случайной величины с такой функцией распределения равно $\sqrt{\pi}/2$, то при $N \rightarrow \infty$ получим $M[L] = \sqrt{\pi}N/2$. Такая оценка длины отрезка аperiodичности «груба», но полезна на практике для предварительного определения L с целью дальнейшего уточнения экспериментальным путем.

Рассмотрим некоторые особенности статистической проверки стохастичности псевдослучайных последовательностей. Для такой проверки могут быть использованы различные статистические критерии оценки, например критерии Колмогорова, Пирсона и т. д. Но в практике моделирования чаще всего пользуются более простыми приближенными способами проверки [29, 37].

Для проверки равномерности базовой последовательности случайных чисел x_i , $i = \overline{1, N}$, можно воспользоваться такими оценками:

$$(1/N) \sum_{i=1}^N x_i = 1/2, \quad (1/N) \sum_{i=1}^N x_i^2 = 1/3.$$

Для проверки таблиц случайных цифр обычно применяют различные тесты, в каждом из которых цифры классифицируются по некоторому признаку и эмпирические частоты сравниваются с их математическими ожиданиями с помощью критерия Пирсона [29, 46].

Для проверки аппаратных генераторов случайных чисел можно использовать те же приемы, что и для проверки последовательностей псевдослучайных чисел, полученных программным способом. Особенностью такой проверки будет то, что проверяются не те числа, которые потом будут необходимы для моделирования системы S . Поэтому кроме проверки качества выдаваемых генератором случайных чисел должна еще гарантироваться устойчивая работа генератора на время проведения машинного эксперимента с моделью M_n .

Улучшение качества последовательностей. В силу рассмотренных преимуществ основное применение в практике имитационного моделирования систем находят различные программные способы получения чисел. Поэтому рассмотрим возможные методы улучшения качества последовательностей псевдослучайных чисел. Одним из наиболее употребительных методов такого улучшения является употребление вместо формул вида (4.9), представляющих собой рекуррентные формулы первого порядка, рекуррентных формул порядка r , т. е.

$$x_{i+1} = \Phi(x_i, x_{i-1}, \dots, x_{i-r+1}),$$

где начальные значения x_0, x_1, \dots, x_{r-1} заданы. В этом случае длина отрезка аperiodичности L у такой последовательности при $r > 1$

гораздо больше, чем при $r = 1$. Однако при этом возрастает сложность метода, что приводит к увеличению затрат машинного времени на получение чисел и ограничивает возможности его применения на практике.

Для получения последовательности псевдослучайных чисел с большой длиной отрезка аperiodичности L можно воспользоваться методом возмущений [29, 37]. В основу этого метода получения последовательности чисел положена формула вида

$$x_{i+1} = \begin{cases} \Phi(x_i), & \text{если } i \equiv 0 \pmod{M}, \\ \Psi(x_i), & \text{если } i \not\equiv 0 \pmod{M}, \end{cases}$$

где функции $\Phi(u)$ и $\Psi(u)$ различны.

В этом случае в основном используется формула $x_{i+1} = \Phi(x_i)$, и только когда i кратно M , последовательность «возмущается», т. е. реализуется переход к формуле $x_{i+1} = \Psi(x_i)$. Целое число M называется *периодом возмущения*.

Все рассмотренные критерии проверки последовательностей псевдослучайных чисел являются необходимыми при постановке имитационных экспериментов на ЭВМ с моделью M_x , но об их достаточности можно говорить лишь при рассмотрении задачи моделирования конкретной системы S .

4.4. МОДЕЛИРОВАНИЕ СЛУЧАЙНЫХ ВОЗДЕЙСТВИЙ НА СИСТЕМЫ

При моделировании системы S методом имитационного моделирования, в частности методом статистического моделирования на ЭВМ, существенное внимание уделяется учету случайных факторов и воздействий на систему. Для их формализации используются случайные события, дискретные и непрерывные величины, векторы, процессы. Формирование на ЭВМ реализаций случайных объектов любой природы из перечисленных сводится к генерации и преобразованию последовательностей случайных чисел. Вопросы генерации базовых последовательностей псевдослучайных чисел $\{x_i\}$, имеющих равномерное распределение в интервале $(0, 1)$, были рассмотрены в § 4.2, поэтому остановимся на вопросах преобразования последовательностей случайных чисел $\{x_i\}$ в последовательность $\{y_i\}$ для имитации воздействий на моделируемую систему S .

Эти задачи очень важны в практике имитационного моделирования систем на ЭВМ, так как существенное количество операций, а значит, и временных ресурсов ЭВМ расходуется на действия со случайными числами. Таким образом, наличие эффективных методов, алгоритмов и программ формирования, необходимых для моделирования конкретных систем последовательностей случайных чисел $\{y_i\}$, во многом определяет возможности практического ис-

пользования машинной имитации для исследования и проектирования систем [37, 46].

Моделирование случайных событий. Простейшими случайными объектами при статистическом моделировании систем являются *случайные события*. Рассмотрим особенности их моделирования [4].

Пусть имеются случайные числа x_i , т. е. возможные значения случайной величины ξ , равномерно распределенной в интервале (0, 1). Необходимо реализовать случайное событие A , наступающее с заданной вероятностью p . Определим A как событие, состоящее в том, что выбранное значение x_i случайной величины ξ удовлетворяет неравенству

$$x_i \leq p. \quad (4.16)$$

Тогда вероятность события A будет $P(A) = \int_0^p dx = p$. Противоположное событие \bar{A} состоит в том, что $x_i > p$. Тогда $P(\bar{A}) = 1 - p$.

Процедура моделирования в этом случае состоит в выборе значений x_i и сравнении их с p . При этом, если условие (4.16) выполняется, исходом испытания является событие A .

Таким же образом можно рассмотреть группу событий. Пусть A_1, A_2, \dots, A_r — полная группа событий, наступающих с вероятностями p_1, p_2, \dots, p_r соответственно. Определим A_m как событие, состоящее в том, что выбранное значение x_i случайной величины ξ удовлетворяет неравенству

$$l_{m-1} < x_i \leq l_m, \quad (4.17)$$

где $l_r = \sum_{i=1}^r p_i$. Тогда

$$P(A_m) = \int_{l_{m-1}}^{l_m} dx = p_m.$$

Процедура моделирования испытаний в этом случае состоит в последовательном сравнении случайных чисел x_i со значениями l_i . Исходом испытания оказывается событие A_m , если выполняется условие (4.17). Эту процедуру называют определением исхода испытания *по жребью* в соответствии с вероятностями p_1, p_2, \dots, p_r .

Эти процедуры моделирования были рассмотрены в предположении, что для испытаний применяются случайные числа x_i , имеющие равномерное распределение в интервале (0, 1). При моделировании на ЭВМ используются псевдослучайные числа с квазиравномерным распределением, что приводит к некоторой ошибке. Оценим ее.

Пример 4.7. Пусть имеются n -разрядные случайные числа с возможными значениями $x_i^* = i/(2^n - 1)$, $i = 0, 2^n - 1$. Подставив в (4.16) вместо x_i число x_i^* , определим A^* как событие, состоящее в том, что $x_i^* \leq p$.

Вероятность наступления события A^* может быть определена как $P(A^*) = m/2^n$, где m — количество случайных чисел, меньших или равных p . Отсюда следует, что использование числа x_i^* вместо x_i приводит к ошибке в определении вероятности события $\Delta p = m/2^n - p$.

Очевидно, что максимальное значение ошибки не превосходит величины $1/(2^n - 1)$. Таким образом, для уменьшения влияния ошибок можно воспользоваться увеличением разрядности случайных чисел.

При моделировании систем часто необходимо осуществить такие испытания, при которых искомый результат является сложным событием, зависящим от двух (я более) простых событий. Пусть, например, независимые события A и B имеют вероятности наступления p_A и p_B . Возможными исходами совместных испытаний в этом случае будут события $AB, \bar{A}B, A\bar{B}, \bar{A}\bar{B}$ с вероятностями $p_A p_B, (1 - p_A)p_B, p_A(1 - p_B), (1 - p_A)(1 - p_B)$.

Для моделирования совместных испытаний можно использовать два варианта процедуры: 1) последовательную проверку условия (4.16); 2) определение одного из исходов $AB, \bar{A}B, A\bar{B}, \bar{A}\bar{B}$ по жребию с соответствующими вероятностями, т. е. аналогия (4.17). Первый вариант требует двух чисел x_i и сравнений для проверки условия (4.16). При втором варианте можно обойтись одним числом x_i , но сравнений может потребоваться больше. С точки зрения удобства построения моделирующего алгоритма и экономии количества операций и памяти ЭВМ более предпочтителен первый вариант.

Рассмотрим теперь случай, когда события A и B являются зависимыми и наступают с вероятностями p_A и p_B . Обозначим через $P(B/A)$ условную вероятность наступления события B при условии, что событие A произошло. При этом считаем, что условная вероятность $P(B/A)$ задана.

Рассмотрим один из вариантов построения модели. Из последовательности случайных чисел $\{x_i\}$ извлекается очередное число x_m и проверяется справедливость неравенства $x_m < p_A$. Если это неравенство справедливо, то наступило событие A . Для испытания, связанного с событием B , используется вероятность $P(B/A)$. Из совокупности чисел $\{x_i\}$ берется очередное число x_{m+1} и проверяется условие $x_{m+1} \leq P(B/A)$. В зависимости от того, выполняется или нет это неравенство, исходом испытания являются AB или $A\bar{B}$.

Если неравенство $x_m < p_A$ не выполняется, то наступило событие \bar{A} . Поэтому для испытания, связанного с событием B , необходимо определить вероятность

$$P(B/\bar{A}) = [P(B) - P(A)P(B/A)] / (1 - P(A)).$$

Выберем из совокупности $\{x_i\}$ число x_{m+1} и проверим справедливость неравенства $x_{m+1} \leq P(B/\bar{A})$. В зависимости от того, выполняется оно или нет, получим исходы испытания $\bar{A}B$ или $\bar{A}\bar{B}$.

Логическая схема алгоритма для реализации этого варианта модели показана на рис. 4.13. Здесь ВИД[...] — процедура ввода исходных данных; ГЕН[...] — генератор равномерно распределенных случайных чисел; $XM \equiv x_m$; $XMI \equiv x_{m+1}$; $PA \equiv p_A$; $PB \equiv p_B$; $PVA \equiv P(B/A)$; $PVNA \equiv P(B/\bar{A})$; KA , KNA , KAB , $KANB$, $KNAV$, $KNANB$ — число событий A , \bar{A} , AB , $\bar{A}B$, $\bar{A}B$, $\bar{A}\bar{B}$ соответственно; ВРМ[...] — процедура выдачи результатов моделирования.

Рассмотрим особенности моделирования на ЭВМ марковских цепей, служащих, например, для формализации процессов в P -схемах (см. § 2.4). Простая однородная марковская цепь определяется матрицей переходов

$$P = \begin{pmatrix} p_{11} & p_{12} & \dots & p_{1k} \\ p_{21} & p_{22} & \dots & p_{2k} \\ \dots & \dots & \dots & \dots \\ p_{k1} & p_{k2} & \dots & p_{kk} \end{pmatrix}, \quad 0 \leq p_{ij} \leq 1,$$

где p_{ij} — вероятность перехода из состояния z_i в состояние z_j .

Матрица переходов P полностью описывает марковский процесс. Такая матрица является стохастической, т. е. сумма элементов каждой строки равна единице: $\sum_{j=1}^k p_{ij} = 1$; $i = \overline{1, k}$.

Обозначим через $p_i(n)$, $i = \overline{1, k}$, вероятности того, что система будет находиться в состоянии z_i после n переходов. По определению, $\sum_{i=1}^k p_i(n) = 1$.

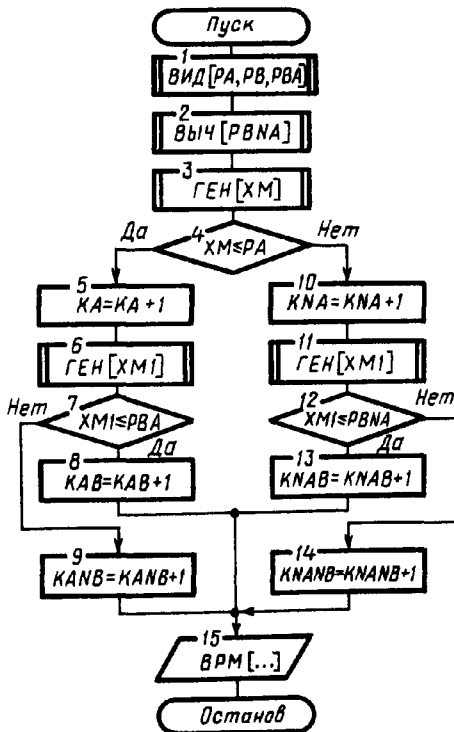


Рис. 4.13. Схема моделирующего алгоритма при зависимых событиях

Используя событийный подход, можно подойти к моделированию марковской цепи следующим образом. Пусть возможными исходами испытаний являются события A_1, A_2, \dots, A_k . Вероятность p_{ij} — это условная вероятность наступления события A_j в данном испытании при условии, что исходом предыдущего испытания было событие A_i . Моделирование такой цепи Маркова состоит в последовательном выборе событий A_j по жребию с вероятностями p_{ij} .

Сначала выбирается начальное состояние z_0 , задаваемое начальными вероятностями $p_1(0), p_2(0), \dots, p_k(0)$. Для этого из последовательности чисел $\{x_i\}$ выбирается число x_m и сравнивается с l , из (4.17), где в качестве p_i используются значения $p_1(0), p_2(0), \dots, p_k(0)$. Таким образом, выбирается номер m_0 , для которого оказывается справедливым неравенство (4.17). Тогда начальным событием данной реализации цепи будет событие A_{m_0} . Затем выбирается следующее случайное число x_{m+1} , которое сравнивается с l , где в качестве p_i используются p_{m_0j} . Определяется номер m_1 , и следующим событием данной реализации цепи будет событие A_{m_1} и т. д. Очевидно, что каждый номер m_i определяет не только очередное событие A_{m_i} формируемой реализации, но и распределение вероятностей $p_{m_i1}, p_{m_i2}, \dots, p_{m_ik}$ для выбора очередного номера m_{i+1} , причем для эргодических марковских цепей влияние начальных вероятностей быстро уменьшается с ростом номера испытаний. *Эргодическим* называется всякий марковский процесс, для которого предельное распределение вероятностей $p_i(n), i = \overline{1, k}$, не зависит от начальных условий $p_i(0)$. Поэтому при моделировании можно принимать, что

$$p_1(0) = p_2(0) = \dots = p_k(0) = 1/k.$$

Аналогично можно построить и более сложные алгоритмы, например для моделирования неоднородных марковских цепей [29].

Рассмотренные способы моделирования реализаций случайных объектов дают общее представление о наиболее типичных процедурах формирования реализаций в моделях процессов функционирования стохастических систем, но не исчерпывают всех приемов, используемых в практике статистического моделирования на ЭВМ.

Для формирования возможных значений *случайных величин* с заданным законом распределения исходным материалом служат базовые последовательности случайных чисел $\{x_i\}$, имеющие равномерное распределение в интервале $(0, 1)$. Другими словами, случайные числа x_i как возможные значения случайной величины ζ , имеющей равномерное распределение в интервале $(0, 1)$, могут быть преобразованы в возможные значения y_j случайной величины η , закон распределения которой задан [4].

Моделирование дискретных случайных величин. Рассмотрим особенности преобразования для случая получения дискретных слу-

j	1	2	3	4	5	6	7
y_j	0	1	2	3	4	5	6
p_j	0,01562	0,09375	0,23438	0,31250	0,23438	0,09375	0,01562
$\sum_{j=1}^n p_j$. . .	0,01562	0,10937	0,34375	0,65625	0,89063	0,98438	1,00000

Например, получив из равномерного распределения число $x_1=0,85393$ и проведя сравнения по алгоритму (4.21), найдем, что $x_1=0,85393$ 0,89063, т. е. $y_1=4$.

При этом среднее число циклов сравнения $\bar{\mu}=1 \cdot 0,01562 + 2 \cdot 0,09375 + 3 \cdot 0,23438 + 4 \cdot 0,31250 + 5 \cdot 0,23438 + 6(0,09375 + 0,01562) \approx 3,98$.

Пример 4.9. Необходимо проверить стохастичность последовательности из N случайных чисел $\{y_j\}$, полученных при имитации биномиального распределения при заданных параметрах n и p . Простейшим способом проверки является оценка выполнения условий

$$\frac{1}{N} \sum_{j=1}^N y_j \approx np, \quad \frac{1}{N} \sum_{j=1}^N y_j^2 \approx np(1+np-p).$$

Проверим на соответствие биномиальному распределению с параметрами $n=5$ и $p=0,1$ такой последовательности случайных чисел: $\{y_j\}=0, 0, 1, 0, 1, 0, 2, 0, 1, 0$; $N=10$.

Вычислим $np=0,5$; $np(1+np-p)=0,7$;

$$\frac{1}{N} \sum_{j=1}^N y_j = \frac{5}{10} = 0,5, \quad \frac{1}{N} \sum_{j=1}^N y_j^2 = \frac{7}{10} = 0,7.$$

Как видно из оценок, данная последовательность чисел $\{y_j\}$ хорошо в условиях данного примера представляет биномиальное распределение с заданными параметрами.

Можно привести и другие примеры алгоритмов и программ получения дискретных случайных величин с заданным законом распределения, которые находят применение в практике моделирования систем на ЭВМ.

Моделирование непрерывных случайных величин. Рассмотрим особенности генерации на ЭВМ 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 $\eta = F^{-1\eta}(\xi)$, 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 $(0, 1)$ [4, 29]. 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

$$\int_{-\infty}^{y_j} f_{\eta}(y) dy = x_i. \quad (4.22)$$

Рассмотрим некоторые примеры получения методом обратной функции непрерывных случайных величин с заданным законом распределения на основе случайных чисел, имеющих равномерное распределение в интервале $(0, 1)$.

Пример 4.10. Необходимо получить случайные числа с показательным законом распределения

$$f_{\eta}(y) = \lambda e^{-\lambda y}, \quad y > 0.$$

В силу соотношения (4.22) получим

$$\lambda \int_0^{y_j} e^{-\lambda y} dy = x_i,$$

где x_i — случайное число, имеющее равномерное распределение в интервале $(0, 1)$. Тогда

$$1 - e^{-\lambda y_j} = x_i, \quad y_j = -\frac{1}{\lambda} \ln(1 - x_i).$$

Учитывая, что случайная величина $\xi_1 = 1 - \xi$ имеет также равномерное распределение в интервале $(0, 1)$, можно записать $y_j = -\frac{1}{\lambda} \ln x_i$.

Пример 4.11. Необходимо получить случайные числа y , с законом распределения

$$f_{\eta}(y) = \lambda(1 - \lambda y/2), \quad 0 \leq y \leq 2/\lambda.$$

Воспользовавшись (4.22), получим $\lambda(y_j - \lambda y_j^2/4) = x_i$. Отсюда

$$y_j = 2(1 - \sqrt{1 - x_i})/\lambda, \quad \text{или} \quad y_j = 2(1 - \sqrt{x_i})/\lambda.$$

Можно привести и другие примеры использования соотношения (4.22). Но этот способ получения случайных чисел с заданным законом распределения имеет ограниченную сферу применения в практике моделирования систем на ЭВМ, что объясняется следующими обстоятельствами: 1) для многих законов распределения, встречающихся в практических задачах моделирования, интеграл

(4.22) не берется, т. е. приходится прибегать к численным методам решения, что увеличивает затраты машинного времени на получение каждого случайного числа; 2) даже для случаев, когда интеграл (4.22) берется в конечном виде, получаются формулы, содержащие действия логарифмирования, извлечения корня и т. д., которые выполняются с помощью стандартных подпрограмм ЭВМ, содержащих много исходных операций (сложения, умножения и т. п.), что также резко увеличивает затраты машинного времени на получение каждого случайного числа.

Поэтому в практике моделирования систем часто пользуются приближенными способами преобразования случайных чисел, которые можно классифицировать следующим образом: а) универсальные способы, с помощью которых можно получать случайные числа с законом распределения любого вида; б) неуниверсальные способы, пригодные для получения случайных чисел с конкретным законом распределения [4, 46].

Рассмотрим приближенный универсальный способ получения случайных чисел, основанный на кусочной аппроксимации функции плотности. Пусть требуется получить последовательность случайных чисел $\{y_i\}$ с функцией плотности $f_\eta(y)$, возможные значения которой лежат в интервале (a, b) . Представим $f_\eta(y)$ в виде кусочно-постоянной функции, т. е. разобьем интервал (a, b) на m интервалов, как это показано на рис. 4.14, и будем считать $f_\eta(y)$ на каждом интервале постоянной. Тогда случайную величину η можно представить в виде $\eta = a_k + \eta_k^*$, где a_k — абсцисса левой границы k -го интервала; η_k^* — случайная величина, возможные значения которой располагаются равномерно внутри k -го интервала, т. е. на каждом участке $a_k \div a_{k+1}$ величина η_k^* считается распределенной равномерно. Чтобы аппроксимировать $f_\eta(y)$ наиболее удобным для практических целей способом, целесообразно разбить (a, b) на интервалы так, чтобы вероятность попадания случайной величины η в любой интервал (a_k, a_{k+1}) была постоянной, т. е. не зависела от номера интервала k . Таким образом, для вычисления a_k воспользуемся следующим соотношением:

$$\int_{a_k}^{a_{k+1}} f_\eta(y) dy = 1/m. \quad (4.23)$$

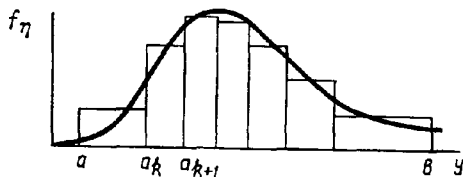


Рис. 4.14. Кусочная аппроксимация функции плотности

Алгоритм машинной реализации этого способа получения случайных чисел сводится к последовательному выполнению следующих действий: 1) генерируется случайное равномерно распределенное число x , из интервала $(0, 1)$; 2) с помощью этого числа

случайным образом выбирается интервал (a_k, a_{k+1}) ; 3) генерируется число x_{i+1} и масштабируется с целью приведения его к интервалу (a_k, a_{k+1}) , т. е. домножается на коэффициент $(a_{k+1} - a_k)x_{i+1}$; 4) вычисляется случайное число $y_j = a_k + (a_{k+1} - a_k)x_{i+1}$ с требуемым законом распределения.

Рассмотрим более подробно процесс выборки интервала (a_k, a_{k+1}) с помощью случайного числа x_j . Целесообразно для этой цели построить таблицу (сформировать массив), в которую предварительно поместить номера интервалов k и значения коэффициента масштабирования, определенные из соотношения (4.23) для приведения числа к интервалу (a, b) . Получив из генератора случайное число x_i , с помощью этой таблицы сразу определяем абсциссу левой границы a_k и коэффициент масштабирования $(a_{k+1} - a_k)$.

Достоинства этого приближенного способа преобразования случайных чисел: при реализации на ЭВМ требуется небольшое количество операций для получения каждого случайного числа, так как операция масштабирования (4.23) выполняется только один раз перед моделированием, и количество операций не зависит от точности аппроксимации, т. е. от количества интервалов m .

Рассмотрим способы преобразования последовательности равномерно распределенных случайных чисел $\{x_i\}$ в последовательность с заданным законом распределения $\{y_j\}$ на основе предельных теорем теории вероятностей. Такие способы ориентированы на получение последовательностей чисел с конкретным законом распределения, т. е. не являются универсальными [29, 43]. Поясним сказанное примерами.

Пример 4.12. Пусть требуется получить последовательность случайных чисел $\{y_j\}$, имеющих нормальное распределение с математическим ожиданием a и средним квадратическим отклонением σ :

$$f_{\eta}(y) = e^{-\frac{(y-a)^2}{2\sigma^2}} / \sqrt{2\pi\sigma}.$$

Будем формировать случайные числа y_j в виде сумм последовательностей случайных чисел $\{x_i\}$, имеющих равномерное распределение в интервале $(0, 1)$. Так как исходной (базовой) последовательностью случайных чисел $\{x_i\}$ при суммировании является последовательность чисел, имеющих равномерное распределение в интервале $(0, 1)$, то можно воспользоваться центральной предельной теоремой для одинаково распределенных случайных величин (см. § 4.1): если независимые одинаково распределенные случайные величины x_1, \dots, x_N имеют каждая математическое ожида-

ние a_1 и среднее квадратическое отклонение σ_1 , то сумма $\sum_{i=1}^N x_i$ асимптотически нормальна с математическим ожиданием $a = Na_1$ и средним квадратическим отклонением $\sigma = \sigma_1 \sqrt{N}$.

Как показывают расчеты, сумма $\sum_{i=1}^N x_i$ имеет распределение, близкое к нормальному, уже при сравнительно небольших N . Практически для получения последовательности нормально распределенных случайных чисел можно пользоваться значени-

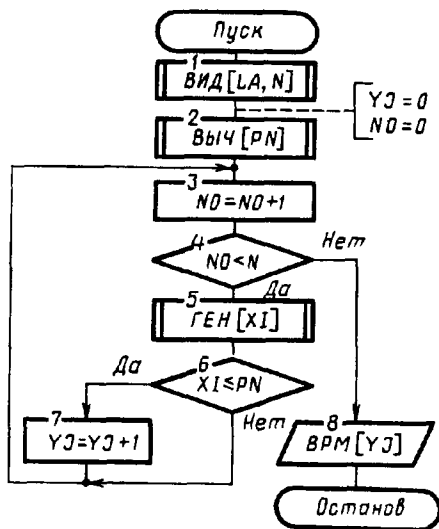


Рис. 4.15. Схема алгоритма генерации последовательности случайных чисел, имеющих пуассоновское распределение

приближенно следовать закону Пуассона, причем тем точнее, чем больше N . Практически N должно выбираться таким образом, чтобы $p \leq 0,1 + 0,2$ [4].

Алгоритм генерации последовательности случайных чисел y_j , имеющих пуассоновское распределение, с использованием данного способа показан на рис. 4.15. Здесь $LA \equiv \lambda$; $N \equiv N$; $PN \equiv p$; $XI \equiv x_i$ — случайные числа последовательности, равномерно распределенной в интервале $(0, 1)$; $YJ \equiv y_j$; NO — вспомогательная переменная; ВИД [...] — процедура ввода исходных данных; ВЫЧ [...] — процедура вычисления; ГЕН [...] — процедура генерации случайных чисел; ВРМ [...] — процедура выдачи результатов моделирования.

Моделирование случайных векторов. При решении задач исследования 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 [4].

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

ями $N=8+12$, а в простейших случаях — меньшими значениями N , например $N=4+5$ [4].

Пример 4.13. Пусть необходимо получить случайные числа, имеющие закон распределения Пуассона:

$$P(m) = \frac{\lambda^m}{m!} e^{-\lambda}.$$

Для этого воспользуемся предельной теоремой Пуассона (см. § 4.1): если P — вероятность наступления события A при одном испытании, то вероятность наступления m событий в N независимых испытаниях при $N \rightarrow \infty$, $p \rightarrow 0$, $Np = \lambda$ асимптотически равна $P(m)$.

Выберем достаточно большое N , такое, чтобы $p = \lambda/N < 1$, будем проводить серии по N независимых испытаний, в каждом из которых событие A происходит с вероятностью p , и будем подсчитывать число случаев y_j фактического наступления события A в серии с номером j . Числа y_j будут

значению x_i случайной величины ξ будет соответствовать

$$p_i = \sum_{j=1}^n p_{ij}.$$

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

$$p_{i_1,1}, p_{i_1,2}, \dots, p_{i_1,n} \quad (4.24)$$

которая описывает условное распределение величины η при условии, что $\xi = x_i$. Затем по тем же правилам определяем конкретное значение y_i случайной величины η в соответствии с распределением вероятностей (4.24). 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

$$p_{i_2,1}, p_{i_2,2}, \dots, p_{i_2,n} \quad (4.25)$$

и находим y_{i_2} в соответствии с распределением (4.25). 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_1} , а затем при условии, что $\xi = x_{i_1}$, определить условное распределение случайной величины η :

$$f_{\eta}(y/\xi = x_{i_1}) = f(x, y)/f_{\xi}(x_{i_1}).$$

В соответствии с этой функцией плотности можно определить случайное число y_{i_1} . Тогда пара чисел (x_{i_1}, y_{i_1}) будет являться искомой реализацией вектора (ξ, η) .

Рассмотренный способ формирования реализаций двумерных векторов можно обобщить и на случай многомерных случайных векторов. Однако при больших размерностях этих векторов объем вычислений существенно увеличивается, что создает препятствия к использованию этого способа в практике моделирования систем.

В пространстве с числом измерений более двух практически доступным оказывается формирование случайных векторов, заданных в рамках корреляционной теории. Рассмотрим случайный

вектор с математическими ожиданиями a_1, a_2, \dots, a_n и корреляционной матрицей

$$K = \begin{pmatrix} k_{11} & k_{12} & \dots & k_{1n} \\ k_{21} & k_{22} & \dots & k_{2n} \\ \dots & \dots & \dots & \dots \\ k_{n1} & k_{n2} & \dots & k_{nn} \end{pmatrix},$$

где $k_{ij} = k_{ji}$.

Пример 4.14. Для определенности остановимся на трехмерном случае $n=3$. Пусть требуется сформировать реализации трехмерного случайного вектора (ξ, η, ψ) , имеющего нормальное распределение с математическими ожиданиями $M[\xi]=a_1$, $M[\eta]=a_2$, $M[\psi]=a_3$ и корреляционной матрицей K , элементы которой k_{11} , k_{22} и k_{33} являются дисперсиями случайных величин ξ , η и ψ соответственно, а элементы $k_{12}=k_{21}$, $k_{13}=k_{31}$, $k_{23}=k_{32}$ представляют собой корреляционные моменты ξ и η , ξ и ψ , η и ψ соответственно.

Пусть имеется последовательность некоррелированных случайных чисел $\{v_i\}$, имеющих одномерное нормальное распределение с параметрами a и σ . Выберем три числа v_1, v_2, v_3 и преобразуем их так, чтобы они имели характеристики a_1, a_2, a_3 и K . Искомые составляющие случайного вектора (ξ, η, ψ) обозначим как x, y, z и представим в виде линейного преобразования случайных величин v_i :

$$x = c_{11}(v_1 - a) + a_1, \quad y = c_{12}(v_1 - a) + c_{22}(v_2 - a) + a_2,$$

где c_{ij} — некоторые коэффициенты (пока не известные). Для вычисления этих коэффициентов воспользуемся элементами корреляционной матрицы K . Так как случайные величины v_1, v_2, v_3 независимы между собой, то $M[(v_i - a)(v_j - a)] = 0$ при $i \neq j$.

В итоге имеем:

$$\begin{aligned} k_{11} &= c_{11}^2 \sigma^2, & k_{22} &= c_{12}^2 \sigma^2 + c_{22}^2 \sigma^2, \\ k_{33} &= c_{13}^2 \sigma^2 + c_{33}^2 \sigma^2, & k_{12} &= c_{11} c_{12} \sigma^2, \\ k_{13} &= c_{11} c_{13} \sigma^2, & k_{23} &= c_{12} c_{13} \sigma^2 + c_{23}^2 \sigma^2. \end{aligned}$$

Решая эту систему уравнения относительно c_{ij} , получим:

$$\begin{aligned} c_{11} &= \sqrt{k_{11}}/\sigma, & c_{12} &= k_{12}/(\sigma \sqrt{k_{11}}), & c_{13} &= k_{13}/(\sigma \sqrt{k_{11}}), \\ c_{22} &= \sqrt{k_{11}k_{22} - k_{12}^2}/(\sigma \sqrt{k_{11}}), \\ c_{23} &= \sqrt{k_{11}k_{23} - k_{12}k_{13}}/(\sigma \sqrt{k_{11}}), \\ c_{33} &= \sqrt{k_{11}k_{33} - k_{13}^2 - k_{11}k_{23} + k_{12}k_{13}}/(\sigma \sqrt{k_{11}}). \end{aligned}$$

Вычислив коэффициенты c_{ij} , легко три последовательных случайных числа v_i , $i=1, 2, 3$, преобразовать в составляющие случайного вектора (x_i, y_i, z_i) , используя соотношения, приведенные выше [4].

При таком формировании реализаций случайного вектора требуется хранить в памяти ЭВМ $n(n+1)/2$ корреляционных моментов k_{ij} и n математических ожиданий a_i . При больших n в связи с этим могут встречаться затруднения при использовании полученных та-

ким способом многомерных случайных векторов для моделирования систем.

Конкретные алгоритмы имитации стохастических воздействий в процессе машинного эксперимента рассмотрим далее применительно к случаям использования для формализации процесса функционирования системы S типовых математических схем (см. гл. 8).

Контрольные вопросы

- 4.1. В чем сущность метода статистического моделирования систем на ЭВМ?
- 4.2. Какие способы генерации последовательностей случайных чисел используются при моделировании на ЭВМ?
- 4.3. Какая последовательность случайных чисел используется в качестве базовой при статистическом моделировании на ЭВМ?
- 4.4. Почему генерируемые на ЭВМ последовательности чисел называются псевдослучайными?
- 4.5. Что собой представляют конгруэнтные процедуры генерации последовательностей?
- 4.6. Какие существуют методы проверки (тестирования) качества генераторов случайных чисел?
- 4.7. Что собой представляет процедура определения исхода испытаний по жребью?
- 4.8. Какие существуют способы генерации последовательностей случайных чисел с заданным законом распределения на ЭВМ?

ГЛАВА 5

ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА МОДЕЛИРОВАНИЯ СИСТЕМ

Успех или неудача проведения имитационных экспериментов с моделями сложных систем существенным образом зависит от инструментальных средств, используемых для моделирования, т. е. набора аппаратно-программных средств, представляемых пользователю-разработчику или пользователю-исследователю машинной модели. В настоящее время существует большое количество языков имитационного моделирования — специальных языков программирования имитационных моделей на ЭВМ — и перед разработчиком машинной модели возникает проблема выбора языка, наиболее эффективного для целей моделирования конкретной системы. Языки моделирования заслуживают пристального внимания, так как, во-первых, число существующих языков и систем моделирования превышает несколько сотен и необходимо научиться ориентироваться в них, а во-вторых, почти каждый новый язык моделирования не только является средством, облегчающим доведение концептуальной модели до готовой машинной моделирующей программы, но и представляет собой новый способ «видения мира», т. е. построения моделей реальных систем. Необходимость эффективной реализации имитационных моделей предъявляет все более высокие требования как к инструментальным ЭВМ, так и к средствам организации информации в ЭВМ при моделировании. Поэтому с учетом использования новой информационной технологии в процессе моделирования следует учитывать особенности построения баз знаний и банков данных и систем управления ими.

5.1. ОСНОВЫ СИСТЕМАТИЗАЦИИ ЯЗЫКОВ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

Использование современных ЭВМ и вычислительных комплексов и сетей является мощным средством реализации имитационных моделей и исследования с их помощью характеристик процесса функционирования систем S . В ряде случаев в зависимости от сложности объекта моделирования, т. е. системы S , рационально использование персональных ЭВМ (ПЭВМ) или локальных вычислительных сетей (ЛВС). В любом случае эффективность исследования системы S на программно-реализуемой модели M_m прежде всего зависит от правильности схемы моделирующего алгоритма, совершенства программы и только косвенным образом зависит от технических характеристик ЭВМ, применяемой для моделирования. Большое значение при реализации модели на ЭВМ имеет вопрос правильного выбора языка моделирования.

Моделирование систем и языки программирования. *Алгоритмические языки* при моделировании систем служат вспомогательным аппаратом разработки, машинной реализации и анализа характеристик моделей. Каждый язык моделирования должен отражать определенную структуру понятий для описания широкого класса явлений. Выбрав для решения задачи моделирования процесса функционирования системы 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_x .

Основными моментами, характеризующими качество языков моделирования, являются: удобство описания процесса функционирования системы S , удобство ввода исходных данных моделирования и варьирования структуры, алгоритмов и параметров модели, реализуемость статистического моделирования, эффективность анализа и вывода результатов моделирования, простота отладки и контроля работы моделирующей программы, доступность восприятия и использования языка. Будущее языков моделирования определяется прогрессом в области создания мультимедийных систем машинной имитации, а также проблемно-ориентированных на цели моделирования информационно-вычислительных систем [17, 31, 41, 46].

Рассмотрим основные понятия, связанные с алгоритмическими языками и их реализацией на ЭВМ вообще и языками моделирования в частности.

Язык программирования представляет собой набор символов, распознаваемых ЭВМ и обозначающих операции, которые можно реализовать на ЭВМ. На низшем уровне находится основной язык машины, программа на котором пишется в кодах, непосредственно соответствующих элементарным машинным действиям (сложение, запоминание, пересылка по заданному адресу и т. д.). Следующий уровень занимает автокод (язык *АССЕМБЛЕРА*) вычислительной машины. Программа на автокоде составляется из мнемонических символов, преобразуемых в машинные коды специальной программой — ассемблером.

Компилятором называется программа, принимающая инструкции, написанные на алгоритмическом языке высокого уровня, и преобразующая их в программы на основном языке машины или на автокоде, которые в последнем случае транслируются еще раз с помощью ассемблера.

Интерпретатором называется программа, которая, принимая инструкции входного языка, сразу выполняет соответствующие опе-

рации в отличие от компилятора, преобразующего эти инструкции в запоминающиеся цепочки команд. Трансляция происходит в течение всего времени работы программы, написанной на языке интерпретатора. В отличие от этого компиляция и ассемблирование представляют собой однократные акты перевода текста с входного языка на объектный язык машины, после чего полученные программы выполняются без повторных обращений к транслятору.

Программа, составленная в машинных кодах или на языке АС-СЕМБЛЕРА, всегда отражает специфику конкретной ЭВМ. Инструкции такой программы соответствуют определенным машинным операциям и, следовательно, имеют смысл только в той ЭВМ, для которой они предназначены, поэтому такие языки называются *машинно-ориентированными языками*.

Большинство языков интерпретаторов и компиляторов можно классифицировать как процедурно-ориентированные языки. Эти языки качественно отличаются от машинно-ориентированных языков, описывающих элементарные действия ЭВМ и не обладающих проблемной ориентацией. Все *процедурно-ориентированные языки* предназначены для определенного класса задач, включая в себя инструкции, удобные для формулировки способов решения типичных задач этого класса. Соответствующие алгоритмы программируются в обозначениях, не связанных ни с какой ЭВМ.

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 [31].

Особенности использования алгоритмических языков. Рассмотрим преимущества и недостатки использования для моделирования процесса функционирования систем языков имитационного моделирования (ЯИМ) и языков общего назначения (ЯОН), т. е. универсальных и процедурно-ориентированных алгоритмических языков. Целесообразность использования 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.

Языки моделирования позволяют описывать моделируемые системы в терминах, разработанных на базе основных понятий имитации. До того как эти понятия были четко определены и формализованы в ЯИМ, не существовало единых способов описания имитационных задач, а без них не было связи между различными

разработками в области постановки имитационных экспериментов. Высокоуровневые языки моделирования являются удобным средством общения заказчика и разработчика машинной модели M_d .

Несмотря на перечисленные преимущества ЯИМ, в настоящее время выдвигаются основательные аргументы как технического, так и эксплуатационного характера против полного отказа при моделировании от универсальных и процедурно-ориентированных языков. Технические возражения против использования ЯИМ: вопросы эффективности рабочих программ, возможности их отладки и т. п. В качестве эксплуатационных недостатков упоминается нехватка документации по существующим ЯИМ, сугубо индивидуальный характер соответствующих трансляторов, усложняющий их реализацию на различных ЭВМ, и трудности исправления ошибок. Снижение эффективности ЯИМ проявляется при моделировании задач более разнообразных, чем те, на которые рассчитан конкретный язык моделирования. Но здесь следует отметить, что в настоящее время не существует и ЯОН, который был бы эффективен при решении задач любого класса.

Серьезные недостатки ЯИМ проявляются в том, что в отличие от широко применяемых ЯОН, трансляторы с которых включены в поставляемое изготовителем математическое обеспечение всех современных ЭВМ, языки моделирования, за небольшим исключением, разрабатывались отдельными организациями для своих достаточно узко специализированных потребностей. Соответствующие трансляторы плохо описаны и приспособлены для эксплуатации при решении задач моделирования систем, поэтому, несмотря на достоинства ЯИМ, приходится отказываться от их практического применения в ряде конкретных случаев.

При создании системы моделирования на базе любого языка необходимо решить вопрос о синхронизации процессов в модели, так как в каждый момент времени, протекающего в системе (системного времени), может потребоваться обработка нескольких событий, т. е. требуется псевдопараллельная организация имитируемых процессов в машинной модели M_d . Это является основной задачей монитора моделирования, который выполняет следующие функции: управление процессами (согласование системного и машинного времени) и управление ресурсами (выбор и распределение в модели ограниченных средств моделирующей системы).

Подходы к разработке языков моделирования. К настоящему времени сложились два различных подхода к разработке языков моделирования: непрерывный и дискретный — отражающие основные особенности исследуемых методом моделирования систем [35, 43, 46]. Поэтому 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 могут быть использованы не только АВМ, но и ЭВМ, последние при соответствующем программировании имитируют различные непрерывные процессы. При этом ЭВМ обладают большей надежностью в эксплуатации и позволяют получить высокую точность результатов, что привело к разработке языков моделирования, отображающих модель в виде блоков таких типов, которые играют роль стандартных блоков АВМ (усилителей, интеграторов, генераторов функций и т. п.). Заданная 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 устанавливаются соотношения между значениями функций в моменты времени t и $t + dt$ и 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 [46].

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_s , 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 системах S .

Архитектура языков моделирования. *Архитектуру ЯИМ*, т. е. концепцию взаимосвязей элементов языка как сложной системы, и технологию перехода от системы S к ее машинной модели M_s 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 состояние модели системы M в пространстве и во времени.

Типовая схема архитектуры ЯИМ и технология его использования при моделировании систем показана на рис. 5.1.

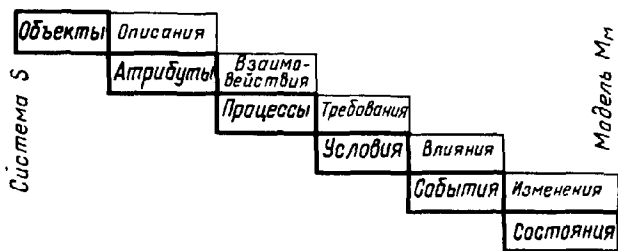


Рис. 5.1. Типовая схема архитектуры ЯИМ и технология его использования

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 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, обеспечивая 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 [17, 46].

Задание времени в машинной модели. Как уже отмечалось в гл. 3, существует два основных подхода к заданию времени: с помощью постоянных и переменных интервалов времени, которым соответствуют два принципа реализации моделирующих алгоритмов, т. е. «принцип Δt » и «принцип δz ».

Рассмотрим соответствующие способы управления временем в модели системы $M(S)$ на примере, показанном на рис. 5.2, где по оси реального времени отложена последовательность событий в системе $\{s_i\}$ во времени, причем события s_4 и s_5 происходят одновременно (рис. 5.2, а). Под действием событий s_i изменяются состояния модели z_i в момент времени t_{z_i} , причем такое изменение происходит скачком δz .

В модели, построенной по «принципу Δt » (рис. 5.2, б), моменты системного

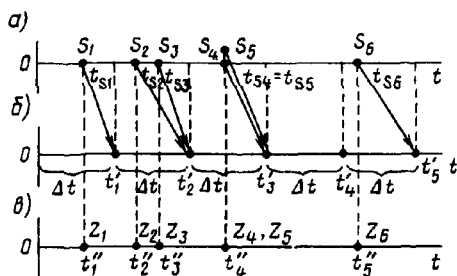


Рис. 5.2. Способы управления временем в модели системы

$t_5'' = t_{z5}$, т. е. моменты системного времени t_k'' (δz) непосредственно связаны с моментами появления событий в системе s_i .

У каждого из этих методов есть свои преимущества с точки зрения адекватного отражения реальных событий в системе S и затрат машинных ресурсов на моделирование. При использовании «принципа δz » события обрабатываются последовательно и время смещается каждый раз вперед до начала следующего события. В модели, построенной по «принципу Δt », обработка событий происходит по группам, пакетам или множествам событий. При этом выбор Δt оказывает существенное влияние на ход процесса и результаты моделирования, и если Δt задана неправильно, то результаты могут получиться недостоверными, так как все события появляются в точке, соответствующей верхней границе каждого интервала моделирования. При применении «принципа δz » одновременная обработка событий в модели имеет место только тогда, когда эти события появляются одновременно и в реальной системе. Это позволяет избежать необходимости искусственного введения ранжирования событий при их обработке в конце интервала Δt .

При моделировании по «принципу Δt » можно добиться хорошей аппроксимации: для этого Δt должно быть малым, чтобы два неодновременных события не попали в один и тот же временной интервал. Но уменьшение Δt приводит к увеличению затрат машинного времени на моделирование, так как значительная часть тратится на корректировку «часов» и отслеживание событий, которых в большинстве интервалов может и не быть. При этом даже при сильном «сжатии» Δt два неодновременных события могут попасть в один и тот же временной интервал Δt , что создает ложное представление об их одновременности.

Для выбора принципа построения машинной модели M_m и соответственно ЯИМ необходимо знать: цель и назначение модели; требуемую точность результатов моделирования; затраты машинного времени при использовании того или иного принципа; необходимый объем машинной памяти для реализации модели, постро-

времени будут последовательно принимать значения $t'_1 = \Delta t$, $t'_2 = 2\Delta t$, $t'_3 = 3\Delta t$, $t'_4 = 4\Delta t$, $t'_5 = 5\Delta t$. Эти моменты системного времени t'_i (Δt) никак не связаны с моментами появления событий s_i , которые имитируются в модели системы. Системное время при этом получает постоянное приращение, выбираемое и задаваемое перед началом имитационного эксперимента.

В модели, построенной по «принципу δz » (рис. 5.2, в), изменение времени наступает в момент смены состояния системы, и последовательность моментов системного времени имеет вид $t''_1 = t_{z1}$, $t''_2 = t_{z2}$, $t''_3 = t_{z3}$, $t''_4 = t_{z4}$,

енной по принципу Δt и δz ; трудоемкость программирования модели и ее отладки.

Требования к языкам имитационного моделирования. Таким образом, при разработке моделей систем возникает целый ряд специфических трудностей, поэтому в ЯИМ должен быть предусмотрен набор таких программных средств и понятий, которые не встречаются в обычных ЯОН.

Совмещение. Параллельно протекающие в реальных системах S процессы представляются с помощью последовательно работающей ЭВМ. Языки моделирования позволяют обойти эту трудность путем введения понятия системного времени, используемого для представления упорядоченных во времени событий.

Размер. Большинство моделируемых систем имеет сложную структуру и алгоритмы поведения, а их модели велики по объему. Поэтому используют динамическое распределение памяти, когда компоненты модели системы M_m появляются в оперативной памяти ЭВМ или покидают ее в зависимости от текущего состояния. Важным аспектом реализуемости модели M_m на ЭВМ в этом случае является блочность ее конструкции, т. е. возможность разбиения модели на блоки, подблоки и т. д.

Изменения. Динамические системы связаны с движением и характеризуются развитием процесса, вследствие чего пространственная конфигурация этих систем претерпевает изменения по времени. Поэтому во всех ЯИМ предусматривают обработку списков, отражающих изменения состояний процесса функционирования моделируемой системы S .

Взаимосвязанность. Условия, необходимые для свершения различных событий в модели M_m процесса функционирования системы S , могут оказаться весьма сложными из-за наличия большого количества взаимных связей между компонентами модели. Для разрешения связанных с этим вопросом трудностей в большинство ЯИМ включают соответствующие логические возможности и понятия теории множеств.

Стохастичность. Для моделирования случайных событий и процессов используют специальные программы генерации последовательностей псевдослучайных чисел, квазиравномерно распределенных на заданном интервале, на основе которых можно получить стохастические воздействия на модель M_m , имитируемые случайными величинами с соответствующим законом распределения.

Анализ. Для получения наглядного и удобного в практическом отношении ответа на вопросы, решаемые методом машинного моделирования, необходимо получать статистические характеристики процесса функционирования модели системы $M(S)$. Поэтому предусматривают в языках моделирования способы статистической обработки и анализа результатов моделирования.

Перечисленным требованиям при исследовании и проектировании различных систем S отвечают такие наиболее известные языки моделирования дискретных событий, как *SIMULA*, *SIMSCRIPT*, *GPSS*, *SOL*, *CSL* и др.

5.2. СРАВНИТЕЛЬНЫЙ АНАЛИЗ ЯЗЫКОВ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

За сравнительно небольшой срок в области машинного моделирования систем произошел скачок, который был обусловлен потребностью в принципиально новом методе исследования и развитием средств вычислительной техники и который в первую очередь выразился в увеличении количества специализированных ЯИМ, причем этот процесс имеет лавинообразный характер. К настоящему времени насчитывается несколько сотен развитых ЯИМ, поэтому очень важно разобраться в этом многообразии ЯИМ и выбрать для моделирования конкретной системы S наиболее эффективный язык [17, 31, 37, 46].

Основы классификации языков моделирования. Как уже отмечалось, для машинного моделирования системы S пригодны три способа проведения вычислений, в основе которых лежит применение цифровой, аналоговой и гибридной вычислительной техники. Рассмотрим методы моделирования систем с точки зрения использования языков программирования. При этом в данном параграфе опустим рассмотрение чисто аналоговых способов вычислений, так как они реализуются не программно, а путем составления электрических цепей, т. е. когда язык программирования не требуется (не обсуждая вопросы программирования АВМ). По этой же причине не будем рассматривать использование ЯИМ при гибридных методах вычислений. Тогда классификация языков для программирования моделей систем имеет вид, приведенный на рис. 5.3.

Для моделирования систем используются как универсальные и процедурно-ориентированные ЯОН, так и специализированные ЯИМ. При этом ЯОН предоставляют программисту-разработчику модели M_m больше возможностей в смысле гибкости разработки, отладки и использования модели. Но гибкость приобретается ценой больших усилий, затрачиваемых на программирование модели, так как организация выполнения операций, отсчет системного времени и контроль хода вычислений существенно усложняются.

Имеющиеся ЯИМ можно разбить на три основные группы, соответствующие трем типам математических схем: непрерывные, дискретные и комбинированные. Языки каждой группы предназначены для соответствующего представления системы S при создании ее машинной модели M_m .

В основе рассматриваемой классификации в некоторых ЯИМ лежит принцип формирования *системного времени*. Так как

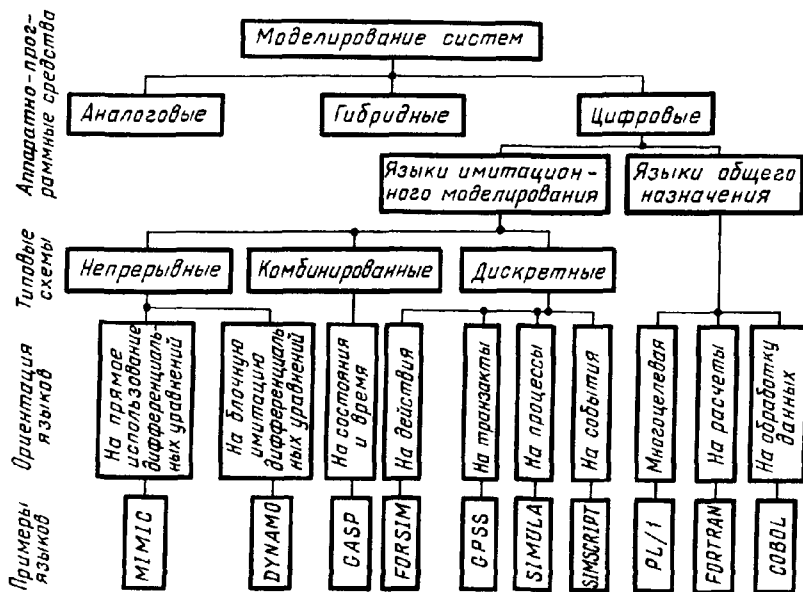


Рис. 5.3. Классификация языков для программирования моделей систем

«системные часы» предназначены не только для продвижения системного времени в модели M_m , но также для синхронизации различных событий и операций в модели системы S , то при отнесении того или иного конкретного языка моделирования к определенному типу нельзя не считаться с типом механизма «системных часов».

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 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 компонент заложены в струк-

туру, объединяющую дифференциальные уравнения и условия относительно переменных. Предполагается, что в системе могут наступать события двух типов: 1) события, зависящие от состояния z ; 2) события, зависящие от времени t . 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 *FORTRAN* 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.

Языки моделирования дискретных систем. 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 состояние 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.

При использовании ЯИМ второй группы после пересчета системного времени, в отличие от схемы языка событий, просмотр действий с целью проверки выполнения условий начала или окончания какого-либо действия производится непрерывно. Просмотр действий определяет очередность появления событий. Языки данного типа имеют в своей основе поисковый алгоритм, и динамика

системы S описывается в терминах действий. Примером языка действий (работ) является ЯИМ *FORSIM*, представляющий собой пакет прикладных программ, который позволяет оперировать только фиксированными массивами данных, описывающих объекты моделируемой системы. С его помощью нельзя имитировать системы переменного состава. При этом размеры массивов устанавливаются либо во время компиляции программы, либо в самом начале ее работы.

Язык *FORSIM* удобен для описания систем с большим числом разнообразных ресурсов, так как он позволяет записывать условия их доступности в компактной форме. Конкретный способ формализации модели на языке действий в достаточной степени произволен и остается на усмотрение программиста, что требует его достаточно высокой квалификации. Полное описание динамики модели M_m можно получить с помощью разных наборов подпрограмм.

Третья группа ЯИМ описывает системы, поведение которых определяется процессами. В данном случае под процессом понимается последовательность событий, связь между которыми устанавливается с помощью набора специальных отношений. Динамика заложена в независимо управляемых программах, которые в совокупности составляют программу процесса. Пример языка процессов — язык *SIMULA*, в котором осуществляется блочное представление моделируемой системы S с использованием понятия процесса для формализации элементов, на которые разбивается моделируемая система. Процесс задается набором признаков, характеризующих его структуру, и программой функционирования. Функционирование каждого процесса разбивается на этапы, протекающие в системном времени.

Главная роль в языке *SIMULA* отводится понятию параллельного оперирования с процессами в системном времени, а также универсальной обработке списков с процессами в роли компонент. Специальные языковые средства предусмотрены для манипуляций с упорядоченными множествами процессов.

В отдельную группу могут быть выделены ЯИМ типа *GPSS*, хотя принципиально их можно отнести к группе языков процессов. Язык *GPSS* представляет собой интерпретирующую языковую систему, применяющуюся для описания пространственного движения объектов. Такие динамические объекты в языке *GPSS* называются транзактами и представляют собой элементы потока. В процессе имитации транзакты «создаются» и «уничтожаются». Функцию каждого из них можно представить как движение через модель M_m с поочередным воздействием на ее блоки. Функциональный аппарат языка образуют блоки, описывающие логику модели, сообщая транзактам, куда двигаться и что делать дальше. Данные для ЭВМ подготавливаются в виде пакета управляющих

и определяющих карт, который составляется по схеме модели, набранной из стандартных символов. Созданная GPSS-программа, работая в режиме интерпретации, генерирует и передает транзакты из блока в блок в соответствии с правилами, устанавливаемыми блоками. Каждый переход транзакта приписывается к определенному моменту системного времени.

Сравнение эффективности языков. При анализе эффективности использования для моделирования конкретной системы S того или иного ЯИМ (или ЯОН) 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 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. 5.1. Языки даны в порядке уменьшения их эффективности.

Таблица 5.1

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, 12, 25, 34].

Выбор языка моделирования системы. Основываясь на классификации языков (рис. 5.3) и исходя из оценки эффективности (табл. 5.1), можно рассмотреть подход к выбору языка для решения задачи машинного моделирования конкретной системы S . Такой подход можно представить в виде дерева решений с соответствующими комментариями (рис. 5.4). Перед тем как пользоваться деревом

решений, разработчику машинной модели M_m необходимо выполнить все подэтапы первого этапа (построение концептуальной модели системы и ее формализация), а также предшествующие данному 6-му подэтапу подэтапы 2-го этапа (алгоритмизация модели и ее машинная реализация) (см. рис. 3.1). Приход в тот или иной конечный (нечетный) блок схемы дерева решений (рис. 5.4) означает рекомендацию более подробно рассмотреть указанные в нем технические средства или языки (ЯИМ и ЯОН), причем здесь для иллюстрации приведены лишь примеры основных языков, употребляемых наиболее часто при моделировании систем.

Исходя из постановки задачи машинного моделирования конкретной системы S , поставленных целей, выбранных критериев оценки эффективности и заданных ограничений (блок 1), можно сделать вывод о размерности задачи моделирования и требуемой точности и достоверности ее решения (блок 2). Для задач большой размерности моделирования на АВМ (блок 3) позволяет получить достаточно высокую точность. При этом АВМ позволит наглядно выявить компромисс между сложностью и точностью модели M , проиллюстрирует влияние изменения параметров и переменных на характеристики модели системы и т. п. Если в модели M при моделировании системы S имеют место как непрерывные, так и дискретные переменные, отражающие динамику системы и логику ее поведения (блок 4), то рекомендуется использовать для моделирования ГVK (блок 5). Подробно особенности и возможности

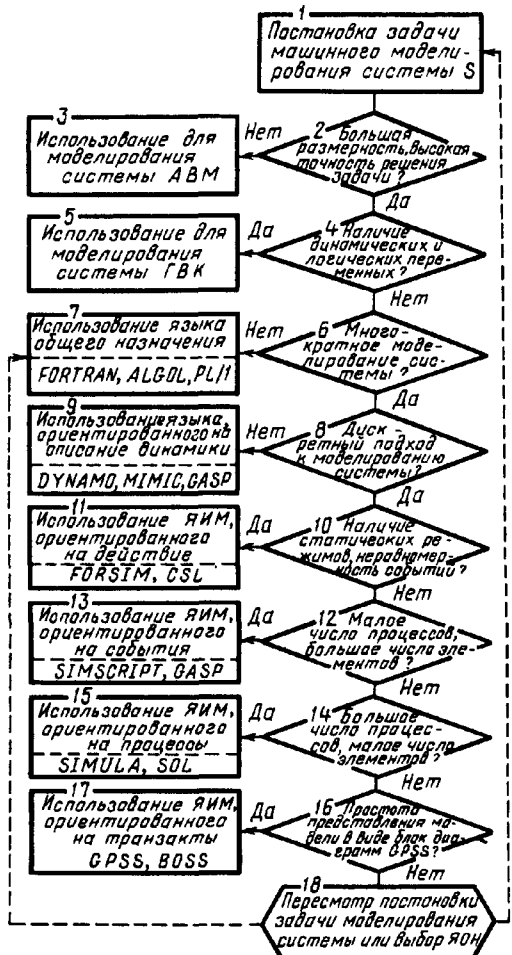


Рис. 5.4. Дерево решений выбора языка для моделирования системы

для задач большой размерности моделирования на АВМ (блок 3) позволяет получить достаточно высокую точность. При этом АВМ позволит наглядно выявить компромисс между сложностью и точностью модели M , проиллюстрирует влияние изменения параметров и переменных на характеристики модели системы и т. п. Если в модели M при моделировании системы S имеют место как непрерывные, так и дискретные переменные, отражающие динамику системы и логику ее поведения (блок 4), то рекомендуется использовать для моделирования ГVK (блок 5). Подробно особенности и возможности

применения гибридных (аналого-цифровых) моделирующих комплексов рассмотрены в § 5.5.

Если моделирование конкретной системы S представляет собой единичный акт (блок 6), то, вероятно, в ущерб концептуальной выразительности модели M_s и отладочным средствам для проверки логики машинной модели M_m следует выбрать более распространенные и более гибкие ЯОН (блок 7). Очевидно, на выбор конкретного языка существенно повлияют специфика модели M (особенности процесса функционирования системы S) и квалификация пользователя в программировании на конкретном языке.

Если при моделировании на универсальной ЭВМ выбран непрерывный подход (блок 8), то следует остановить выбор на одном из языков, позволяющих отразить динамику системы при наличии обратных связей (блок 9). При этом могут быть приняты языки непрерывного типа *DYNAMO*, *MIMIC* либо комбинированные (дискретно-непрерывные) — *GASP*.

Если в основу модели M положена дискретная математическая схема и в ней при построении моделирующего алгоритма используется «принцип Δt » (постоянный шаг во времени) или «принцип δz » (переменный шаг во времени, задаваемый сменой состояний), причем имитируются взаимодействующие элементы статической природы при неравномерности событий во времени (блок 10), то рационально воспользоваться ЯИМ, ориентированным на действия, например *FORSIM*, *CSL*.

Если в модели M описывается малое число взаимодействующих процессов и имеется большое число элементов (блок 12), то целесообразно выбрать для построения моделирующих алгоритмов «принцип Δt » и остановиться на ЯИМ событий (блок 13), например *SIMSCRIPT*, *GASP* и т. п.

Если для программирования модели более эффективен ЯИМ, позволяющий описать большое число взаимодействующих процессов (блок 14), то следует использовать языки процессов (блок 15), которые не связаны с использованием блоков только определенных типов, например в транзактных языках. Наиболее распространенными языками описания процессов являются языки *SIMULA* и *SOL*.

И наконец, если предпочтение отдается блочной конструкции модели M при наличии минимального опыта в программировании (блок 16), то следует выбирать ЯИМ транзактов типа *GPSS*, *BOSS* (блок 17), но при этом надо помнить, что они негибки и требуют большого объема памяти и затрат машинного времени на прогон программ моделирования.

Если перечисленные средства по той или иной причине не подходят для целей моделирования конкретной системы S (блок 18), то надо снова провести модификацию модели M либо попытаться решить задачу с использованием ЯОН на универсальной ЭВМ.

5.3. ПАКЕТЫ ПРИКЛАДНЫХ ПРОГРАММ МОДЕЛИРОВАНИЯ СИСТЕМ

Метод машинного моделирования все глубже входит в практику решения конкретных задач исследования и проектирования систем, находит свое применение для широкого круга проблем в различных сферах (автоматизированные системы управления, системы автоматизации научных исследований и экспериментов, информационно-вычислительные системы и сети коллективного пользования, системы автоматизированного проектирования и т. д.). В решение этих задач вовлекается все большее количество специалистов разных квалификаций, часто далеких от использования средств вычислительной техники. Поэтому для таких пользователей должны быть разработаны специальные средства подготовки и общения с ЭВМ, позволяющие автоматизировать этот трудоемкий процесс [17, 37].

Таким образом, возникает вопрос о создании *автоматизированной системы моделирования* (АСМ), которая должна повысить эффективность выполнения пользователем следующей совокупности процедур: преобразование к типовым математическим схемам элементов моделируемой системы S и построение схем сопряжения; обработка и анализ результатов моделирования системы S ; реализация интерактивного режима с пользователем в процессе моделирования системы S .

Понятие пакета прикладных программ. Создание проблемно-ориентированных комплексов, в том числе и АСМ, называемых пакетами прикладных программ, является важным направлением работ в современной вычислительной математике. При создании пакетов прикладных программ моделирования (ППМ) помимо разработки и отбора моделирующих алгоритмов и программ существенное место занимают работы по соответствующему системному обеспечению. Быстрота и удобство решения задач моделирования конкретных классов систем S при использовании ППМ достигаются сочетанием в единой архитектуре функционального наполнения, состоящего из модулей и покрывающего предметную область моделирования, и специализированных средств системного обеспечения, позволяющих сравнительно легко реализовать различные задания и обеспечивающих пользователя разнообразным сервисом при подготовке задач моделирования и проведении машинных экспериментов с моделью M_d .

Характерно, что в ходе разработки и машинного эксперимента модель объекта претерпевает многочисленные изменения, которые неизбежно влекут за собой изменения соответствующих рабочих программ. В настоящее время ППМ является практически единственной приемлемой формой организации программ моделирования, позволяющей «удержаться на плаву» в безбрежном море версий и вариантов исходной концептуальной модели. Кроме того,

пакетная организация программ машинного эксперимента с моделью M_x дает возможность систематизировать выполнение исследований, используя теорию планирования экспериментов и способствуя тем самым повышению достоверности получаемых результатов моделирования конкретной системы S .

Одной из важных проблем в области использования вычислительной техники для моделирования систем является проблема общения человека с ЭВМ при разработке модели и ее эксплуатации. Для повышения эффективности такого общения требуются соответствующие алгоритмы и программные средства. Здесь можно выделить три направления работ: 1) создание программных средств, обеспечивающих пользователя различными инструментами для автоматизации разработки программ; 2) создание программных средств, упрощающих процесс эксплуатации сетей ЭВМ инженерно-диспетчерским персоналом, а также обеспечивающих эффективное использование всех вычислительных ресурсов; 3) создание программных средств, предоставляющих пользователям разнообразные услуги при решении прикладных задач. Эти три направления сводятся соответственно к повышению уровня инструментальной, исполнительской и тематической квалификации вычислительной машины.

Таким образом, пакеты прикладных программ 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 АСМ.

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

Функциональное наполнение пакета. 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.

Уточним, что понимается под конструктивностью модуля. Прежде всего имеется в виду алгоритмическая конструктивность, так как модуль представляет собой элемент полученного в результате модульного анализа предметной области алгоритмического базиса, служащего основой для построения программ моделирования. Кроме того, на алгоритмическую конструктивность модулей влияют структуры типичных вычислительных алгоритмов, связи между элементами алгоритмического базиса, используемые в этих структурах, информационные потоки.

Помимо алгоритмической следует выделить и технологическую конструктивность модулей, определяемую дисциплиной работы в конкретной машинной модели M_m и системной средой, на базе которой разрабатывается и эксплуатируется ППМ. На технологическую конструктивность воздействуют такие факторы:

- формы представления программных модулей M_m ;
- виды управляющих связей между отдельными частями программных комплексов (открытые и закрытые подпрограммы);
- методы разработки (сверху вниз, снизу вверх и др.) программных комплексов, применяемые при работе с моделирующим алгоритмом системы S ;
- базовый язык или языки программирования, используемые при подготовке программ моделирования;
- ограничения на размеры программ моделирования;
- возможности штатных системных средств, обеспечивающих редактирование связей, загрузку и сегментацию программных комплексов, редактирование текстов.

Требования, вытекающие из алгоритмической и технологической конструктивности, составляют в совокупности регламент модуляризации, т. е. принятую разработчиками пакета форму представления материала в функциональном наполнении, а также способы его создания и эволюции. Если описание языка заданий рассматривать как спецификацию сопряжения пользователя с пакетом, то посредством регламента модуляризации определяется сопряжение с пакетом (точнее, с функциональным наполнением пакета) его разработчиков.

Язык заданий пакета. Язык заданий ППМ 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. Архитектура ППМ, т. е. предоставляющийся пользователю внешний вид АСМ, определяется тем, какие задачи система может решать и какие возможности дает она пользователю. Язык заданий отражает основные архитектурные решения, принятые разработчиками ППМ, стремившимися повысить уровень квалификации вычислительной системы в определенной прикладной области. Именно через язык заданий пользователь воспринимает и оценивает, какие «вычислительные услуги» предоставляет АСМ и насколько удобно их использование, т. е., другими словами, каков фактический уровень тематической квалификации системы.

Общая структура и стиль языка заданий ППМ в значительной степени зависят от дисциплины работы, принятой в обслуживаемой пакетом предметной области. Можно выделить две основные (в определенном смысле противоположные) дисциплины проведения моделирования:

— активную дисциплину, предусматривающую при создании конкретных рабочих программ модели M_m модификацию и настройку имеющихся модулей функционального наполнения, а также разработку новых модулей;

— пассивную дисциплину, предусматривающую проведение машинных экспериментов с моделью M_m без модификации функционального наполнения ППМ.

Активная дисциплина работы свойственна специалистам, создающим программное обеспечение АСМ, а пассивная дисциплина характерна для деятельности так называемых конечных пользователей, т. е. специалистов, которые не обязательно имеют высокий уровень подготовки в области программирования. Такое выделение двух дисциплин работы достаточно условно и преследует цель подчеркнуть контрастность системных подходов, используемых при автоматизации процесса моделирования.

Так, характерной особенностью языков заданий пакетов, обслуживающих проведение моделирования в режиме активной дисциплины, является их направленность на описание схем программ решения конкретных задач моделирования процессов, причем центральное место в таких языках (их обычно называют языками сборки) занимают не средства описания данных и манипулирования ими, что свойственно универсальным процедурно-ориентированным языкам программирования, а средства:

— конструирования схем программ, в которых указывается по-

рядок выполнения и взаимодействия модулей при моделировании конкретной системы S ;

— развития или модификации функционального наполнения ППМ;

— управления процессами генерации и исполнения рабочей программы, реализующей задание пользователя.

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.

Язык заданий ППМ может быть реализован как в форме самостоятельного языка, так и в форме встроенного языка, т. е. расширения существующего языка программирования. Независимо от формы реализации разработчик языка должен стремиться к тому, чтобы лексика, синтаксис и семантика языка заданий были как можно ближе к пользовательскому восприятию решаемых задач моделирования, т. е. чтобы языковые конструкции приближались к концептуальной модели M_x .

Системное наполнение пакета. 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. Реализация функций системного наполнения ППМ осуществляется на основе согласованного использования:

— штатных общецелевых средств системного обеспечения;

— средств системного наполнения, расширяющих и сопрягающих возможности компонентов штатного обеспечения;

— специальных средств системного наполнения, выполняющих управляющие, архивные и обрабатывающие процедуры с учетом специфики моделирования процесса функционирования системы S .

Язык (языки), на котором пишутся программы системного наполнения пакета, называется *инструментальным языком ППМ*.

Можно выделить такие ставшие уже традиционными составляющие системного наполнения пакета:

— резидентный монитор, осуществляющий интерфейс как

между отдельными компонентами системного наполнения, так и между ними и штатным программным обеспечением;

— транслятор входных заданий, формирующий внутреннее представление заданий и реализуемый обычно в виде макрогенератора или препроцессора;

— интерпретатор внутреннего представления задания;

— архив функционального наполнения (подсистема хранения программного материала);

— банк данных об объекте моделирования и машинном эксперименте;

— монитор организации процесса машинного моделирования (взаимодействия модулей по данным и управлению);

— планировщик процесса машинного моделирования, который определяет последовательность выполнения модулей, реализующую задание ППМ;

— монитор организации интерактивного взаимодействия с пользователем (исследователем системы S).

Программные средства АСМ. Напомним, что под ППМ, ориентированным на решение задач машинного моделирования систем, понимается комплекс программных средств и документов, предназначенных для реализации функционального завершеного алгоритма моделирования процесса функционирования системы S и обеспечивающих автоматизацию управления ведением эксперимента с моделью M_m на ЭВМ [17, 37].

Сущность такого определения состоит в том, что ППМ не является набором готовых программ для проведения машинных экспериментов с моделью M_m , а представляет собой набор средств для разработки конкретных, удовлетворяющих требованиям пользователя рабочих программ моделирования, служащих для автоматизации определенных функций при построении модели, машинном эксперименте и обработке результатов моделирования системы S .

К программным средствам ППМ относится набор программных модулей (тело пакета), из которых в соответствии с требованиями пользователя по заданному алгоритму набирается конкретная рабочая программа моделирования заданного объекта. В состав ППМ также входит управляющая программа, представляющая собой аналог супервизора ЭВМ; средства генерации рабочих программ для конкретного применения при решении задач моделирования систем. Специальная программа (монитор) принимает от пользователя информацию о требуемой модификации программ, формирует из набора стандартных модулей законченные рабочие программы, готовые к реализации машинного эксперимента с моделью M_m . Такой процесс генерации (настройки) ППМ на конкретные условия его использования создает значительную гибкость при решении задач автоматизации моделирования различных объектов.

Различают две разновидности генерации рабочих программ моделирования: статическую и динамическую.

При статической генерации из отдельных модулей формируется рабочая программа моделирования, необходимая пользователю при исследовании конкретного объекта. При этом определяются необходимые устройства ввода-вывода информации, описываются на специальном языке генерации необходимые свойства разрабатываемой программы. Созданная таким образом программа моделирования является одновариантной и при необходимости внесения изменений в процессе моделирования системы S требуется проведение новой генерации.

При динамической генерации заранее оговариваются все варианты рабочей программы моделирования системы S , которые могут потребоваться пользователю при машинном эксперименте с моделью M_m . При решении конкретной задачи моделирования, т. е. перед каждым новым прогоном программы в ходе машинного эксперимента, вводится специальная параметрическая карта, определяющая требуемый на этом прогоне вариант программы. Монитор пакета собирает необходимые модули и помещает их в оперативную память ЭВМ для решения задачи моделирования. Условия проведения машинного эксперимента при динамической генерации являются более гибкими, но при этом увеличиваются затраты машинных ресурсов на моделирование (увеличивается необходимый объем памяти и время моделирования каждого варианта модели системы S).

Кроме использования программных модулей, входящих в тело ППМ, пользователь имеет возможность подключать свои собственные программы моделирования в точках пользователя. Имеется также возможность замены имеющихся модулей ППМ на собственные, что еще больше расширяет возможности моделирования различных вариантов систем.

Таким образом, программные средства ППМ объединяют в себе три главных качества: 1) содержат алгоритмические решения по проведению моделирования и обработке результатов моделирования систем, доведенные до законченной машинной реализации; 2) имеют механизм автоматической настройки на параметры конкретных машинных экспериментов, задаваемые пользователем пакета на этапе генерации рабочих программ моделирования; 3) позволяют дополнять генерируемые ППМ рабочие программы моделирования пользовательскими блоками, расширяющими возможности проведения машинных экспериментов с конкретными объектами моделирования из заданного класса.

Кроме программных средств ППМ содержит комплект документов, т. е. ППМ является хорошо документированной системой для разработки рабочих программ. Без наличия этой документации использование ППМ становится неэффективным. В состав комплекта документов ППМ входят проектная документация, являющаяся

документацией разработчиков пакета, и пользовательская, необходимая для эксплуатации пакета при решении конкретных задач моделирования.

Существенный момент подготовки к генерации рабочих программ моделирования — обеспечение необходимой технической базы, т. е. выбор необходимой для реализации функциональных возможностей ППМ конфигурации технических средств. Это не только выбор типа ЭВМ и используемой версии операционной системы, но и выбор объема памяти, средств сбора, преобразования и представления информации. В описании каждого пакета указывается минимальная конфигурация технических средств, необходимая для его работы. При этом определяющими параметрами являются объем оперативной памяти ЭВМ, количество накопителей, необходимый набор средств ввода-вывода информации. Минимальная конфигурация технических средств обеспечивает удовлетворительную работу ППМ. Некоторое увеличение оперативной и промежуточной памяти, а также использование дополнительных периферийных средств повышает оперативность и расширяет возможности моделирования. Но при этом увеличиваются затраты на технические средства АСМ.

Структурно АСМ можно разбить на следующие комплексы программ: формирования базы данных об объекте моделирования (БДО); формирования базы данных о машинном эксперименте (БДЭ); моделирования процесса функционирования объекта; расширения возможностей ППМ; организации различных режимов работы ППМ.

Комплекс программ формирования БДО реализует все работы по созданию в АСМ сведений о моделируемом объекте, т. е. системе S . Причем в БДО эти сведения хранятся в стандартной форме, принятой в АСМ. Информация об объекте может корректироваться по мере получения новых сведений в процессе машинного моделирования. Для формирования БДО требуются следующие программы: ввода данных об объекте (сведения об элементах системы, типовых математических схемах и операторах их сопряжения); корректировки введенной информации; перевода в стандартную форму; диспетчеризации процедур ввода; формирования БДО (расположения информации во внешней памяти).

В результате работы комплекса программ формирования БДЭ в АСМ формируется база данных, т. е. сведения, достаточные для проведения конкретных экспериментов с машинной моделью объекта M_m . Информация для проведения машинного эксперимента переписывается из БДО и дополняется в соответствии с планом эксперимента факторами, реакциями, критериями оценки и т. п. Для формирования БДЭ необходимы следующие программы: ввода

данных о планируемом эксперименте (сведений о факторах, реакциях, начальных состояниях и т. п.); формирования БДЭ (выделения сведений из БДО, необходимых и достаточных для реализации конкретного машинного эксперимента с моделью M_m); корректировки введенной информации о машинном эксперименте; расположения информации в архивах во внешней и оперативной памяти ЭВМ.

Комплекс программ моделирования процесса функционирования объекта непосредственно осуществляет решение поставленной задачи моделирования, т. е. реализует план ведения машинных экспериментов, их организацию на ЭВМ и обработку промежуточных данных и результатов эксперимента, взаимодействие с пользователем. Для решения задачи моделирования требуются следующие программы: управления машинным экспериментом, реализации стратегии эксперимента и его диспетчеризация; машинной имитации, включая организацию вычислений и взаимосвязь модулей модели M_m ; обработки и выдачи результатов моделирования системы S в различных режимах взаимодействия с пользователем.

Комплекс программ расширения возможностей ППМ призван обеспечить пользователя средствами генерации новых программ моделирования при различных перестройках (объекта моделирования, машинного эксперимента, обработки результатов и т. п.), возникающих при решении различных задач моделирования. При этом в качестве базового языка пакета может быть выбран либо алгоритмический язык общего назначения, либо язык имитационного моделирования (*SIMSCRIPT*, *SIMULA*, *GPSS* и т. д.).

Комплекс программ организации различных режимов работы ППМ кроме основной работы по диспетчеризации процесса функционирования ППМ призван организовать его работу в режиме диалога с пользователем как на этапе ввода данных об объекте моделирования и эксперименте, корректировки БДО и БДЭ, так и непосредственно в ходе машинного эксперимента с моделью M_m использования мультимедиа технологий. Необходимо также обеспечить режим коллективного пользования пакетом, что существенно расширяет возможности и эффективность АСМ.

Обучение методам программной имитации систем. Специализированные языки и системы моделирования являются одним из базовых средств современной информационной технологии [35]. В последнее время наблюдается широкое развитие данных средств, совершенствование их возможностей и пользовательского интерфейса. Имеются программные пакеты *GSPT*, *Vissim*, *SIMEX*, *GPSS/H*, *Stella*, *SimPack*, *DOBSim*, *Mosis*, *QSIM*, *SIMPLORER*, *Modsim* и десятки других [2, 48]. Рализация систем моделирования на персональных компьютерах расширила сферу их использования в качестве

простого и эффективного средства поддержки этапа проектирования сложных объектов и систем.

Анализ рынка средств моделирования показывает, что наибольшую популярность получили транзактно-ориентированные языки класса *GPSS*. Это пакеты *GPSS/World*, *GPSS/H*, *GPSS/PC*, *MicroGRSS*. Языки данного класса хорошо согласуются с удобным описанием модели системы в форме *Q-схемы* (системы массового обслуживания).

Система моделирования *GPSS/PC*, отличающаяся возможностью установки на персональные компьютеры разного класса, была выбрана в качестве базовой для обучения методам программной имитации сложных систем. Пакет функционирует на персональных компьютерах моделей с i386 и выше, имеет более 70 типов блоков и команд, а также около 50 системных параметров. В процессе прогона имитационной модели обеспечивается возможность постоянного наблюдения в шести интерактивных графических окнах (машинная мультипликация, работа одноканальных и многоканальных устройств, функционирование блоков, отображение таблиц и матриц). Система выдает отчет о результатах моделирования, содержащий широкий спектр параметров блоков, устройств, очередей, таблиц, пользовательских списков.

В процессе лабораторного практикума и курсового проектирования обучающиеся приобретают навыки построения концептуальной модели исследуемой системы и ее формализации, алгоритмизации и машинной реализации модели, интерпретации результатов моделирования.

Широкие возможности *GPSS/PC* позволили создать на ее базе практикум по курсу «Моделирование систем», состоящий из лабораторного практикума и комплекса, поддерживающего курсовое проектирование [38, 39, 40, 51].

Особенности *GPSS*. Особенности построения и использования в процессе моделирования *GPSS* целесообразно рассмотреть на конкретном примере такого пакета, который получил широкое распространение в практике моделирования дискретных систем.

Пример 5.1. Рассмотрим пакет прикладных программ моделирования и исследования на ЭВМ дискретных систем [33, 38, 40]. Этот пакет реализован на базе языка *GPSS*. Для ПЭВМ имеется версия языка *GPSS/PC*. Основное достоинство этого пакета — наличие необходимого набора типовых элементов (устройств, накопителей, переключателей и т. п.), соответствующих компонентам реальных систем (например, формализуемых в виде *Q-схем*), и программная реализация алгоритмов их функционирования, позволяющая строить сложные модели, сохраняя привычную для исследователя или разработчика систем *S* запись.

В пакете *GPSS* для представления моделируемой системы *S* в виде машинной модели M_m используется язык блок-диаграмм. *Блок-диаграммой* в пакете *GPSS* называется графическое представление операций, происходящих в моделируемой системе *S*. В этом случае блок-диаграмма описывает взаимодействия, происходящие внутри моделируемой системы *S* в процессе ее функционирования.

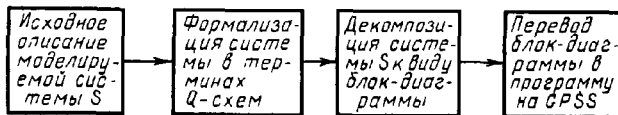


Рис. 5.5. Схема процесса создания машинной модели с использованием языка GPSS

Для пакета GPSS были введены некоторые условности и общие представления о моделируемых системах. Введение таких обобщений позволило создать язык блок-диаграмм, в котором блоки соответствуют некоторым действиям, встречающимся в системах исследуемого класса. Вводимый набор блоков для блок-диаграмм однозначно определяет наборы операторов языка, осуществляющих описание структуры моделируемой системы S , и логических правил, определяющих ее функционирование.




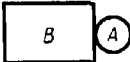
В блок-диаграммах GPSS блоки представляют собой выполняемые над динамическими объектами операции, а стрелки между блоками отражают маршруты передвижения данных объектов по системе. Альтернативные ситуации отражаются более чем одной стрелкой, выходящей из блока.





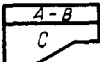
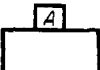


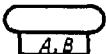


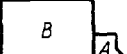
Таким образом, процесс создания модели M_M на языке блок-диаграмм GPSS сводится к декомпозиции исходной системы S до уровня элементарных операций, используемых в пакете GPSS, формированию фиксированной схемы, отражающей последовательность элементарных операций, выполняемых над динамическими объектами, и определению набора логико-вероятностных правил продвижения потоков объектов по имеющейся схеме.






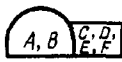
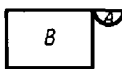

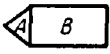

Построение блок-диаграмм GPSS предполагает знакомство программиста с набором операторов пакета GPSS. Набор операторов языка однозначно соответствует набору блоков для описания блок-диаграмм, поэтому построение блок-диаграммы не является самоцелью, а лишь промежуточным этапом при построении имитационной модели исследуемой системы S с использованием операторов пакета GPSS. При этом процесс создания машинной модели M_M можно изобразить в виде схемы, показанной на рис. 5.5.

Условные обозначения, используемые на блок-диаграммах GPSS, представлены в табл. 5.2.

Таблица 5.2

Имя и обозначение блока	Назначение
ADVANCE 	Задерживает транзакт на время $A=B$, если $B=\text{const}$, или $A \times B$, если B — функция
ASSEMBLE 	Собирает A транзактов одного ансамбля, пропускает в следующий блок первый транзакт, остальные уничтожает
ASSIGN 	Присваивает параметру A входящего транзакта значение B , модифицированное параметром C
DEPART 	Обеспечивает освобождение в очереди A B единиц

Имя и обозначение блока		Назначение
ENTER		Обеспечивает вхождение в накопитель <i>A</i> транзакта с зачетом <i>B</i> единиц памяти
GATE (X) LS LR		Проверяет условие нахождения логического ключа <i>A</i> в состоянии <i>X</i>
GATE (X) I NI U NU		Проверяет условие нахождения устройства <i>A</i> в состоянии <i>X</i>
MATCH		Синхронизирует движение транзактов по блок-диаграмме совместно с блоком MATCH с меткой <i>A</i>
PRINT		Осуществляет вывод статистики <i>C</i> в пределах от <i>A</i> до <i>B</i>
PRIORITY		Присваивает входящему транзакту приоритет <i>A</i>
QUEUE		Обеспечивает занятие в очереди <i>A B</i> единиц
RELEASE		Освобождает устройство с номером <i>A</i>
SAVEVALUE		Сохраняет заданное значение <i>B</i> в ячейке <i>A</i>
SEIZE		Занимает устройство с номером <i>A</i>
SPLIT		Генерирует <i>A</i> копий входящего транзакта и направляет их по адресу <i>B</i> . Основной транзакт переходит в следующий блок
TABULATE		Табулирует значения входящих транзактов в таблице <i>A</i>

Имя и обозначение блока	Назначение
TEST (X) E NE GE LE G L	 <p>Проверяет соотношение X между A и B и направляет входящий транзакт в следующий блок при выполнении или по адресу C при невыполнении соотношения</p>
TERMINATE	 <p>Уничтожает A транзактов</p>
TRANSFER	 <p>Изменяет направление движения транзактов согласно режиму A</p>
GATE (X) SE SF SNE SNF	 <p>Проверяет условие нахождения накопителя A в состоянии X</p>
GATHER	 <p>Собирают A транзактов одного ансамбля и пропускает их одновременно в следующий блок</p>
GENERATE	 <p>Генерирует транзакты через A единиц времени, модифицированных B, с задержкой C, D транзактов, с приоритетом E, форматом F</p>
LEAVE	 <p>Освобождает в памяти A B единиц памяти</p>
LOGIC (X) S R I	 <p>Устанавливает логический ключ A в состояние X</p>
LOOP	 <p>Осуществляет повторение A раз группы блоков от адреса B до данного блока</p>
MARK	 <p>Осуществляет отметку времени в параметре A</p>

Особенности использования ППМ *GPSS* для моделирования систем, формализуемых в виде *Q-схем*, рассмотрены в гл. 8, там же приводятся примеры программ, генерированных пакетом *GPSS*.

Создание ППМ является крупным достижением в автоматизации моделирования больших систем. Однако любой ППМ, созданный на базе ЯИМ, характеризуется некоторой узостью представлений, диктуемой особенностями языка, что является одним из слабых мест традиционных способов имитационного моделирования. В настоящее время проводится много работ по созданию систем моделирования для целых классов объектов, в основу которых положены идеи типизации способов описания структуры и динамики моделируемой системы *S*. Например, широко распространенной типовой математической схемой, используемой при моделировании организационно-производственных и информационно-вычислительных процессов, является *Q-схема*. Для создания машинных программ моделирования таких процессов могут быть привлечены также ЯИМ, как *GPSS*, *SIMSCRIPT* и т. д. Тем не менее существуют (и продолжают разрабатываться) ППМ, в основу которых положены алгоритмические языки общего назначения, позволяющие детально и адекватно описать специфику процесса функционирования определенного класса систем и создать более эффективные программы моделирования, причем такие ППМ, ориентированные на определенный класс объектов, способные успешно конкурировать с известными ЯИМ при решении задач моделирования конкретного класса систем.

Примеры дискретных, непрерывных и комбинированных ЯИМ приведены в табл. 5.3.

Таблица 5.3

Название языка	Тип	Примечание
<i>ACSL</i>	Непрерывный	Advanced Continuous Simulation Language
<i>ALSIM</i>	Дискретный	ALGOL SIMulation
<i>ANDISIM</i>	Непрерывный	ANALogue DIGital SIMulator
<i>APLIS</i>	Комбинированный	APL Simulation
<i>CDL</i>	Дискретный	Computer Design Language
<i>COSMO</i>	Непрерывный	COmpartmental System MODdedling
<i>COSL</i>	Дискретный	Control and Simulation Language
<i>CSSL</i>	Непрерывный	Continuous System Simulation Language
<i>DEMOS</i>	Дискретный	Discrete Event MOdeling in Simula
<i>DIANA</i>	Комбинированный	Digital — ANALog simulator
<i>DISCO</i>	»	DIScrete — CONTinuous
<i>DYNAMO</i>	Непрерывный	DYNAMIC MOdels
<i>FORSIN</i>	»	Fortran ORiented SIMulator
<i>GASP</i>	Комбинированный	General Activity Simulation Program
<i>GEMS</i>	Непрерывный	General Equation Modeling System
<i>GPSS</i>	Дискретный	General Purpose Systems Simulator
<i>IPSS</i>	»	Information Processing System Simulator

Название языка	Тип	Примечание
<i>МИДАС</i>	Непрерывный	Modified Integration Digital Analog Simulator
<i>SAM</i>	Непрерывный	Simulation of Analogue Methods
<i>SIMSCRIPT</i>	Дискретный	SIMulation SCRIPTure
<i>SIMULA</i>	»	SIMulation LAnguage
<i>SOL</i>	»	Simulation Oriented Language
<i>VANS</i>	»	Value Added Network Simulator

Наиболее эффективно использование ППМ при исследовании и разработке систем на основе метода машинного моделирования при реализации диалоговых процедур и концепции базы данных моделирования.

5.4. БАЗЫ ДАННЫХ МОДЕЛИРОВАНИЯ

Расширение возможностей моделирования различных классов систем S неразрывно связано с совершенствованием средств вычислительной техники и техники связи. Перспективным направлением является использование для целей моделирования иерархических многомашинных информационно-вычислительных систем и связанных с ними телекоммуникационными сетями удаленных персональных ЭВМ, работающих в режиме телеобработки.

При создании больших систем S их компоненты разрабатываются различными коллективами, которые используют средства моделирования при анализе и синтезе отдельных подсистем. При этом разработчикам необходим доступ как к коллективному, так и индивидуальным средствам моделирования, а также оперативный обмен результатами моделирования отдельных взаимодействующих подсистем. Таким образом, появляется необходимость в создании диалоговых систем моделирования коллективного пользования, для которых характерны следующие особенности: возможность одновременной работы многих пользователей, занятых разработкой одной системы S ; доступ пользователей к программно-техническим ресурсам системы моделирования, включая распределенные банки данных и пакеты прикладных программ моделирования; обеспечение диалогового режима работы с различными вычислительными машинами и устройствами, включая цифровые и аналоговые вычислительные машины, установки физического моделирования, элементы реальных систем и т. п.; диспетчирование работ в автоматизированных системах моделирования (АСМ) и оказание различных услуг пользователям, включая обучение работе с диалоговой системой моделирования; использование сетевых технологий.

Рассмотрим основные моменты связанные с разработкой распределенной базы данных моделирования (РБДМ).

Ключевые аспекты разработки баз данных. Технология баз данных (БД) относится к числу основных компьютерных технологий и представляет собой совокупность методов и средств определения и манипулирования интегрированными в базу данными [2, 14, 16]. Важной целью применения технологии БД является создание разделяемого между функционально связанными приложениями *информационного ресурса* с обеспечением независимости внешнего, логического представления БД от способов ее внутренней, физической организации в памяти компьютера. Для достижения поставленной цели технология БД использует соответствующий набор технологических инструментов.

Современное представление технологии БД определяется тем, что в основу этой технологии положено применение *реляционной модели данных* (РМД), базирующейся на строгом аппарате реляционной алгебры и математической логики. Технологические операции определения и манипулирования БД выполняются с использованием систем реляционного исчисления. Реляционный подход в целом рассматривается в качестве идеологии создания баз данных и *баз знаний* [2, 14, 52]. Такой подход является наиболее эффективным при решении многих задач моделирования сложных систем *S*.

С одной стороны, широкое применение РМД позволило разрешить одну из серьезнейших проблем достижения модельной однородности баз данных, создаваемых в средах различных *систем управления базами данных* (СУБД), поскольку практически все современные СУБД используют модели, приводимые к реляционной. С другой стороны, опора на реляционную модель существенно ограничивает возможности определения данных в БД и, тем самым, предопределяет соответствующие границы применения всей технологии БД.

Такой подход, безусловно, оправдан при проектировании БД в тех случаях, когда администратор БД владеет схемой соответствия множества данных в реляционной модели с множеством данных о реальном мире. В тоже время, интеграционные тенденции, характерные для современного этапа развития компьютеризированных технологий (в том числе и в моделировании систем), ставят на повестку дня проблему построения *интегрированных распределенных баз данных* (ИРБД), для которых обеспечение схемной однородности на основе РМД в силу целого ряда причин оказывается недостаточно. Это не означает требования революционных изменений принципов реляционного подхода при проектировании БД в условиях построения ИРБД. Это означает только то, что при определении и построении ИРБД реляционный подход должен применяться с учетом классической схемы проектирования баз данных [2, 52], согласно которой необходимо знать, каким образом был

выполнен полный цикл этапов моделирования заданной предметной области в виде реляционных схем интегрируемых БД. Очевидно, что расширение границ применения реляционного подхода, при этом, позволит проектировать новые БД уже с учетом возможности их будущей интеграции и ИРБД. Характерным примером реализации расширения реляционного подхода для разработки распределенных приложений на основе интегрированных реляционных баз данных стало создание методов и средств CASE-технологий [15].

С учетом сказанного, все основные понятия и определения технологии баз данных будут формулироваться именно с ориентацией на реализацию расширенного реляционного подхода для достижения цели методологического определения ИРБД. Полная технологическая схема определения и манипулирования интегрированными в базу данными представлена на рис. 5.6.

База данных. База данных составляет ключевое понятие технологии БД и стержневой объект управления в системах баз данных. Определение базы данных в качестве разделяемого информационного ресурса компьютеризированных технологий требует уточнения самих понятий *данные* и *информация*. Иногда база данных трактуется в качестве «подобия электронной картотеки», «хранилища для некоторого набора занесенных в компьютер файлов данных», подразумевая под термином файл «абстрактный набор данных, не обязательно совпадающий с физическим дисковым файлом». Очевидно, что при таком взгляде данные и информация рассматриваются в качестве синонимов. Как следствие, истинным становится утверждение о том, что в этом случае любые данные, извлеченные любым способом из БД, являются информацией.

Классическое определение «база данных это — данные и связи между ними» представляется более точным и уместным с учетом высказанных выше соображений. Тогда данные, извлеченные из БД на основе установленных связей, являются информацией. В противном случае извлеченные из БД данные требуют *интерпретации*. Безусловно, хранящиеся в БД фрагменты связанных данных также соответствуют понятию информации. Вне связей данные являются информацией только в том случае, если они типизированы, или классифицированы, и известна примененная классификационная схема. С учетом применения реляционного подхода связи между данными можно разделить на связи совместности (совместность атрибутивных значений табличного определения прикладного объекта) и связи соответствия (совместность атрибутивных значений межтабличного определения прикладного объекта).

Таким образом, в дальнейшем под термином *база данных* будем понимать совокупность *связанных данных*, с одной стороны, являющихся информацией, и с другой стороны, составляющих основу для получения информации, как произвольных комбинаций

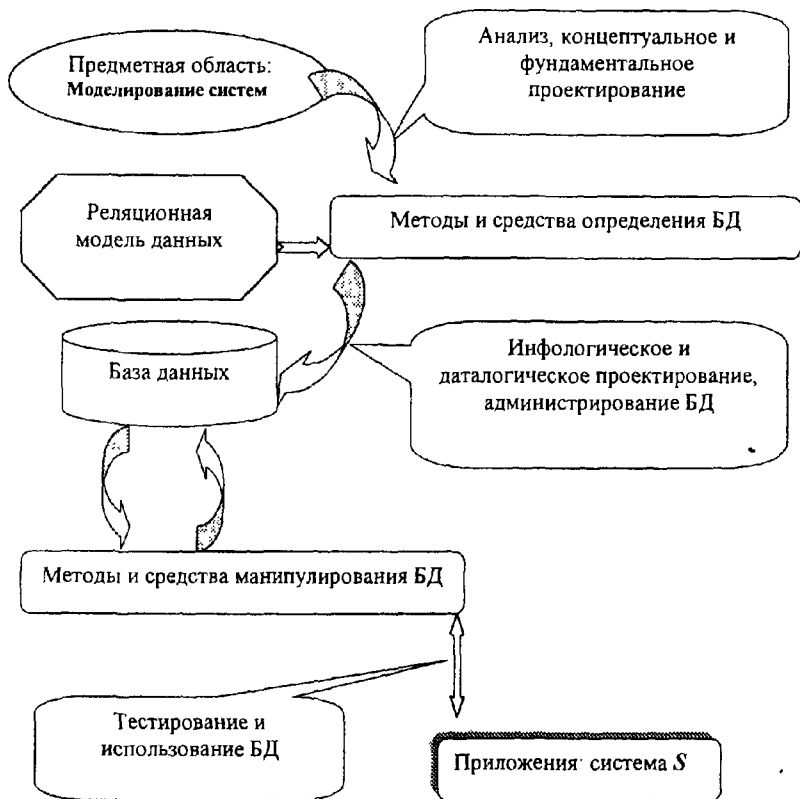


Рис. 5.6. Полная технологическая схема реализации БДМ

хранящихся связанных данных. Тогда *данные БД* и *информация по определению* оказываются синонимами.

Методы и средства определения и манипулирования БД. В технологии БД определены две основные группы механизмов определения и манипулирования БД.

К первой группе относится совокупность методов и средств определения связанных данных, включающая формальное описание структур данных, а также администрирование БД. Методы и средства определения данных реализуют ту или иную степень информативности хранящихся в базе данных в зависимости от возможностей и ограничений принятой модели данных. Определение данных выполняется статически, поскольку информативные связи между данными сохраняются и заносятся в БД наряду с собственно данными. На начальных этапах развития технологии БД именно разработка мощного языка определения данных (ЯОД)

составляла главное направление развития. Хорошо известна многолетняя деятельность рабочей группы CODASYL [2] по созданию развитого ЯОД. Однако вывести языки определения данных на уровень общих языков программирования не удалось по целому ряду причин [3].

Вторую группу составляют методы и средства манипулирования данными, реализующие информативное связывание данных в динамике, в процессе доступа в БД. На начальных этапах языки манипулирования данными (ЯМД) сводились к определению простого *CALL*-интерфейса, однако на рубеже 80-х годов тенденция развития ЯМД практически перекрыла направление разработки ЯОД. Благодаря широкому применению реляционной модели языки манипулирования смогли пройти путь становления до уровня общих языков программирования. Наиболее известным представителем семейства ЯМД на сегодняшний день является язык *SQL* (*Structured Query Language*) [2], составляющий основу и являющийся сам международным стандартом ЯМД.

У многих пользователей зачастую складывается впечатление, что независимо от типа спроектированной БД посредством языка *SQL* можно получить доступ к любой информации на основе хранящихся данных в БД. На самом деле это далеко не так. Язык *SQL* действительно обеспечивает произвольный доступ к таблицам БД в любых сочетаниях и комбинациях. Но получение информации из БД при этом ограничивается возможностями связей между данными, хранящимися в БД. При отсутствии этих связей обычной практикой является встраивание обработки связей между данными в программный код приложений. Тогда можно выделить два основных компонента манипулирования БД:

- собственно язык манипулирования как *инструмент*;
- *процедуры* связывания данных и управления извлечением информации из БД, реализованные средствами ЯМД.

Для реляционного подхода наиболее распространен процедурный способ управления извлечением информации из БД. При этом возможны три основных метода реализации этого способа:

1) модули связывания и манипулирования данными встраиваются в приложения путем программирования в профессиональных средах (*MS Visual Studio*, *C++ Builder*, *Dphi*);

2) модули связывания и манипулирования разрабатываются на языках *SQL*-серверов и хранятся непосредственно в серверной БД, становясь также разделяемыми *информационными ресурсами*;

3) модули связывания и манипулирования оформляются в виде системных динамических загружаемых библиотек *DLL*, формируя таким образом доступ в БД в виде системного *Windows*-ресурса.

Построение модулей связывания и манипулирования БД в виде разделяемых информационных ресурсов в среде *SQL*-серверов или в виде системных *DLL*-библиотек существенно при-

ближает совокупное содержание таких БД к классическому определению. Характерно, что получаемая таким образом реализация БД по полной технологической схеме рис. 5.6 остается в границах реляционного подхода.

Разновидности систем баз данных. В зависимости от способов определения и манипулирования связанными данными системы БД можно разделить на следующие основные разновидности.

Системы с файловыми базами данных в качестве БД используют простые структурированные файлы в форматах *dbf*, *bd* и др., а все информативные связи определяются и обрабатываются в приложениях, использующих такие БД. Эффективность организации структурированных файлов обычно повышается путем построения индексов и других систем указателей, что, вообще говоря, характерно при создании картотек. Индексируются, как правило, ключевые поля структур с целью ускорения доступа (за счет сортировки индексов), обеспечения уникальности значений полей, запрета на существование неопределенных значений и т. п. К числу наиболее существенных недостатков систем файловых БД (только в смысле их использования) можно отнести полную зависимость от приложений. Доступ к информации файловых БД возможен только посредством содержащего программы связи приложения. Очевидно, что как разделяемый информационный ресурс файловые БД могут существовать только в симбиозе с обеспечивающими связывание данных приложениями. Программная реализация связей на *SQL*-серверах или в виде *DLL*-библиотек естественно придает файловым БД совершенно новое качество реально разделяемого информационного ресурса.

К противоположной разновидности относятся такие системы БД, в которых все связи между данными определены как данные и хранятся в БД. Такие системы можно назвать системами с предметными базами данных. Суть названия предметная БД заключается в достижении полной независимости предметных баз данных от приложений. Предметные БД являются полноценными, самостоятельными ресурсами компьютеризированных технологий, что составляет главное преимущество их применения. В тоже время, полное определение всех связей между данными порождает существенную сложность проектирования таких БД.

Промежуточные варианты организации баз данных, при которых связи распределяются между приложениями и БД, определяют разновидность систем с прикладными базами данных. Суть названия отражает слабо или сильно выраженную ориентацию организации прикладной БД на потребности использующих ее приложений. Очевидно, что как компромиссный вариант, прикладные БД могут выступать в качестве оптимизируемого информационного ресурса компьютеризированных технологий. Прикладные базы данных находят широкое применение при моделировании на ЭВМ конкретных систем *S*.

Предметная область. Первичным источником связанных данных при проектировании баз данных любых разновидностей является соответствующая *предметная область*, которая будет рассматриваться как совокупность знаний и данных об *объектах и процессах*, подлежащих проектированию и хранению в БД. В данном случае в качестве предметной области рассматривается проблема имитационного моделирования сложных систем на ЭВМ.

Знания о предметной области, как это уже отмечалось в п. 3.1, посвященном методике разработки и машинной реализации моделей систем S , могут быть получены из разнообразных источников, таких, как:

— фундаментальные законы и устойчивые закономерности, теоретические знания прикладных наук о процессах в моделируемой системе S ;

— эвристические знания в виде принятых правил, соглашений и обозначений, характерных для систем S ;

— экспертные знания и экспертные оценки специалистов в области моделирования конкретных систем S ;

— существующие базы данных, компьютеризированные системы, технологии и проекты.

Перечисленные источники знаний о предметной области «моделирование систем» не являются источниками собственно данных, а определяют методологические принципы выделения прикладных объектов и процессов, а также методы приобретения и связывания данных о выделенных объектах и процессах. Источники знаний о предметной области определяют выбор *классификационной схемы* формализованного описания объектов и процессов, которая в процессе проектирования БД отображается в результирующей реляционной схеме.

Таким образом, *процесс проектирования базы данных* по полной технологической схеме есть процесс пошагового отображения исходной классификационной схемы предметной области в реляционную схему реализации базы данных. Выбор той или иной методологической основы процесса проектирования БД определяет вид реализованных в БД связей между данными и характеризует *интеграционные возможности* построенной БД.

Важным методологическим признаком проектируемой БД и ее интеграционных возможностей является естественная или искусственная природа выделения прикладных объектов и процессов, связанные данные о которых будут храниться в проектируемой БД. Особое значение это обстоятельство приобретает при применении *объектно-ориентированного подхода* к проектированию баз данных и построению ИРБД.

В практике моделирования сложных систем S при проектировании локальных баз данных (ЛБД) применение классификационных схем зачастую осуществляется интуитивно, с твердым убеждением в том, что выбранный способ определения данных БД естествен для

решения данной задачи. На самом деле, проектировщик ЛБД просто принимает одну из существующих классификационных схем (например, традиционную схему построения базы данных «система S — концептуальная модель M_x — машинная модель M_m » из литературы). В тоже время, отсутствие знаний о правилах формирования и связывания атрибутивных значений даже в такой БД может привести к искажению и ошибкам в трактовке извлекаемых из БД данных как информации.

Концептуальный анализ и проектирование баз данных. Выбор или конструирование классификационной схемы выделения объектов и процессов составляет суть анализа предметной области (в данном случае это «моделирование сложных систем S »). Отображение классификационной схемы в результирующую реляционную схему определяет многошаговый процесс проектирования БДМ.

Прежде всего, под формализованным представлением предметной области будет пониматься классификационное выделение объектов и процессов, что в терминах *объектно-ориентированного подхода* соответствует классификационному определению абстракций сущностей и поведения [2,16]. Классифицирование может выполняться путем классической категоризации, концептуальной кластеризации или с применением методов теории прототипов. Абстракции сущностей определяют классы объектов предметной области и выражаются через совокупности характерных свойств объектов данного класса, в тоже время, отличающих объекты данного класса от объектов других классов. Абстракции поведения выражают правила взаимодействия объектов через общие или связанные свойства.

Таким образом, выбранная или сконструированная классификационная схема с применением той или иной классификационной методологии определяет систему формирования универсума проектируемой БД. Принципиально важным для технологии баз данных является принцип отделения классификационной схемы в виде совокупности элементов моделирования представлений данных от собственно системы классифицированных данных БД. Именно реализация этого принципа способна привести к разрешению проблемы построения определенной категории объектно-ориентированных баз данных. Элементы моделирования представлений данных также включаются в состав универсума.

Проектирование концептуального представления системы классифицирования предметной области на основе выбранной классификационной схемы называется *концептуальным проектированием* предметной области, в результате которого строится *концептуальная модель* семейства концептуально однородных баз данных. Очевидно, что выбор готовых проектных решений относит проектируемую базу данных к семейству БД с однородным концептуальным представлением. Например, типизация классификации данных о модели системы S (концептуальная модель, типовая математическая

схема — см. гл. 3) практически исключает необходимость концептуального проектирования подобного фрагмента предметной области. Все базы данных, включающие описания моделей системы S , согласно принятой классификационной схеме определения математической схемы, в этом фрагменте БД концептуально однородны. Отметим, что такая классификационная схема имеет искусственную природу построения, в основе которой находится система эвристик.

Совершенно иначе обстоит дело, если проектируется фрагмент БД, посвященный описанию свойств материальных объектов (например, объектов моделирования, т. е. систем S). Классификационная схема для этого случая будет опираться на методологию классической категоризации, а знания о соответствующей предметной области будут носить фундаментальный характер. В этом случае, концептуальное проектирование окажется связанным с необходимостью проведения серьезной научно-методической работы по абстрагированию системы знаний, обычно описанной документально в словарях, справочниках, тезаурусах. Достижение концептуальной однородности, а, следовательно, и достаточных интеграционных возможностей семейства подобных баз данных становится крайне сложной проблемой. Таким образом, концептуальное моделирование предметной области включает выполнение следующих операций:

— определение природы источника знаний о предметной области (фундаментальной, эвристической, экспертной, существующего компьютеризированного решения в области моделирования конкретных систем S);

— выбор или конструирование классификационной схемы на основе классификационной методологии (классической категоризации, концептуальной кластеризации, теории прототипов);

— выделение абстракций объектов и процессов (взаимодействия объектов) через определение их свойств;

— построение структуры (иерархии) базовых классов предметной области и формализация представления такой структуры.

В результате концептуального моделирования определяются следующие концептуальные компоненты:

— совокупность типизированных абстрактных представлений сущностей предметной области, определяющих систему правил и ограничений на формирование атрибутивных значений свойств объектов;

— совокупность типизированных абстрактных представлений взаимодействия сущностей предметной области;

Классификационные правила формирования атрибутивных значений свойств объектов, а также правила взаимодействия объектов посредством их конкретных свойств образуют *концептуальную семантику* предметной области в базисе выбранной классификационной схемы. Абстракции в целом и отдельные их свойства,

выделенные на основе фундаментальных знаний о предметной области, образуют *фундаментальную семантику*, сфера действия которой выходит за границы данной предметной области. Фундаментальная семантика является основой глобальной интеграции БД по определению. Концептуальная семантика является основой построения прототипов схем индексирования и установки ограничений элементов реляционных схем БД. Другими словами, построение или владение концептуальной семантикой предопределяет способность извлечения информации из реализованной БД. В значительной мере, концептуальная семантика управляет процедурами нормализации и упорядочения реляционных схем.

Концептуальная модель предметной области является основой проектирования (и интеграции) семейства концептуально однородных БДМ. Присутствие фундаментальной семантики способно принципиально повысить интеграционные возможности соответствующих БДМ. Концептуальная модель формирует заголовочные разделы универсума, определяет *интерфейс* семейства концептуально однородных баз данных, выражает множество концептуальных компонентов (абстракций, свойств связей), на основе которых путем комбинирования можно осуществлять варианты *инфологического* проектирования баз данных.

Инфологический анализ и проектирование БДМ. Инфологическое проектирование выполняется в базе концептуальных компонентов и учитывает возможности и ограничения *модели данных* реализации БДМ. Для инфологического проектирования характерны следующие операции:

— расширение интерфейсов описания абстракций концептуально не классифицированными свойствами (например, дополнение к определению математической схемы свойства — примечание для записи произвольных дополнительных сведений о модели системы *S*); атрибутивные значения таких свойств носят характер сопутствующей информации и не могут использоваться в качестве критерия доступа к информации до проведения их концептуального моделирования, для реляционного подхода выполнение таких операций определяет процедуры инфологического проектирования связей совместности свойств объектов, т. е., прототипов таблиц;

— построение инфологических структур реализации концептуальной модели, в качестве структур могут использоваться любые структуры определения структурных абстракций [2, 16, 25]; для реляционного подхода выполнение таких операций определяет процедуры инфологического проектирования связей соответствия, т. е. прототипов межтабличных отношений.

Инфологическая модель, как интерпретация концептуальной модели, расширяет содержимое универсума. Инфологическая модель учитывает специфику проектируемой БД (например, различные моделирующие алгоритмы для реализации *Q-схем* при использовании

статистического моделирования), сохраняя концептуальную однородность семейства подобных БДМ (например, единая система обозначений для типовых математических схем — см. гл. 2). Инфологическая модель является основой определения источников, накопителей и получателей информации и определения информационных потоков. И если своеобразным аналогом элементов концептуального проектирования в существующих CASE-технологиях является моделирование сущностей-связей, то аналогом инфологического проектирования в CASE-технологиях может рассматриваться моделирование процессов прикладных областей [2, 26, 30].

Совокупность правил построения инфологической модели образует *инфологическую семантику* проектируемой БД, состоящую из определений связей совместности и соответствия. Инфологическая семантика полностью определяет пути доступа к информации данной инфологической реализации БД. Инфологическими компонентами являются логические объекты и связи, представляющие собой логические обобщения концептуальных компонентов. Образно говоря, концептуальные компоненты являются символами, а инфологически компоненты соответствуют логическим словам и выражениям, построенным с использованием концептуальных символов.

Даталогический анализ и проектирование БДМ. Даталогическое проектирование баз данных выполняется с учетом среды конкретной выбранной СУБД. Строго говоря, именно даталогическая схема БД является реляционной схемой. Даталогическое проектирование выполняется на основе принятой инфологической схемы и заключается в выполнении следующих операций:

- выделение таблиц для реализации схем логических объектов;
- определение физических форматов атрибутивного описания свойств логических объектов на основе типов соответствующих концептуальных символов;
- выделение внешних, первичных и потенциальных ключей таблиц;
- определение индексируемых полей и полей реализации логических связей между таблицами;
- проектирование представлений для организации хранимых результатов промежуточного доступа в БД;
- определение процедур обработки изменений атрибутивных значений по связям между таблицами и т. д.

Таким образом, даталогическое проектирование выполняет разложение логических слов и выражений на схемные элементы, которыми оперирует реляционная схема. Другими словами, на этапе даталогического проектирования логические выражения кодируются в терминах языка реляционной модели. Даталогическое проектирование окончательно формирует универсум.

Важнейшей процедурой на этапе даталогического проектирования считается *нормализация* реляционных схем [3, 19, 23]. Однако,

если рассматривать полный технологический цикл проектирования БД, то окажется, что нормализация есть ни что иное, как постклассифицирование представления данных предметной области в БДМ. Рассмотрим основные операции, выполняемые при нормализации схем БД.

Администрирование БДМ. Реляционная модель в значительной степени является идеалом, поскольку практически все современные СУБД работают в моделях, приводимых к реляционной. Поэтому чисто математическое моделирование и тестирование схем БДМ, как правило, неприменимо на практике. В тоже время, реализация расширенного реляционного подхода и проектирование или определение БДМ по полной технологической схеме создает необходимые предпосылки построения систем анализа и тестирования проектов БД.

Традиционное *администрирование* баз данных включает множество операций по сопровождению БД и описано в литературе [16, 52]. К числу решаемых в процессе администрирования БД задач обычно относятся:

- обеспечение физической целостности БД (разработка и реализация плана архивации физических файлов, обеспечение ведения журнала изменений, формирование точек отката);

- обеспечение безопасности данных, включая процедуры санкционирования доступа и другие средства защиты;

- обеспечение целостности, достоверности и многие другие функции администрирования.

Администрирование БДМ должно осуществляться путем организации соответствующей службы, в состав которой должны входить специалисты в области моделирования систем S , специалисты в области управления данными, системные программисты и операторы. Статус администратора БДМ должен соответствовать уровню полномочий руководителя проекта по моделированию сложного объекта, т. е. системы S , для возможности принятия легитимных решений.

Представление баз данных по полной технологической схеме расширенного реляционного подхода порождает иной взгляд на функции и состав служб администрирования БДМ. В частности, к числу задач администрирования БД можно отнести:

- администрирование фундаментальной семантики, естественно, что подобная служба должна быть организована для множества семейств фундаментально однородных БДМ, администратор фундаментальной семантики должен возглавлять научно-методическую службу ведения и актуализации фундаментальной семантики;

- администрирование концептуальной семантики для семейств концептуально однородных БДМ, администратор концептуальной семантики должен профессионально владеть классификационной схемой формализации представления предметной области, смысловым содержанием данных о классифицированных объектах и процессах;

— администрирование инфологических представлений БДМ; соответствующий администратор должен обладать знаниями профессионала прикладника и осуществлять сопровождение БДМ как разделяемого информационного ресурса компьютеризированных систем; также администратору данного уровня необходимо владеть механизмами представлений логических структур данных, знать основы примененных моделей данных;

— администрирование компьютерной реализации БДМ, на этом уровне администрирования требуются знания системного программиста и владение методами и средствами СУБД;

— администрирование физической целостности и защиты БДМ; набор задач на этом уровне администрирования традиционен.

Рассмотренные уровни администрирования баз данных, спроектированных или представленных по полной технологической схеме, формируют прообраз службы администрирования *интегрированными* БДМ. В распределенной информационной среде служба администрирования БДМ также окажется *распределенной* в соответствии с рассмотренной схемой уровней и задач администрирования баз данных.

Использование БДМ при моделировании систем. Возможности использования баз данных в качестве *источника информации*, как следует из вышеизложенного, определяются знанием модельного представления связанных данных на всех уровнях проектирования БДМ. С другой стороны, разновидность конкретной БДМ, выбранный способ классификации предметной области, наличие фундаментальной семантики, особенности концептуальной семантики и многое другое предопределяют требуемый профессиональный уровень пользователей и возможный масштаб применения такой БДМ. Использование баз данных должно быть адекватно реализованному проектному решению, и только в этом случае извлекаемые из БДМ выборки данных могут действительно рассматриваться в качестве информации для целей моделирования системы *S*.

Вопросы использования баз данных должны предусматриваться в виде стратегических целей проектирования БДМ. Особенно формулирование стратегических целей важно в условиях проектирования, построения и использования *интегрированных распределенных баз данных* при моделировании сложных систем.

Реляционная модель данных. Такая модель обладает, по меньшей мере, двумя принципиальными преимуществами по сравнению с другими моделями данных (древовидной, сетевой). Во-первых, РМД обеспечивает представление БД в виде повседневно встречающихся и привычных человеку двумерных таблиц независимо от способов компьютерной реализации БД. Таким образом, реляционная модель удобна и наглядна. Во-вторых, реляционная БД с математической точки зрения представляется конечным набором конечных отношений различной арности. Над отношениями модели можно осуществлять алгебраические операции. Можно сказать, что тем

самым теория реляционных баз данных становится областью приложений математической логики и современной алгебры и опирается на точный математический формализм. Другими словами, вторым преимуществом реляционной модели является математическая строгость ее определения.

Реляционная модель *аднородна*, поскольку все данные хранятся (представляются) в таблицах с фиксированным форматом строк, соответствующих данным об объектах и процессах предметной области. Очевидно, что однородность реляционной модели не распространяется на сущности хранимых в строках таблиц данных, конечно, если БД не была спроектирована по полной технологической схеме расширенного реляционного подхода. Математический формализм реляционной модели базируется на использовании аппарата современной реляционной алгебры.

Теоретические основы построения систем реляционного исчисления обычно скрыты за относительно простыми и понятными пользователем (разработчиком машинной модели M_m систем S) синтаксическими конструкциями формулирования запросов к БДМ. В тоже время, точное классифицирование того или иного языка манипулирования, а, следовательно, его возможностей и ограничений, особенно в условиях применения расширенного реляционного подхода, зачастую потребует учета заложенных в основу такого языка теоретических принципов и механизмов.

Объектно-ориентированный подход к БДМ. В современном представлении объектно-ориентированный подход (ОП — объектный подход) составляет основу методологий построения сложных систем. Объектно-ориентированный подход связан с представлением предметной области в виде *классов* и *объектов*, которые в зависимости от предназначения методологии могут иметь различную природу. Отображение предметной области моделирования систем S в виде совокупностей классов и объектов означает реализацию *объектной модели* представления систем, которая является ключевым понятием ОП. Таким образом, объектно-ориентированный подход, по сути, является методологией построения методологий *объектного моделирования систем*. Возможность применения ОП определяется способностью представить *предмет моделирования* (предметную область «моделирование систем S) в виде объектной модели.

На сегодняшний день сформировано несколько методологий, использующих ОП. В первую очередь, к ним относится методология объектно-ориентированного программирования (ООП), реализация которой привела к построению ряда систем ООП, таких, как *Visual C++*, *C++ Builder*, *Delphi*. Успешность и высокая результативность разработки систем ООП в значительной степени объясняется тем, что классифицируемые объекты предметных областей систем программирования по своей природе носят искусственный характер, обладают конечным числом известных свойств и хорошо

поддаются абстрагированию в соответствии с понятными классификационными схемами. Реализация принципиально новой технологии программирования с использованием систем ООП оказалась настолько эффективной и, что, зачастую, собственно объектно-ориентированный подход стал отождествляться с объектно-ориентированным программированием. На самом деле это далеко не так, хотя отсутствие хороших результатов применения ОП для реализации других методологий способствует формированию такого мнения.

Особое значение ОП приобретает при разработке методологии построения объектно-ориентированных баз данных (ООБД). В отличие от систем программирования, системы баз данных характеризуются принципиально иной, естественной природой классов и объектов предметных областей. Объектное представление какой-либо предметной области БДМ вызывает серьезные проблемы. Поэтому раздел информатики, посвященный вопросам построения ООБД, на сегодняшний день находится в начальной стадии становления, а многие работы в этой области ограничиваются уровнем предположений и гипотез. Представляется, что именно ОП может оказаться требуемым способом не только построения методологии ООБД, но и разработки методологии объектно-ориентированной интеграции распределенных баз данных (ООРБД). Очевидно, что центральной проблемой при этом становится проблема композиции объектно-ориентированного и реляционного подходов, способной привести к реализации расширенного реляционного подхода по полной технологической схеме (см. рис. 5.6).

Методология классифицирования предметной области. В основе механизмов выделения классов и объектов предметной области лежит применение выбранной классификационной схемы. К числу основных классификационных методологий относятся методология классической категоризации, методология концептуальной кластеризации и теория прототипов [16, 35].

Таким образом, говоря о классифицированных объектах и процессах предметной области, всегда будем иметь в виду выделенные классы — категории объектов, классы — кластеры объектов и классы — прототипы процессов. Для удобства, в дальнейшем все разновидности классов объектов и процессов будут называться просто классы объектов. Все классы объектов при применении объектно-ориентированного подхода определяются в виде абстракций, а представление предметной области с позиции классов объектов при применении ОП соответствует представлению предметной области «моделирование систем S » в *объектной модели*.

Основные свойства объектной модели. Рассмотрение свойств объектной модели часто выполняется с ориентацией на системы объектно-ориентированного программирования. Поэтому, некоторые, уже ставшие привычными трактовки понятий и определений

объектной модели потребуют уточнения. Например, вряд ли стоит доказывать необходимость обеспечения свойства сохраняемости для классов объектов систем объектно-ориентированных баз данных.

Абстрагирование реализует свойство концентрации внимания на внешних особенностях объектов и позволяет отделить самые существенные особенности поведения от несущественных. Минимизацию связей, когда интерфейс объектов содержит только существенные аспекты поведения, реализует принцип барьера абстракции. Таким образом, абстракция выражается через свойства, присущие всем объектам заданного абстракцией класса и, в то же время, отличающие объекты заданного ей класса от объектов других классов. Определение понятия *абстракция*, таким образом, содержит конфликтное условие, которое порождает многие классификационные проблемы. Суть такого конфликта заключается в необходимости совмещения в абстракции элементов общности и исключительности существенных свойств. Очевидно, что естественная природа выделения классов объектов предметной области обостряет указанный конфликт, доводя его до состояния неразрешимых противоречий. Именно с этим аспектом связаны главные трудности построения систем объектно-ориентированных баз данных.

Применение механизмов абстрагирования в рамках объектно-ориентированного подхода формулирует главное правило — представление предметной области в объектной модели. Классификационная схема, основанная на той или иной системе знаний о предметной области и классификационной методологии, обеспечивает выделение классов объектов. Заголовочное определение классов описывается в виде абстракций, интерфейсные свойства которых обладают качествами существенности, общности для данного класса и различимости для отделения объектов других классов.

Отметим, что совокупность классифицированных абстракций предметной области «моделирование систем S » в общем случае состоит из абстракций двух типов:

- абстракции определения объектов данных предметной области, которые называются *абстракциями данных*;
- абстракции определения модельных элементов представления данных, которые называются *абстракциями моделей* или *модельными абстракциями*.

Для систем объектно-ориентированного программирования абстракции данных определяют классы объектов программирования оконного интерфейса, работы с *Internet*, доступа в БД, связывания и внедрения объектов *ActiveX* и т. д. Абстракции моделей для систем ООП соответствуют структурным абстракциям. Для обоих типов абстракций систем ООП характерным является то, что определяемые абстракциями объекты выступают одновременно и в роли данных, и в роли инструментов управления такими данными. На-

пример, стандартное окно системы *Windows* является объектом соответствующих классов (кнопка, панель, окно редактирования и т. п.).

Инкапсуляция реализует свойство объектной модели, связанное с абстрагированием и обеспечивающее разделение описания класса на *интерфейс* и *реализацию*. В интерфейсной части описания класса содержится суть определения классифицированной абстракции, т. е. то, что присуще всем объектам задаваемого абстракцией класса. Через интерфейс объекты взаимодействуют, и именно интерфейс является предметом интеграционной отладки при разработке проектируемой системы. Таким образом, интерфейс представляет собой формальное описание абстракции средствами реализации проектируемой системы. Реализация класса точно соответствует названию, и содержит детали реализации интерфейса (т. е. абстракции) при построении объектов данного класса.

Разделение на интерфейс и реализацию соответствует определению внешнего (логического) и внутреннего (физического) устройства объектов. Принцип инкапсуляции соответствует сути вещей, интерфейс при этом играет роль определения степени интереса и полезности знания такой сути. Объединение инкапсуляции с абстрагированием обеспечивает формирование видимости единой сущности множества физически различных объектов. Так, в технологии БД, табличная форма представления данных является интерфейсом, в то время как реализация таблиц в разных СУБД физически различна (начиная с различий форматов файлов — *db*, *dbf* и др.).

Интерфейс и реализация исполняются в среде разработки проектируемой системы, поэтому можно построить следующую логическую цепочку взаимосвязи понятий: *интерфейс* есть исполнение абстракции, а *реализация* есть исполнение интерфейса. Например, в системах ООП интерфейс и реализация исполняются на языках *C++*, *Object Pascal* и др. Для систем объектно-ориентированных БД смысл инкапсуляции, в первую очередь, связан с разделением модельных представлений абстракций данных и реализации объектов данных одного классификационного раздела в различном схемном исполнении. Логическая последовательность взаимосвязи понятий в этом случае следующая.

Модельные абстрагирование и инкапсуляция исполняют концептуальные, фундаментальные, инфологические и даталогические представления БД в реляционной модели, а абстрагирование и инкапсуляция данных предметной области исполняют собственно реализацию объектно-ориентированной базы данных с применением расширенного реляционного подхода. Таким образом, объектная модель ООБД многослойна и включает иерархию объектных представлений моделей и данных с реализацией всех уровней объектной модели БД в реляционной модели данных

Иерархия реализует свойство объектной модели упорядочения и расположения по уровням выделенных абстракций предметной области. Основным свойством и преимуществом иерархической организации объектной модели является наследование свойств и других элементов определения объектов по схеме «родитель-потомок», при этом наследование касается только интерфейсных определений. О наследовании говорят, как об иерархии «обобщение-специализация», что в значительной степени напоминает определение родовидовых зависимостей. И если в системах объектно-ориентированного программирования иерархии классов относительно просты и понятны (например, иерархии классов библиотеки *MFC* языка *Visual C++*, или библиотеки *VCL* системы *Delphi*), то для систем объектно-ориентированных баз данных иерархии классов оказываются многомерными.

Одной из главных целей применения иерархической организации объектной модели является обеспечение возможностей синтеза и декомпозиции сложных объектов через простые объекты более низких уровней. Преимущество такого подхода очевидно, и особенно важным представляется реализация свойства иерархии в ООБД. Даже интуитивно понятия синтеза, декомпозиции, зависимостей ассоциируются с аспектами рассмотрения теории реляционных баз данных и модельными представлениями БД по полной технологической схеме. Свойство иерархии позволяет реализовать протокольный принцип организации информационной среды, что особенно важно при построении интегрированных РБД.

Свойство типизации дополняет свойства абстрагирования, инкапсуляции и иерархии определением типов сущностей классифицированных объектов предметной области. Типизация трактуется двояко: помимо типизации сущности объектов данного класса, это свойство обеспечивает защиту при использовании объектов, определенного рода различимость объектов разных классов. В системах ООП типизация осуществляется посредством объявления традиционных типов данных (*integer*, *float*, *character* и др.). Для объектно-ориентированных БД многомерность объектной модели порождает существование различных видов типизации объектов модельных классов и классов данных. Так, типизация объектов даталогических моделей похожа на типизацию языков программирования (типы атрибутов в столбцах таблиц соответствуют языковым типам данных). Типизация объектов инфологических, концептуальных и фундаментальных моделей, а, также объектов данных существенно иная. Различают сильную и слабую типизацию, что в первую очередь связано с возможностями преобразования типов и объявлением объектов множественных типов (полиморфизм). Для объектно-ориентированных БД преобразования модельных типов будет играть ключевую роль при решении интеграционных задач.

Свойство модульности достаточно очевидно и должно быть

реализовано в любой программной разработке. Модульность реализует абстрагирование не на уровне классов объектов, а на уровне программных единиц разрабатываемой системы. В этом смысле модуль можно уподобить классу с описанием интерфейса и реализации, содержащему один объект в виде программной единицы (*unit* в языке *Object Pascal*). При разработке модулей следует придерживаться правила их типизации (модуль должен управлять однотипными объектами), что зачастую приводит к реализации модулей в виде объектов какого-то класса (например, диалоговые модули «открыть файл», «сохранить файл», «открыть графический файл» и другие реализованы в системе *Delphi* в виде объектных компонентов). Модульная организация системы является необходимым требованием, а сама суть модульности тесно переплетена с сутью других свойств объектной модели. Для объектно-ориентированных БД свойство модульности должно быть обеспечено на уровне манипулирования объектами БД (например, модульный принцип построения программ *SQL*-сервера на языке *SQL*).

Рассмотрим свойства параллелизма и сохраняемости. Для систем ООБД параллелизм особенно важен при построении или интеграции баз данных по полной технологической схеме. В любом случае БД в качестве разделяемого *информационного ресурса* должна функционировать в архитектуре «клиент-сервер», даже в локальном исполнении. Сохраняемость естественна для объектов данных БД. В то же время, расширенный реляционный подход распространяет свойство сохраняемости не только на объекты данных, но и на объекты модельных классов, и это является принципиальной особенностью реализации полной технологической схемы построения БД в объектной модели.

Рассмотренные свойства объектной модели иллюстрирует рис. 5.7. В системах объектно-ориентированного программирования иерархия выделенных в предметной области классов обычно реализуется в виде библиотек базовых классов (например, библиотека *MFC*). Для систем ООБД реализация должна быть несколько иной. Иерархия классов данных выражает информативное связывание данных. Собственно данные организуются в табличном формате, связи реализуются либо в виде данных, либо в виде процедур связывания данных. *Модельные абстракции* должны исполняться либо в виде определения данных, либо путем построения библиотек базовых классов в среде *SQL*-серверов или в составе *DLL*-библиотек.

Инфологическое проектирование баз данных. К числу основных задач этапа инфологического проектирования баз данных (рис. 5.8) относятся следующие три задачи:

1. Проектирование логических объектов.
2. Проектирование логических структур данных.
3. Проектирование информационных связей (поток) между логическими объектами.

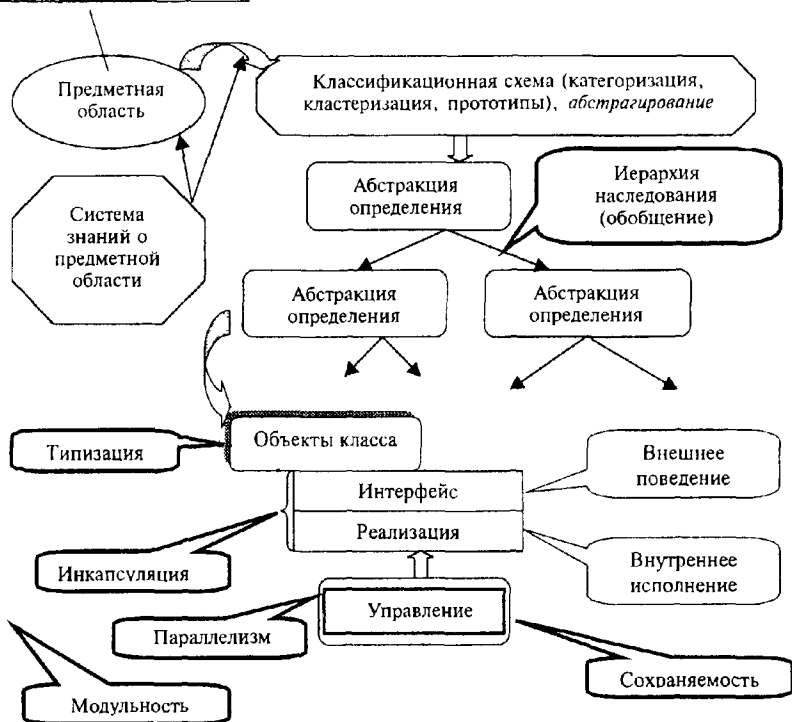


Рис. 5.7. Свойства объектной модели БДМ системы S

Для решения поставленных задач в качестве исходных условий для выполнения инфологического проектирования базы данных требуется обеспечить:

- реализацию концептуальной модели БД;
- выбор модели данных для реализации БД.

Модель и логические структуры данных. Существует множество типовых структур организации данных [2]. На сегодняшнем этапе развития компьютерных технологий типовые структуры важно рассматривать, с одной стороны, как основу построения абстракций при создании библиотек базовых классов объектно-ориентированных систем программирования, с другой стороны, как основу объектно-ориентированной ориентации моделирования баз данных.

Инфологическая модель БДМ. Инфологическая модель базы данных с ориентацией на ее реализацию в РМД представлена на рис. 5.9.

Логические объекты выделяются из концептуального представления базы данных на основе принятой классификационной схемы

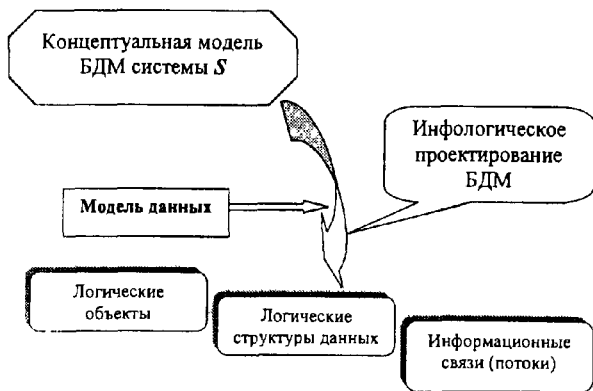


Рис. 5.8. Инфологическое проектирование БДМ

БД. Логические объекты порождаются на основе идентификационных совокупностей свойств концептуальных абстракций объектов и абстракций концептуальных связей. С точки зрения проектировщика базы данных, логические объекты соответствуют объектам прикладных данных, понятны по своему определению (например, абстракция «пользователь компьютерной системы» логически разделима на объекты «пользователь» и «компьютерная система» при условии соответственно построенной концептуальной модели). В любом случае, выделение логических объектов для каждой идентифицирующей группы свойств концептуальной модели БД является важным методологическим принципом инфологического проектирования БДМ.

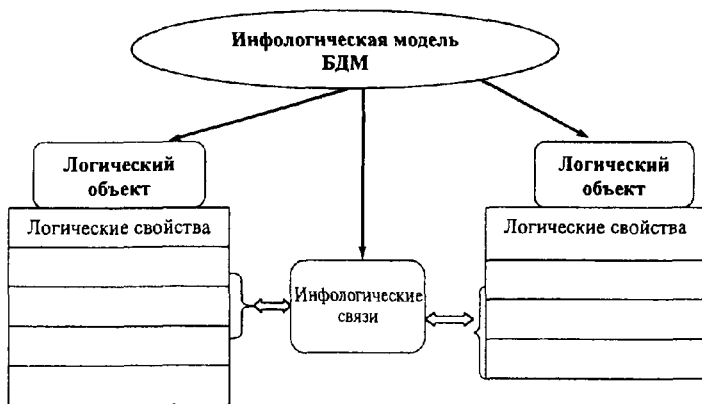


Рис. 5.9. Инфологическая модель БДМ с ориентацией на ее реализацию в РМД

Интеграция распределенных БДМ. В практике моделирования сложных систем приходится иметь дело с многомашинными комплексами и сетевыми структурами, что требует решить проблему интеграции БД. Основные понятия и определения формируемой интеграционной методологии представлены на рис. 5.10.

Сформированная таким образом архитектура интеграционной методологии позволяет сформулировать и определить ключевые понятия и принципы построения объектов компьютерных технологий класса *интегрированных распределенных баз данных (ИРБД)* [41, 52, 54].

Классы баз данных, распределенных баз данных и интегрированных распределенных баз данных образуют иерархию «обобщение-специализация», поэтому для определения интегрированной РБД можно использовать наследование существенных свойств объектов порождающих классов.



Рис. 5.10. Основные понятия и определения интеграционной методологии

Любая база данных определяется через совокупность связанных (интегрированных в базу) данных и является разделяемым информационным ресурсом компьютеризированных технологий. Тогда любая распределенная база данных также является разделяемым информационным ресурсом (совокупностью локальных информационных ресурсов) в виде связанных *распределенных данных*. Очевидно, что для определения понятия «интегрированная распределенная база данных» в первую очередь необходимо уточнить суть понятия «распределенные данные».

Распределенные БДМ. В обобщенном виде, классификационное определение распределенной базы данных следует из определения БД и формулируется в виде: распределенная база данных это распределенные данные и связи между ними. Локальная связанность распределенных данных присуща определениям локальных БДМ. Для распределенной БДМ ключевым объектом определения становятся связи между распределенными данными, т. е. теперь необходимо сформулировать условия, при которых распределенная информационная среда приобретает статус распределенной базы данных.

Одним из главных преимуществ применения реляционной модели является обеспечение однородности табличного представления любых БД. В терминах расширенного реляционного подхода такая однородность трактуется как даталогическая однородность реляционных БД. Именно это обстоятельство сопровождается сегодня формированием стандарта доступа в БД на основе языка *SQL*. Специфика СУБД, проявляющаяся, в том числе, и в существовании диалектов языка *SQL* нивелируется путем применения средств логического соединения с БД посредством механизма псевдонимов.

Реляционная модель обеспечивает однородность представления распределенных данных. Механизм псевдонимов обеспечивает однородность среды *манипулирования* распределенными данными (например, система драйверов БД *DAO* использует приложение *Microsoft Jet Database Engine* для однородного манипулирования базами данных *Microsoft Access*, а также *FoxPro* и *Excel*). В совокупности, таким образом, реализуется однородная среда определения и манипулирования распределенными данными. Назовем представление такой среды схемным модельным уровнем интегрированного представления совокупности распределенных распределенных данных, учитывая общепринятое понимание реализации реляционных БД в виде реляционных схем.

Тогда можно сказать, что реализация представления совокупности (интегрированного представления) распределенных данных на уровне схемной модели обеспечивает удовлетворение представленной таким образом совокупности распределенных данных требованиям реляционной модели. Совокупность распределенных данных становится при этом реляционной БД, и именно такая

реляционная БД и будет называться *распределенной базой данных*. Учитывая интеграционный характер объединения распределенных данных в совокупность, схемную однородность РБД будем называть *схемной интеграцией*. Тогда определение распределенной БД формулируется следующим образом: распределенная база данных — это, по меньшей мере, схемно интегрированная совокупность распределенных данных.

По сути дела, такое определение распределенной БДМ означает построение совокупности локальных информационных ресурсов в виде однородной среды распределенных данных, причем однородность рассматривается с точки зрения определения и манипулирования распределенными данными в границах реляционного подхода, или на уровне даталогических схем локальных элементов БДМ. Обобщенная даталогическая схема РБД в принципе доступна для выполнения стандартных операций обобщенной нормализации, однако, при этом необходимо учитывать цели построения РБД, которую далеко не всегда целесообразно и возможно сформировать в виде единой реляционной БДМ.

Расширение реляционного подхода предусматривает представление баз данных по полной технологической схеме, т. е. включая формализованные описания инфологической, концептуальной и фундаментальной моделей БД. Распределенные объекты данных на этих уровнях представления разнородны по определению, содержательно, а не технически, в силу их построения в интересах локальных систем и приложений. Определим инфологический, концептуальный и фундаментальный уровни семантическими уровнями представления РБД. И если для локальных баз данных семантическая интеграция выполняется явно или интуитивно проектировщиком БДМ, то для распределенных БДМ достижение семантической однородности становится существенной проблемой. Декларируя факт возможности достижения семантической однородности распределенных БДМ, можно сформулировать следующее определение: интегрированная распределенная база данных — это семантически интегрированная распределенная база данных.

Другими словами, при интеграции РБД подразумевается семантическая интеграция, достижимая в рамках расширенного реляционного подхода. Схемная интеграция РБД однозначна (табличное представление распределенных данных независимо от специфики их реализации). Семантическая интеграция многозначна и допускает множество вариантов реализации. Это означает, что определение класса интегрированных РБД постулирует существование разновидностей ИРБД, объединенных существенным свойством семантической однородности (логической, концептуальной или фундаментальной) [2, 16].

5.5. ГИБРИДНЫЕ МОДЕЛИРУЮЩИЕ КОМПЛЕКСЫ

В практике машинного моделирования сложных систем используется вычислительная техника трех типов: ЭВМ, АВМ и ГВК. При этом ГВК, обеспечение которых ориентировано на решение задач машинного моделирования (например, по составу программного обеспечения, наличию операционной системы реального времени и диалога, интерфейсу с натурными блоками моделируемой системы S и т. д.), называются *гибридными* или *аналого-цифровыми моделирующими комплексами* (АЦМК). Преимущества каждого типа вычислительных средств в первую очередь определяются спецификой основных свойств цифровых и аналоговых ЭВМ, используемых для моделирования конкретной системы S .

Рассмотрим достоинства и недостатки этих трех типов вычислительных средств (АВМ, ЭВМ и ГВК) применительно к машинному моделированию систем [52]. В общем случае с любой задачей, которую решает АВМ, может справиться и достаточно мощная универсальная ЭВМ. Но на АВМ можно решать задачи моделирования систем быстрее и эффективнее.

Основные черты, характерные для АВМ:

1) зависимые переменные модели системы S представляются *в непрерывном виде*;

2) *точность результатов* моделирования определяется качеством компонентов электрических схем АВМ;

3) возможно *одновременное выполнение* параллельных вычислительных операций, что особенно важно при моделировании сложных систем;

4) возможно выполнение операций *в реальном* или *ускоренном масштабе времени* (скорость вычислений ограничена главным образом частотными характеристиками элементов, а не сложностью решаемой задачи моделирования системы S);

5) операции сложения, вычитания, умножения, дифференцирования, интегрирования, генерирования непрерывных функций выполняются весьма эффективно, но имеются ограниченные возможности выполнения логических действий, накопления цифровых данных, обеспечения длительных задержек, обработки информации, которые весьма характерны для моделирования систем;

6) технология программирования состоит в основном в замещении элементами АВМ (такими, как операционные усилители, интеграторы и т. п.) соответствующих элементов моделируемой системы S ;

7) к АВМ можно подключить блоки реальной системы S при комбинированном моделировании; пользователь имеет возможность в ходе машинного эксперимента на АВМ изменять значения установок, т. е. коэффициентов, устанавливаемых на АВМ, что обеспечивает более наглядное проведение эксперимента с моделью системы S .

Характерные черты ЭВМ:

1) вся обработка промежуточной и результирующей информации в процессе моделирования системы S реализуется в *дискретном виде*;

2) все операции по работе с машинной моделью M_m выполняются *последовательно*;

3) *точность результатов* моделирования системы S определяется главным образом выбранными численными методами решения задачи и формой представления чисел;

4) *время решения* определяется сложностью задачи моделирования системы S , т. е. числом операций, необходимых для получения результатов моделирования;

5) наличие компромисса между временем решения и точностью результатов моделирования системы S ;

6) применяется ограниченное число арифметических операций (сложение, вычитание, умножение и деление), но с помощью численных методов можно в модели на базе этих исходных операций реализовать и более сложные, например дифференцирование, интегрирование и т. д.;

7) для выполнения логических операций и принятия решений в процессе моделирования используются как цифровые, так и нецифровые данные;

8) предусматриваются операции с плавающей запятой, что устраняет трудности масштабирования модели;

9) методы программирования базируются как на ЯОН (часто не имеющих непосредственного отношения к задаче моделирования), так и на ЯИМ.

Современные ГВК представляют собой попытку объединить все лучшее, присущее цифровой и аналоговой технике, и избежать их недостатков. Некоторые задачи требуют для своего решения усиления цифровой части комплекса аналоговой частью для увеличения скорости вычислений и распараллеливания процессов. При этом *цифровая часть* ГВК дает возможность:

1) управлять аналоговой частью машинной модели M_m при высоком быстродействии;

2) использовать устройства запоминания и хранения данных моделирования;

3) обеспечивать более высокую точность вычислений и применения логических операций при моделировании системы S .

Преимущества ГВК:

1) сочетает быстродействие АВМ и точность ЭВМ, что позволяет расширить класс моделируемых объектов;

2) в процессе машинного моделирования позволяет использовать реальные технические средства и части исследуемой конкретной системы S ;

3) обеспечивает гибкость аналогового моделирования благодаря использованию логики и памяти ЭВМ;

4) увеличивает быстродействие ЭВМ за счет использования аналоговых подпрограмм;

5) делает возможной обработку входной информации о модели системы S , представленной частично в дискретной и непрерывной формах.

Говорить о преимуществах и недостатках ГВК можно применительно к машинному моделированию конкретного класса систем S . Для некоторых объектов использование при реализации модели системы ГВК аналогично их практической реализуемости.

В зависимости от специфики исследуемых объектов в ряде случаев эффективной оказывается ориентация при моделировании систем на ЭВМ. При этом надо иметь в виду, что АВМ значительно уступают ЭВМ по точности и логическим возможностям, но по быстродействию, схемной простоте, сопрягаемости с датчиками внешней информации превосходят или, по крайней мере, не уступают им.

Для сложных динамических объектов перспективным является моделирование на базе ГВК, которые реализуют преимущества цифрового и аналогового моделирования и позволяют наиболее эффективно использовать ресурсы ЭВМ и АВМ в составе единого комплекса. При использовании ГВК существенно упрощаются вопросы взаимодействия с датчиками, установленными на реальных объектах, что позволяет, в свою очередь, проводить комбинированное моделирование с использованием аналого-цифровой части модели и натурной части объекта. Такие гибридные моделирующие комплексы могут входить в состав многомашинного информационно-вычислительного комплекса коллективного пользования, что еще больше расширяет его возможности с точки зрения моделируемых классов больших систем.

Состав и структура технического обеспечения АЦМК определяется множеством задач, на решение которых он ориентирован. В общем виде структура технических средств представлена на рис. 5.11. Здесь приняты следующие обозначения: *АВМ* — аналоговая вычислительная машина; *ЭВМ* — цифровая электронная вычислительная машина; *АЦП* — аналого-цифровой преобразователь; *ЦАП* — цифро-аналоговый преобразователь; *БУС* — блок управляющих связей; *РА* — реальная аппаратура; *Поп* — пульт оператора.

Возможны различные варианты построения многомашинных комплексов, в которых используется по несколько АВМ и ЭВМ. Такие варианты обычно выбираются в случаях, когда не хватает производительности одного вычислителя или есть необходимость разделить средства выполнения отдельных задач моделирова-

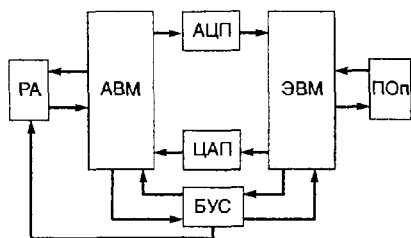


Рис. 5.11. Структура технических средств аналого-цифрового моделирующего комплекса

АЦМК в процессе решения задачи моделирования, как правило, осуществляются ЭВМ. Для реализации этих функций применяются специальные управляющие шины и аппаратура стыковки АВМ и ЭВМ по управлению, которые объединены на рассматриваемой схеме в БУС. Наряду с цифровой и аналоговой частями модели исследования на АЦМК могут использоваться реальные элементы исследуемой системы S . Исследования такого типа называются *полунатурным моделированием*.

Оператор управляет процессом моделирования с помощью средств, номенклатура которых определяется задачами, решаемыми на АЦМК. В состав ПОП могут входить печатающие устройства различного типа, дисплеи, графопостроители, самописцы и т. д., может иметь место специализированная клавиатура для передачи управляющих команд типа «Запуск», «Останов» и т. п. Таким образом, ПОП в АЦМК представляет собой набор технических средств для организации диалога «оператор — машинный эксперимент».

При распределении задачи моделирования системы S по средствам, входящим в состав АЦМК, могут быть выделены три типа комплексов.

Аналого-ориентированные комплексы используются в тех случаях, когда не требуется высокая точность результатов и когда моделируемая система S реализуема аналоговыми средствами. Системы такого класса исследуются на АЦМК, в которых цифровые средства необходимы на этапе подготовки модели для автоматизации набора задачи, накопления и обработки результатов моделирования. Сама же модель системы S реализуется исключительно на аналоговом вычислителе (аналоговое моделирование). Наряду с указанными функциями ЭВМ может выполнять задачи управления АВМ в процессе реализации модели. АЦМК с цифровым управлением и цифровой логикой способны воспроизводить более сложные модели по сравнению со стандартными АВМ. К аналого-ориентированным АЦМК относятся также комплексы, в которых ЦВМ применяются в качестве периферийного

ния системы S из-за ее функциональных или структурных особенностей.

Преобразователи АЦП и ЦАП являются средствами организации информационных связей между АВМ и ЭВМ, т. е. средствами для обмена информацией между цифровой и аналоговой частями модели системы S .

Подготовка, запуск, останов и синхронизация элементов

оборудования. В таких АЦМК малая ЭВМ используется с мощной АВМ для решения 170 специальных задач моделирования, решение которых было бы трудно или невозможно с помощью аналоговой аппаратуры.

К цифро-ориентированным комплексам можно отнести универсальные ЭВМ, где для отображения и регистрации результатов используются аналоговые средства — осциллографы, самописцы и т. д. В таких АЦМК модель M_x полностью реализуется цифровыми методами. Возможны варианты построения АЦМК для полунатурного моделирования, когда реальная аппаратура стыкуется с ЭВМ через аналоговый вычислитель. В цифро-ориентированных АЦМК может иметь место распараллеливание отдельных вычислительных процедур в процессе работы с цифровой моделью M_x за счет реализации их аналоговыми средствами.

Сбалансированные (универсальные) комплексы являются самым мощным средством для решения задач аналого-цифрового моделирования. В их состав входят средства, с помощью которых могут эффективно решаться не только аналого-цифровые задачи, но и задачи аналоговые с цифровым управлением, а также задачи цифрового моделирования. На комплексах такого типа широко используется диалог «оператор — машинный эксперимент», т. е. могут запоминаться, отображаться и регистрироваться результаты решений, оперативно вноситься изменения в модель M_x и осуществляться ее запуск. Другими словами, имеется возможность реализовать итеративный процесс исследования, сходящийся к получению искомого результата, что особенно важно при автоматизации проектирования системы S на базе машинного моделирования.

Задача построения технического обеспечения АЦМК в настоящее время сводится к выбору стандартной аппаратуры, разработке информационный и управляющих связей, реализуемых программно. Такой подход стал возможным благодаря тому, что сейчас промышленность выпускает широкий перечень ЭВМ, в которых предусмотрена возможность неавтономной работы. Это условие важно для построения АЦМК, так как в противном случае необходимы доработки универсальных вычислителей, создание нестандартного оборудования, что, как правило, делать нежелательно. При создании АЦМК должна быть возможность стыковки с периферийными устройствами широких функциональных возможностей: аналого-цифровые и цифро-аналоговые преобразователи, коммутаторы, регистры и т. д.

Современные АВМ, как правило, позволяют осуществлять цифровое управление. Принципиальных трудностей в построении технического обеспечения АЦМК нет. Однако, несмотря на широкие возможности, открывающиеся с выпуском большой номенклатуры цифровых и аналоговых вычислительных устройств, задача выбора

комплекса технических средств АЦМК представляет собой сложную проблему, при решении которой необходимо ответить на следующие вопросы [36].

Прежде всего нужно обосновать преимущества гибридного моделирования системы перед аналоговым или цифровым. При этом задачи, для которых проектируется АЦМК, должны быть достаточно важными, чтобы оправдать затраты на его создание. Решая вопрос о том, должен ли комплекс быть аналого-ориентированным, цифро-ориентированным или сбалансированным, необходимо провести выбор ЭВМ средней, малой или большой мощности. При достаточно высоких требованиях к скорости реализации цифровой части модели системы можно пойти по пути создания многопроцессорного комплекса. Необходимо рассмотреть требования к архитектурным особенностям ЭВМ: длине слова, возможностям системы прерывания, наличию аппаратных средств для работы с плавающей запятой, организации памяти и т. д. При выборе АВМ необходимо учитывать полосу пропускания, эффективность управления от ЭВМ, возможности автоматического набора, точностные характеристики.

Следует ответить на вопросы: требуется ли работа в реальном масштабе времени, какие устройства должны быть включены в гибридный вычислительный контур, какие функции по управлению должны быть возложены на ЭВМ. При определении технологии проведения исследований на АЦМК выбирается номенклатура устройств отображения и регистрации, средств ведения диалога, находятся конфигурация системы связи, алгоритмы обмена и синхронизации работы отдельных устройств. Важными моментами при построении АЦМК являются выбор АЦП и ЦАП, количество каналов информационных связей, требования к точности и быстродействию.

Сложность перечисленных вопросов заключается в том, что большинство из них взаимосвязаны. От правильности их решения зависит эффективность моделирования систем на АЦМК, точность и достоверность результатов моделирования конкретной системы *S*.

Процедура компоновки технического обеспечения АЦМК представляет собой достаточно сложный неформальный процесс, в котором качество созданного комплекса в значительной степени зависит от интуиции, опыта и способностей его разработчиков. Данная процедура включает в себя этап логической, конструктивной и электрической компоновки.

Логическая компоновка подразумевает выбор минимального состава устройств, агрегатов и модулей из номенклатуры определенных семейств вычислительной техники, обеспечивающих выполнение функциональных задач, стоящих перед АЦМК, а также объединение их в единый комплекс, работающий под управлением общего программного обеспечения. При конструктивной компоновке реша-

ются вопросы размещения устройств в типовых конструктивных элементах. Электрическая компоновка предполагает выбор линий связи для конструктивно скомпонованных элементов и порядок их соединений. Исходным материалом для логической компоновки являются агрегаты и модули технического обеспечения.

Опишем вариант построения АЦМК для решения задачи моделирования системы S на базе управляющей ЭВМ, которая имеет развитый интерфейс «Общая шина» (ОШ), т. е. на них могут быть возложены функции управления. Возможности расширения памяти и быстродействия позволяют достаточно эффективно использовать управляющие ЭВМ для реализации расчетных процедур в процессе моделирования различных систем. Мощная операционная система этих ЭВМ позволяет на базе стандартных средств реализовать процедуры генерации модели M_n на этапе ее подготовки [12].

Рассмотрим особенности компоновки устройств организации информационных и управляющих связей между ЭВМ и АВМ. Вариант использования ЭВМ в качестве цифровой части АЦМК состоит в применении для этих целей измерительно-вычислительных комплексов (ИВК). Развитие персональных и микроЭВМ, модульных программно-аппаратных средств, построенных на их базе для целей автоматизации научных исследований, создало благоприятные технико-экономические условия для компоновки модульных информационно-измерительных систем и ЭВМ в единые ИВК.

В общем случае ИВК — это автоматизированное средство измерения и обработки информации, предназначенное для исследования, контроля и испытаний сложных объектов и представляющее собой совокупность программно-управляемых технических средств, имеющих блочно-модульную структуру, определенную организацию и связи, обеспечивающие получение, преобразование, накопление, обработку и выдачу измерительной, командной и другой информации в соответствующей форме, в том числе для воздействия на объект исследования. При построении АЦМК на базе ИВК реальная аппаратура и АВМ выступают в качестве управляемого исследуемого объекта.

Пример 5.2. Рассмотрим возможности построения технического обеспечения АЦМК на базе ИВК (рис. 5.12). В данной структуре через интерфейс ОШ объединены следующие устройства: П — процессор, ОЗУ — оперативное запоминающее устройство, Д — дисплей, ПР — принтер, Н — накопители. Совокупность данных устройств представляет собой вычислительный комплекс. Связь с АВМ осуществляется через два крейта: устройство управления — КК1 — контролер крейта № 1, АЦП — аналого-цифровой преобразователь, К — коммутатор, ЦАП1 и ЦАП2 — цифро-аналоговые преобразователи, РВИС — регистр ввода инициативных сигналов. В данном варианте стыковки с АВМ аппаратура крейта № 2 — КК2, входящего в состав ИВК, не используется, за исключением ЦАП2, перенесенного в крейт № 1. Управление процессом моделирования на АЦМК осуществляется с пульта управления ПУ.

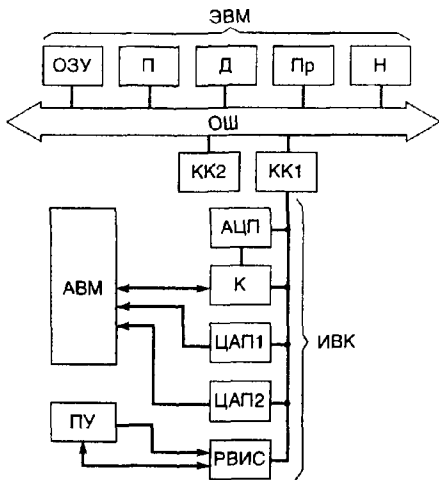


Рис. 5.12. Структура технического обеспечения АЦМК на базе ИВК

Программное обеспечение (ПО) АЦМК (рис. 5.13) строится по модульному принципу и включает в себя комплексы программ: планирования машинных экспериментов, построения модели системы, проведения машинных экспериментов, обработки результатов моделирования. Наиболее перспективной формой реализации ПО АЦМК является построение его в виде набора пакетов прикладных программ (ППП), снабженных развитыми средствами генерации, модификации и расширения.

Функциональное разбиение ПО может быть произведено с привязкой его к составным частям АЦМК. В этом случае в ПО включаются ППП пользователей, ПО аппаратуры КАМАК, ППП управления АВМ, проблемно-ориентированное ПО ЭВМ.

Пример 5.3. Рассмотрим состав ПО АЦМК. В зависимости от вида модели конкретной системы S ППП пользователя включает в себя следующие программы:

STATE — программа, содержащая соотношения между переменными и параметрами и задающая описание непрерывной части модели системы;

SCOND — программа слежения за значениями дискретных переменных и определения наступления событий в модели системы;

SSAVE — программа сбора информации о состояниях моделируемой системы;

EVENTS — программа реализации событий, содержащая алгоритмы обработки событий в модели системы;

USINT — программа определения начальных значений переменных модели (может также переопределять значения переменных в процессе моделирования);

USOUT — программа вывода результатов моделирования, не предусмотренных системой моделирования;

PLNEX — ППП, позволяющий планировать эксперимент, собирать и обрабатывать статистическую информацию о результатах моделирования системы. Решая

Развитая операционная система современных ЭВМ позволяет реализовать на их основе процедуры генерации цифровой части модели M_c , организовать диалоговые режимы на этапе подготовки машинного эксперимента и обработки его результатов. В то же время свойства ИВК как управляющего комплекса дают возможность управлять процессом моделирования системы S , осуществлять сбор, накопление и обработку результатов моделирования.

При рассмотрении приведенных примеров структур комплексов (рис. 5.11 и 5.12) не ставилась цель показать процесс проектирования технического обеспечения АЦМК. Данные структуры приводились как возможные варианты реализации АЦМК для моделирования систем.

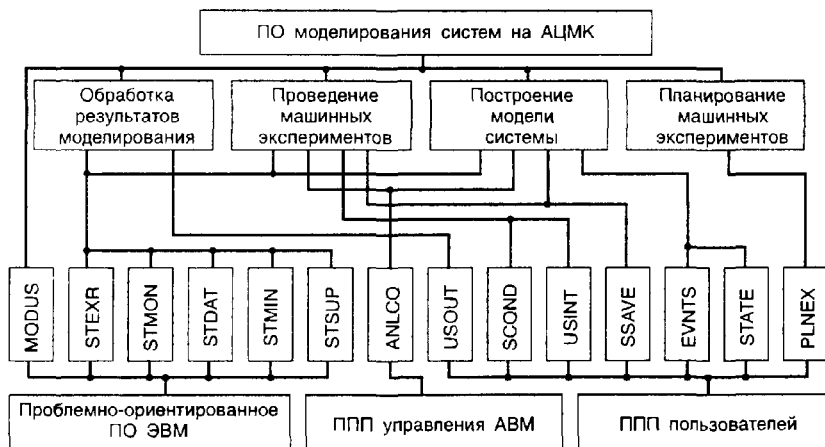


Рис. 5.13. Распределение задач моделирования по средствам АЦМК

задачу моделирования конкретной системы S , пользователь может отказаться от использования тех или иных программ.

MONIT — программа-монитор, которая представляет собой совокупность программ, решающих общие задачи управления аппаратурой и обработки прерываний в программах, написанных на языке ЯОН. В состав программы-монитора входят управляющая программа, предназначенная для организации работы с прерываниями от аппаратуры и блочного обмена данными в рамках операционной системы, и пакет программ, обеспечивающих выполнение основных функций из программ, написанных на ЯОН.

Программное обеспечение АВМ представляется в виде ППП ANLCO, позволяющего проводить автоматическую настройку блоков АВМ, имитирующих непрерывную часть в модели системы и управление этой настройкой и обменом информацией с другими составляющими АЦМК.

Наиболее сложная часть ПО — комплекс программных средств ЭВМ. Не останавливаясь на рассмотрении операционной системы *MODOS*, в рамках которой реализуется модель системы S , рассмотрим состав проблемно-ориентированного ПО ЭВМ, позволяющего управлять процессом моделирования и обеспечивающего при этом взаимодействие пользователя с моделью. В его состав входят:

STSUP — супервизор комплекса, осуществляющий управление процессами моделирования системы по командам пользователя;

STMIN — интерактивный монитор комплекса, позволяющий в процессе моделирования системы в режиме дружественного интерфейса выполнять отображение объектов модели, изменение содержимого модели системы в процессе моделирования и т. п.;

STDAT — модуль, выполняющий инициализацию объектов системы моделирования;

STMON — программный монитор комплекса, обеспечивающий реализацию системных функций в программном режиме по запросам программ модели;

STEXR — управляющий модуль комплекса, осуществляющий контроль за процессом моделирования.

Использование модульной структуры и организации ПО моделирования позволяет в режиме генерации создавать ППП конкретной системы S , ориентированный на режим гибридного моделирования

и конкретную конфигурацию комплекса технических средств АЦМК.

Контрольные вопросы

- 5.1. Чем отличаются языки имитационного моделирования от языков общего назначения?
- 5.2. Как можно представить архитектуру языка имитационного моделирования?
- 5.3. Какие основные требования предъявляются к языкам имитационного моделирования?
- 5.4. Какие имеются группы языков моделирования дискретных систем?
- 5.5. Какие основные идеи положены в основу построения дерева решений по выбору языка для моделирования системы?
- 5.6. Что называется пакетом прикладных программ моделирования систем?
- 5.7. Что является функциональным и системным наполнением пакета прикладных программ моделирования?
- 5.8. Каковы функции языка задания пакета прикладных программ моделирования?
- 5.9. Какие существуют моделирующие комплексы?

ГЛАВА 6

ПЛАНИРОВАНИЕ МАШИННЫХ ЭКСПЕРИМЕНТОВ С МОДЕЛЯМИ СИСТЕМ

Имитационное моделирование является по своей сути машинным экспериментом с моделью исследуемой или проектируемой системы. План имитационного эксперимента на ЭВМ представляет собой метод получения с помощью эксперимента необходимой пользователю информации. Эффективность использования экспериментальных ресурсов существенным образом зависит от выбора плана эксперимента. Основная цель экспериментальных исследований с помощью имитационных моделей состоит в наиболее глубоком изучении поведения моделируемой системы. Для этого необходимо планировать и проектировать не только саму модель, но и процесс ее использования, т. е. проведение с ней экспериментов на ЭВМ. Весь комплекс вопросов планирования экспериментов с имитационными моделями для их успешного решения рационально разбить на стратегическое и тактическое планирование.

6.1. МЕТОДЫ ТЕОРИИ ПЛАНИРОВАНИЯ ЭКСПЕРИМЕНТОВ

Машинный эксперимент с моделью системы S 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 S . В зависимости от поставленных целей моделирования системы S на ЭВМ имеются различные подходы к организации имитационного эксперимента с машинной моделью M_d . Основная задача планирования машинных экспериментов — получение необходимой информации об исследуемой системе S при ограничениях на ресурсы (затраты машинного времени, памяти и т. п.). К числу частных задач, решаемых при планировании машинных экспериментов, относятся задачи уменьшения затрат машинного времени на моделирование, увеличения точности и достоверности результатов моделирования, проверки адекватности модели и т. д.

Машинный эксперимент. Эффективность машинных экспериментов с моделями M_d существенно зависит от выбора плана эксперимента, так как именно план определяет объем и порядок проведения вычислений на ЭВМ, приемы накопления и статистической обработки результатов моделирования системы S . Поэтому основ-

ная задача планирования машинных экспериментов с моделью M_m формулируется следующим образом: необходимо получить информацию об объекте моделирования, заданном в виде моделирующего алгоритма (программы), при минимальных или ограниченных затратах машинных ресурсов на реализацию процесса моделирования.

Таким образом, при машинном моделировании рационально планировать и проектировать не только саму модель M_m системы S , но и процесс ее использования, т. е. проведение с ней экспериментов с использованием инструментальной ЭВМ.

К настоящему времени в физике, биологии и т. д. сложилась теория планирования экспериментов, в которой разработаны достаточно мощные математические методы, позволяющие повысить эффективность таких экспериментов [10, 18, 21, 33]. Но перенос этих результатов на область машинных экспериментов с моделями M_m может иметь место только с учетом специфики моделирования систем на ЭВМ. Несмотря на то что цели экспериментального моделирования на ЭВМ и проведения натуральных экспериментов совпадают, между этими двумя видами экспериментов существуют различия, поэтому для планирования эксперимента наиболее важное значение имеет следующее: 1) простота повторения условий эксперимента на ЭВМ с моделью M_m системы S ; 2) возможность управления экспериментом с моделью M_m , включая его прерывание и возобновление; 3) легкость варьирования условий проведения эксперимента (воздействий внешней среды E); 4) наличие корреляции между последовательностью точек в процессе моделирования; 5) трудности, связанные с определением интервала моделирования $(0, T)$.

Преимуществом машинных экспериментов перед натурным является возможность полного воспроизведения условий эксперимента с моделью исследуемой системы S . Сравнить две альтернативы возможно при одинаковых условиях, что достигается, например, выбором одной и той же последовательности случайных чисел для каждой из альтернатив. Существенным достоинством перед натурными является простота прерывания и возобновления машинных экспериментов, что позволяет применять последовательные и эвристические приемы планирования, которые могут оказаться нереализуемыми в экспериментах с реальными объектами. При работе с машинной моделью M_m всегда возможно прерывание эксперимента на время, необходимое для анализа результатов и принятия решений об его дальнейшем ходе (например, о необходимости изменения значений параметров модели M_m).

Недостатком машинных экспериментов является то, что часто возникают трудности, связанные с наличием корреляции в выходных последовательностях, т. е. результаты одних наблюдений зависят от результатов одного или нескольких предыдущих, и поэтому

в них содержится меньше информации, чем в независимых наблюдениях. Так как в большинстве существующих методов планирования экспериментов предполагается независимость наблюдений, то многие из этих методов нельзя непосредственно применять для машинных экспериментов при наличии корреляции [18, 21, 29, 46, 53].

Основные понятия планирования экспериментов. Рассмотрим основные понятия теории планирования экспериментов. В связи с тем что математические методы планирования экспериментов основаны на кибернетическом представлении процесса проведения эксперимента, наиболее подходящей моделью последнего является абстрактная схема, называемая «черным ящиком». При таком кибернетическом подходе различают входные и выходные переменные: $x_1, x_2, \dots, x_k; y_1, y_2, \dots, y_l$. 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 M_s системы S фактор является экзогенной или управляемой (входной) переменной, а реакция — эндогенной (выходной) переменной. Например, в агрегативной системе (*A-схеме*) факторами будут входные и управляющие сообщения, а реакциями — выходные.

Каждый фактор $x_i, i = \overline{1, k}$, 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.

Каждому фиксированному набору уровней факторов соответствует определенная точка в многомерном пространстве, называемом *факторным пространством*. Эксперименты не могут быть реализованы во всех точках факторного пространства, а лишь в принадлежащих допустимой области, как, например, это показано для случая двух факторов x_1 и x_2 на рис. 6.1 (плоскость $x_1 0_1 x_2$).

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_l , 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:

$$\tilde{y}_l = \varphi_l(x_1, x_2, \dots, x_k), l = \overline{1, m}.$$

Зависимости φ , 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.

При планировании экспериментов необходимо определить основные свойства факторов. Факторы при проведении экспериментов могут быть управляемыми и неуправляемыми, наблюдаемыми и ненаблюдаемыми, изучаемыми и неизучаемыми, количественными и качественными, фиксированными и случайными.

Фактор называется *управляемым*, если его уровни целенаправленно выбираются исследователем в процессе эксперимента. При машинной реализации модели M_m исследователь принимает решения, управляя изменением в допустимых пределах различных факторов.

Фактор называется *наблюдаемым*, если его значения наблюдаются и регистрируются. Обычно в машинном эксперименте с моделью M_m наблюдаемые факторы совпадают с управляемыми, так как нерационально управлять фактором, не наблюдая его. Но неуправляемый фактор также можно наблюдать. Например, на этапе проектирования конкретной системы S нельзя управлять заданными воздействиями внешней среды E , но можно наблюдать их в машинном эксперименте. *Наблюдаемые неуправляемые факторы* получили название *сопутствующих*. Обычно при машинном эксперименте с моделью M_m число сопутствующих факторов велико, поэтому рационально учитывать влияние лишь тех из них, которые наиболее существенно воздействуют на интересующую исследователя реакцию.

Фактор относится к *изучаемым*, если он включен в модель M_m для изучения свойств системы S , а не для вспомогательных целей, например для увеличения точности эксперимента.

Фактор будет *количественным*, если его значения — числовые величины, влияющие на реакцию, а в противном случае *фактор* называется *качественным*. Например, в модели системы, формализуемой в виде схемы массового обслуживания (*Q-схемы*), количественными факторами являются интенсивности входящих потоков заявок, интенсивности потоков обслуживания, емкости накопителей, количество обслуживающих каналов и т. д., а качественными факторами — дисциплины постановки в очередь, выбора из очереди, обслуживания заявок каналами и т. д. Качественным факторам в отличие от количественных не соответствует числовая шкала. Однако и для них можно построить условную порядковую шкалу, с помощью которой производится кодирование, устанавливая соответствие между условиями качественного фактора и числами натурального ряда.

Фактор называется *фиксированным*, если в эксперименте исследуются все интересующие экспериментатора значения фактора, а если экспериментатор исследует только некоторую случайную выборку из совокупности интересующих значений факторов, то *фактор* называется *случайным*. На основании случайных факторов могут быть сделаны вероятностные выводы и о тех значениях факторов, которые в эксперименте не исследовались.

В машинных экспериментах с моделями M_m не бывает неуправляемых или ненаблюдаемых факторов применительно к исследуемой системе S . В качестве воздействий внешней среды E , т. е. неуправляемых и ненаблюдаемых факторов, в машинной имитационной модели выступают стохастические экзогенные переменные. Если имитационная модель сформулирована, то все факторы определены и нельзя во время проведения данного эксперимента (испытания) с моделью M_m вводить дополнительные факторы.

Как уже отмечалось, каждый фактор может принимать в испытании одно или несколько значений, называемых *уровнями*, причем фактор будет управляемым, если его уровни целенаправленно выбираются экспериментатором. Для полного определения фактора необходимо указать последовательность операций, с помощью которых устанавливаются его конкретные уровни. Такое определение фактора называется *операциональным* и обеспечивает однозначность понимания фактора.

Основными требованиями, предъявляемыми к факторам, являются требование управляемости фактора и требование непосредственного воздействия на объект. Под управляемостью фактора понимается возможность установки и поддержания выбранного нужного уровня фактора постоянным в течение всего испытания или изменяющимся в соответствии с заданной программой. Требование непосредственного воздействия на объект имеет большое значение в связи с тем, что трудно управлять фактором, если он является функцией других факторов.

При планировании эксперимента обычно одновременно изменяются несколько факторов. Определим требования, которые предъявляются к совокупности факторов. Основные из них — совместимость и независимость. Совместимость факторов означает, что все их комбинации осуществимы, а независимость соответствует возможности установления фактора на любом уровне независимо от уровней других.

При проведении машинного эксперимента с моделью M_m для оценки некоторых характеристик процесса функционирования исследуемой системы S экспериментатор стремится создать такие условия, которые способствуют выявлению влияния факторов, находящихся в функциональной связи с искомой характеристикой. Для этого необходимо: отобрать факторы x_i , $i = 1, k$, влияющие на искомую характеристику, и описать функциональную зависимость;

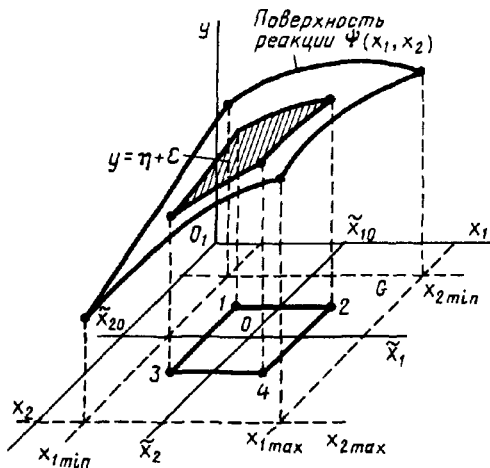


Рис. 6.1. Геометрическое представление поверхности реакции

установить диапазон изменения факторов $x_{i\min} \div x_{i\max}$; определить координаты точек факторного пространства $\{x_1, x_2, \dots, x_k\}$, в которых следует проводить эксперимент; оценить необходимое число реализаций и их порядок в эксперименте.

Свойства объекта исследования, т. е. процесса машинного моделирования системы S , можно описывать с помощью различных методов (моделей планирования). Для выбора конкретной модели необходимо сформулировать такие ее особенности, как адекватность, содержательность, простота и т. д.

Под содержательностью модели планирования понимается ее способность объяснять множество уже известных фактов, выявлять новые и предсказывать их дальнейшее развитие. Простота — одно из главных достоинств модели планирования, выражающееся в реализуемости эксперимента на ЭВМ, но при этом имеет место противоречие с требованиями адекватности и содержательности.

Для экстремального планирования экспериментов наибольшее применение нашли модели в виде алгебраических полиномов. Предполагаем, что изучается влияние k количественных факторов x_i , $i = \overline{1, k}$, на некоторую реакцию η в отведенной для экспериментирования локальной области факторного пространства G , ограниченной $x_{i\min} \div x_{i\max}$, $i = \overline{1, k}$ (рис. 6.1 для случая $k=2$). Допустим, что функцию реакции $\psi(x_1, x_2, \dots, x_k)$ можно с некоторой степенью точности представить в виде полинома степени d от k переменных

$$\eta = \tilde{b}_0 + \sum_{1 \leq i \leq k} \tilde{b}_i x_i + \sum_{1 \leq i < j \leq k} \tilde{b}_{ij} x_i x_j + \dots + \sum_{i_1, i_2, \dots, i_k} \tilde{b}_{i_1, i_2, \dots, i_k} \times x_1^{i_1} x_2^{i_2} \dots x_k^{i_k}, \quad \sum i_j = d, \quad (6.1)$$

который содержит C_{k+d}^d коэффициентов.

Соотношение (6.1) может быть представлено как

$$\eta = \vec{f}(\vec{x}) \vec{B}, \quad (6.2)$$

где $\vec{f}(\vec{x})$ — вектор с элементами $f_\alpha(\vec{x})$, $\alpha = \overline{0, k'}$, входящими в исходный полином; \vec{B} — вектор коэффициентов, которые соответственно имеют такой вид:

$$\begin{aligned}\vec{f}(\vec{x}) &= \|f_0(\vec{x}), f_1(\vec{x}), \dots, f_{k'}(\vec{x})\|, \quad k' = C_{k+d-1}^d, \\ \vec{x} &= \|1, \dots, 1; x_1, \dots, x_k; x_1^2, \dots, x_k^2; \\ & x_1 x_2, \dots, x_{k-1} x_k; \dots; x_1^d, \dots, x_k^d; x_1^{d-1} x_2, \dots, x_{k-(d-1)} \dots x_{k-1} x_k\|; \\ \vec{B} &= \|\tilde{b}_0; \tilde{b}_1, \dots, \tilde{b}_k; \tilde{b}_{11}, \dots, \tilde{b}_{kk}; \tilde{b}_{12}, \dots, \tilde{b}_{(k-1)k}; \dots; \tilde{b}_{1\dots 1}, \dots, \tilde{b}_{k\dots k}; \\ & \tilde{b}_{1\dots 12}, \dots, \tilde{b}_{[k-(d-1)] \dots (k-1)k}\|.\end{aligned}$$

Введем фиктивную переменную $x_0 = 1$, а также переменные

$$\begin{aligned}x_{k+1} &= x_1^2, \quad x_{k+2} = x_2^2, \quad \dots, \quad x_{2k+1} = x_k^2, \\ x_{2k+2} &= x_1 x_2, \quad \dots, \quad x_{k'} = x_{k-(d-1)} \dots x_{k-1} x_k.\end{aligned}$$

Тогда (6.1) запишется как однородное линейное уравнение вида

$$\eta = \vec{f}(\vec{x}) \vec{B} = \sum_{\alpha=0}^{k'} \tilde{b}_\alpha f_\alpha(\vec{x}) = \sum_{\alpha=0}^{k'} \tilde{b}'_\alpha x_\alpha, \quad (6.3)$$

$$\begin{aligned}\text{где } \tilde{b}'_0 &= \tilde{b}_0; \quad \tilde{b}'_1 = \tilde{b}_1, \dots, \tilde{b}'_k = \tilde{b}_k; \quad \tilde{b}'_{k+1} = \tilde{b}_{11}, \dots, \tilde{b}'_{2k+1} = \tilde{b}_{kk}; \quad \tilde{b}'_{2k+2} = \\ &= \tilde{b}_{12}, \dots, \tilde{b}'_{k'} = \tilde{b}_{[k-(d-1)] \dots (k-1)k}.\end{aligned}$$

Для оценки коэффициентов в (6.3) можно применить методы линейной регрессии [10, 18, 21, 22].

Пример 6.1. Аппроксимация полиномов второго порядка функции реакции в однофакторной модели планирования может быть представлена в виде $\eta = \tilde{b}_0 + \tilde{b}_1 x_1 + \tilde{b} x_1^2$.

Более сложные объекты требуют применения полиномиальных моделей планирования большего порядка. Так, модель второго порядка в k -факторном эксперименте будет иметь вид

$$\eta = \tilde{b}_0 + \sum_{i=1}^k \tilde{b}_i x_i + \sum_{i=1}^k \tilde{b}_{ii} x_i^2 + \sum_{i \neq j}^k \tilde{b}_{ij} x_i x_j$$

На практике часто стремятся к использованию линейной модели планирования, преобразуя исходные полиномиальные модели. Например, модель второго порядка

$$\eta = \sum_{i=0}^k \tilde{b}_i x_i + \sum_{i=1}^k \sum_{j=1}^k \tilde{b}_{ij} x_i x_j$$

может быть преобразована к линейному виду путем введения фиктивных переменных $x_{ij} = x_i x_j$. Тогда в результате получается модель множественной линейной регрессии вида

$$\eta = \sum_{i=0}^k \tilde{b}'_i x_i + \sum_{i=1}^k \sum_{j=1}^k \tilde{b}'_{ij} x_{ij}$$

Функция реакции может иметь и более сложную зависимость от факторов. В этом случае некоторые из них удается привести к линейному виду. Такими моделями являются мультипликативная, регрессионная, экспоненциальная и др. [18, 21, 46].

Если выбрана модель планирования, т. е. выбран вид функции $\eta = \varphi(x_1, x_2, \dots, x_k)$ и записано ее уравнение, то остается в отведенной для исследования области факторного пространства G спланировать и провести эксперимент для оценки числовых значений констант (коэффициентов) этого уравнения.

Так как полином (6.2) или (6.3) содержит G^{k+d} коэффициентов, подлежащих определению, то план эксперимента D должен содержать по крайней мере $N \geq C^{k+d}$ различных экспериментальных точек:

$$D = \begin{pmatrix} x_{11} & x_{21} & \dots & x_{k1} \\ x_{12} & x_{22} & \dots & x_{k2} \\ \dots & \dots & \dots & \dots \\ x_{1N} & x_{2N} & \dots & x_{kN} \end{pmatrix},$$

где x_{iu} — значения, которые принимает i -я переменная в u -м испытании, $i = \overline{1, k}$, $u = \overline{1, N}$.

Реализовав испытания в N точках области факторного пространства, отведенной для экспериментирования, получим вектор наблюдений, имеющий следующий вид:

$$\vec{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix},$$

где y_u — реакция, соответствующая u -й точке плана $\vec{x}_u = \|x_{1u}, x_{2u}, \dots, x_{ku}\|$, $u = \overline{1, N}$.

При незначительном влиянии неуправляемых входных переменных и параметров по сравнению с вводимыми возмущениями управляемых переменных в планировании эксперимента предполагается верной следующая модель:

$$y_u = \eta_u + c_u = \varphi_u(\vec{x}) + e_u = b_0 x_{0u} + b_1 x_{1u} + \dots + b_k x_{ku} + \\ + b_{k+1} x_{k+1u} + \dots + b_k x_{k'u} + e_u, \quad k' = C_{k+d-1}^d, \quad u = \overline{1, N},$$

где e_u — ошибка (шум, флуктуация) испытания, которая предполагается независимой нормально распределенной случайной величиной с математическим ожиданием $M[e_u] = 0$ и постоянной дисперсией $D[e_u] = \sigma^2_e = \text{const}$.

Выписав аналогичные соотношения для всех точек плана $u = \overline{1, N}$, получим матрицу планирования

$$X = \begin{pmatrix} x_{01} & x_{11} & x_{21} & \dots & x_{k1} & x_{k+11} & \dots & x_{k'1} \\ x_{02} & x_{12} & x_{22} & \dots & x_{k2} & x_{k+12} & \dots & x_{k'2} \\ \dots & & & & & & & \\ x_{0N} & x_{1N} & x_{2N} & \dots & x_{kN} & x_{k+1N} & \dots & x_{k'N} \end{pmatrix}$$

размерностью $N \times (k' + 1)$.

Рассмотрим особенности планирования эксперимента для линейного приближения поверхности реакции, причем построению плана предшествует проведение ряда неформализованных действий (принятие решений), направленных на выбор локальной области факторного пространства G (рис. 6.1).

Вначале следует выбрать границы $x_{i \min}$ и $x_{i \max}$ области определения факторов, задаваемые исходя из свойств исследуемого объекта, т. е. на основе анализа априорной информации о системе S и внешней среде E . Например, такая переменная, как температура, при термобарических экспериментах принципиально не может быть ниже абсолютного нуля и выше температуры плавления материала, из которого изготовлена термобарокамера.

После определения области G необходимо найти локальную подобласть для планирования эксперимента путем выбора основного (нулевого) уровня x_{i0} и интервалов варьирования Δx_i , $i = \overline{1, k}$.

В качестве исходной точки x_{i0} выбирают такую, которая соответствует наилучшим условиям, определенным на основе анализа априорной информации о системе S , причем эта точка не должна лежать близко к границам области определения факторов $x_{i \min}$ и $x_{i \max}$. На выбор интервала варьирования Δx_i накладываются естественные ограничения снизу (интервал не может быть меньше ошибки фиксирования уровня фактора, так как в противном случае верхний и нижний уровни окажутся неразличимыми) и сверху (верхний и нижний уровни не должны выходить за область определения G).

В рамках выбранной модели планирования в виде алгебраических полиномов строится план эксперимента путем варьирования каждого из факторов x_i , $i = \overline{1, k}$, на нескольких уровнях q относительно исходной точки x_{i0} , представляющей центр эксперимента.

Виды планов экспериментов. Эксперимент, в котором реализуются все возможные сочетания уровней факторов, называется *полным факторным экспериментом* (ПФЭ). Если выбранная модель планирования включает в себя только линейные члены полинома и их произведения, то для оценки коэффициентов модели используется план эксперимента с варьированием всех k факторов на двух уровнях, т. е. $q=2$. Такие планы называются планами типа 2^k , где $N=2^k$ — число всех возможных испытаний.

Начальный этап планирования эксперимента для получения коэффициентов линейной модели основан на варьировании факторов на двух уровнях: нижнем $x_{iн}$ и верхнем $x_{iв}$, — симметрично расположенных относительно основного уровня x_{i0} , $i = \overline{1, k}$. Геометрическая интерпретация показана на рис. 6.2, а. Так как каждый фактор принимает лишь два значения $x_{iн} = x_{i0} - \Delta x_i$ и $x_{iв} = x_{i0} + \Delta x_i$, то для стандартизации и упрощения записи условий каждого испытания и обработки выборочных данных эксперимента масштабы по осям факторов выбираются так, чтобы нижний уровень соответствовал -1 , верхний — $+1$, а основной — нулю. Это легко достигается с помощью преобразования вида

$$\tilde{x}_i = (x_i - x_{i0}) / \Delta x_i, \quad i = \overline{1, k},$$

где \tilde{x}_i — кодированное значение i -го фактора; x_i — натуральное значение фактора; x_{i0} — нулевой уровень; $\Delta x_i = (x_{iв} - x_{iн}) / 2$ — интервал варьирования фактора.

Пример 6.2. Пусть в качестве i -го фактора выступает такая переменная, как температура T , °С, т. е. $x_i = T$, причем выбраны основной уровень $x_{i0} = 100$ °С и интервал варьирования $\Delta x_i = 20$ °С. Тогда кодированные значения \tilde{x}_i по уровням соответственно будут $(80 - 100) / 20 = -1$ для нижнего, $(120 - 100) / 20 = +1$ для верхнего, $(100 - 100) / 20 = 0$ для основного.

Расположение точек для ПФЭ типа 2^2 показано на рис. 6.1, а также на рис. 6.2, б. Выписывая комбинации уровней факторов для каждой экспериментальной точки квадрата, получим план D полного факторного эксперимента типа 2^2 :

Номер испытания	1	2	3	4
\tilde{x}_1	-1	+1	-1	+1
x_2	-1	-1	+1	+1
Обозначения строк	(1)	a	b	ab

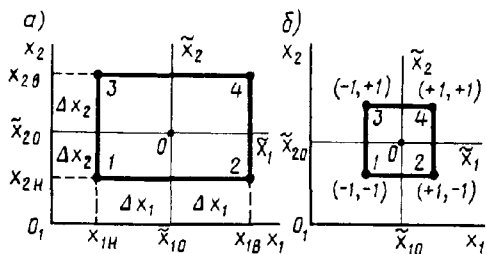


Рис. 6.2. Геометрическая интерпретация полного факторного эксперимента типа 2^2 :

a — без масштабирования; *б* — при масштабировании по осям

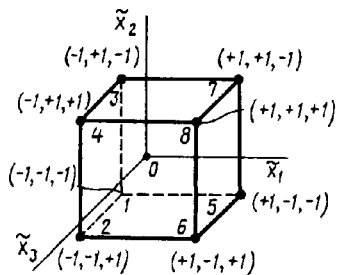


Рис. 6.3. Геометрическая интерпретация полного факторного эксперимента типа 2^3

При этом планы можно записывать сокращенно с помощью условных буквенных обозначений строк. Для этого порядковый номер фактора ставится в соответствие строчной букве латинского алфавита: $x_1 \rightarrow a$, $x_2 \rightarrow b$ и т. д.

Затем для каждой строки плана выписываются латинские буквы только для факторов, находящихся на верхних уровнях; испытание со всеми факторами на нижних уровнях обозначается как (1). Запись плана в буквенных обозначениях показана в последней строчке.

Пример 6.3. Геометрическая интерпретация ПФЭ 2^3 приведена на рис. 6.3, а его план ниже:

Номер испытания	..	1	2	3	4	5	6	7	8
\bar{x}_1	-1	-1	-1	-1	+1	+1	+1	+1
\bar{x}_2	-1	-1	+1	+1	-1	-1	+1	+1
\bar{x}_3	-1	+1	-1	+1	-1	+1	-1	+1
Обозначение строк	..	(1)	c	b	bc	a	ac	ab	abc

Полный факторный эксперимент дает возможность определить не только коэффициенты регрессии, соответствующие линейным эффектам, но и коэффициенты регрессии, соответствующие всем эффектам взаимодействия. Эффект взаимодействия двух (или более) факторов появляется при одновременном варьировании этих факторов, когда действие каждого из них на выход зависит от уровня, на которых находятся другие факторы.

Для оценки свободного члена b_0 и определения эффектов взаимодействия b_{12} , b_{13} , ..., b_{123} , ... план эксперимента D расширяют до матрицы планирования X путем добавления соответствующей «фиктивной переменной»: единичного столбца \bar{x}_0 и столбцов произведений $\bar{x}_1\bar{x}_2$, $\bar{x}_1\bar{x}_3$, ..., $\bar{x}_1\bar{x}_2\bar{x}_3$, ..., как показано, например, для ПФЭ типа 2^3 в табл. 6.1.

Таблица 6.1

Номер испытания	x_0	План ПФЭ			$\tilde{x}_1\tilde{x}_2$	$\tilde{x}_1\tilde{x}_3$	$\tilde{x}_2\tilde{x}_3$	$\tilde{x}_1\tilde{x}_2\tilde{x}_3$	Реакция y
		\tilde{x}_1	\tilde{x}_2	\tilde{x}_3					
1	+1	-1	-1	-1	+1	+1	+1	-1	y_1
2	+1	-1	-1	+1	+1	-1	-1	+1	y_2
3	+1	-1	+1	-1	-1	+1	-1	+1	y_3
4	+1	-1	+1	+1	-1	-1	+1	-1	y_4
5	+1	+1	-1	-1	-1	-1	+1	+1	y_5
6	+1	+1	-1	+1	-1	+1	-1	-1	y_6
7	+1	+1	+1	-1	+1	-1	-1	-1	y_7
8	+1	+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.

Рассмотрим построение планов так называемого дробного факторного эксперимента. Пусть имеется простейший полный факторный эксперимент типа 2^2 . Используя матрицу планирования, приведенную в табл. 6.2, можно вычислить коэффициенты и представить результаты в виде уравнения

$$y = b_0 + b_1\tilde{x}_1 + b_2\tilde{x}_2 + b_{12}\tilde{x}_1\tilde{x}_2.$$

Таблица 6.2

Номер испытания	x_0	План ПФЭ		$(\tilde{x}_1)\tilde{x}_1\tilde{x}_2$	Реакция y
		\tilde{x}_1	\tilde{x}_2		
1	+1	-1	-1	+1	y_1
2	+1	+1	-1	-1	y_2
3	+1	-1	+1	-1	y_3
4	+1	+1	+1	+1	y_4

Если в выбранных интервалах варьирования уровня процесс можно описать линейной моделью, то достаточно определить три коэффициента: b_0 , b_1 и 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 $\tilde{x}_1\tilde{x}_2$ (табл. 6.2) можно использовать для нового фактора \tilde{x}_3 . Поставим в табл. 6.2 этот фактор в скобках над взаимодействием $\tilde{x}_1\tilde{x}_2$. В этом случае раздельных оценок, которые имели место в ПФЭ типа 2^k , уже не будет и оценки смещаются следующим образом: $\tilde{b}_1 \rightarrow \beta_1 + \beta_{23}$, $\tilde{b}_2 \rightarrow \beta_2 + \beta_{13}$, $\tilde{b}_3 \rightarrow \beta_3 + \beta_{12}$.

При постулировании линейной модели все парные взаимодействия не учитывают. Таким образом, вместо восьми испытаний в полном факторном эксперименте типа 2^3 необходимо провести только

четыре. Правило проведения дробного факторного эксперимента формулируется так: для сокращения числа испытаний новому фактору присваивается значение вектор-столбца матрицы, принадлежащего взаимодействию, которым можно пренебречь.

При проведении эксперимента из четырех испытаний для оценки влияния трех факторов пользуются половиной ПФЭ типа 2^3 , так называемой «полурепликой». Если приравнять x_3 и $-x_1x_2$, то можно получить вторую «полуреплику». Для обозначения дробных реплик, в которых d линейных эффектов приравнены к эффектам взаимодействия, пользуются условным обозначением 2^{k-d} . Например, «полуреплика» от 2^6 записывается в виде 2^{6-1} , а «четверть-реплика» — 2^{6-2} .

Пример 6.4. При построении «полуреплика» $2^{3-1} x_3$ можно приравнять к x_1x_2 или $-x_1x_2$. Две «полуреплики» 2^{3-1} показаны в табл. 6.3. Для произведения трех столбцов левой матрицы выполняется соотношение $+1 = x_1x_2x_3$, а правой матрицы $-1 = x_1x_2x_3$, т. е. все знаки столбцов произведений одинаковы и в первом случае равны $+1$, а во втором -1 .

Кроме симметричных двухуровневых планов типа 2^k при планировании экспериментов применяют также многоуровневые планы, в которых факторы варьируются на 3, 4, ..., m -м уровнях и обозначаются соответственно как 3^k , 4^k , ..., m^k -планы. Многоуровневые несимметричные планы, в которых факторы варьируются на различных уровнях, строятся различными способами: комбинированием полных и дробных факторных планов типа 2^k , методом преобразования симметричных планов в несимметричные и т. д. Рассмотренные планы носят название планов регрессионного анализа для многофакторного эксперимента [10, 22].

Таблица 6.3

Номер испытания	$x_3 = x_1x_2$				Номер испытания	$x_3 = -x_1x_2$			
	x_1	x_2	x_3	$x_1x_2x_3$		x_1	x_2	x_3	$x_1x_2x_3$
1	+1	+1	+1	+1	1	+1	+1	+1	-1
2	-1	-1	+1	+1	2	-1	-1	+1	-1
3	+1	-1	-1	+1	3	+1	-1	-1	-1
4	-1	+1	-1	+1	4	-1	+1	-1	-1

Когда модель планирования анализируется методами дисперсионного анализа, применяют планы дисперсионного анализа. Если при постановке эксперимента реализуются все возможные совокупности условий, то говорят о полных классификациях дисперсионного анализа. Если проводится сокращение перебора вариантов — это неполная классификация дисперсионного анализа. Сокращение перебора может проводиться случайным образом (без ограничения на рандомизацию) или в соответствии с некоторыми правилами (с ограничениями на рандомизацию). Чаще всего в качестве таких

планов используют блочные планы и планы типа латинского квадрата [10, 18, 21].

Перейдем далее к рассмотрению вопросов, связанных непосредственно с планированием экспериментов с машинной моделью M_m конкретной системы S .

6.2. СТРАТЕГИЧЕСКОЕ ПЛАНИРОВАНИЕ МАШИННЫХ ЭКСПЕРИМЕНТОВ С МОДЕЛЯМИ СИСТЕМ

Применяя системный подход к проблеме планирования машинных экспериментов с моделями систем, можно выделить две составляющие планирования: стратегическое и тактическое планирование [46].

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

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

Проблемы стратегического планирования. При стратегическом планировании машинных экспериментов с моделями систем возникает целый ряд проблем, взаимно связанных как с особенностями функционирования моделируемого объекта (системы S), так и с особенностями машинной реализации модели M_m и обработки результатов эксперимента. В первую очередь к таким относятся проблемы построения плана машинного эксперимента; наличия большого количества факторов; многокомпонентной функции реакции; стохастической сходимости результатов машинного эксперимента; ограниченности машинных ресурсов на проведение эксперимента.

Рассмотрим существо этих проблем, возникающих при стратегическом планировании машинных экспериментов, и возможные методы их решения. При построении плана эксперимента необходимо помнить, что целями проведения машинных экспериментов с моделью M_m системы S являются либо получение зависимости реакции от факторов для выявления особенностей изучаемого процесса функционирования системы S , либо нахождение такой комбинации значений факторов, которая обеспечивает экстремальное значение реакции. Другими словами, экспериментатору на базе машинной

модели M_x необходимо решить либо задачу анализа, либо задачу синтеза системы S [7, 17, 33, 46].

Очевидно, что при реализации полного факторного плана различия между машинными экспериментами для достижения той или иной цели стираются, так как оптимальный синтез сводится к выбору одного из вариантов, полученного при полном факторном анализе. Но полный факторный эксперимент эквивалентен в этом случае полному перебору вариантов, что нерационально с точки зрения затрат машинных ресурсов. Для более эффективного (с точки зрения затрат машинного времени и памяти на моделирование) нахождения оптимальной комбинации уровней факторов можно воспользоваться выборочным методом определения оптимума поверхности реакции (систематическая или случайная выборка). Методы систематической выборки включают в себя факторный (метод равномерной сетки), одного фактора, предельного анализа, наискорейшего спуска. Выбор того или иного метода рационально проводить на основе априорной информации о моделируемой системе S .

Другая специфическая проблема стратегического планирования машинных экспериментов — наличие большого количества факторов. Это одна из основных проблем реализации имитационных моделей на ЭВМ, так как известно, что в факторном анализе количество комбинаций факторов равно произведению числа значений всех факторов эксперимента. Например, если число факторов $k=10$ и имеется два значения каждого фактора, т. е. $q_i=2$, то полный факторный анализ потребует моделирования $2^k=2^{10}=1024$ комбинаций. Если факторы $x_i, i=\overline{1, k}$, являются количественными, а реакция y связана с факторами некоторой функцией ψ , то в качестве метода обработки результатов эксперимента может быть выбран регрессионный анализ. Когда при моделировании требуется полный факторный анализ, то проблема большого количества факторов может не иметь решения. Достоинством полных факторных планов является то, что они дают возможность отобразить всю поверхность реакции системы, если количество факторов невелико. Эффективность этого метода существенно зависит от природы поверхности реакции.

Так как полные факторные планы изучения даже достаточно простых моделей приводят к большим затратам машинного времени, то приходится строить неполные факторные планы, требующие меньшего числа точек, приводя при этом к потере допустимого количества информации о характере функции реакции. В этом случае рациональный подход — построение плана эксперимента исходя из поверхности реакции (план поверхности реакции), что позволяет по сравнению с факторными планами уменьшить объем эксперимента без соответствующих потерь количества получаемой информации. Методы поверхности реакции позволяют сделать некоторые выводы из самых

первых экспериментов с машинной моделью M_x . Если дальнейшее проведение машинного эксперимента оказывается неэкономичным, то его можно закончить в любой момент. Наконец, эти методы используются на начальном этапе постановки эксперимента для определения оптимальных условий моделирования исследуемой системы S .

Следующей проблемой стратегического планирования машинных экспериментов является многокомпонентная функция реакции. В имитационном эксперименте с вариантами модели системы S на этапе ее проектирования часто возникает задача, связанная с необходимостью изучения большого числа переменных реакции. Эту трудность в ряде случаев можно обойти, рассматривая имитационный эксперимент с моделью по определению многих реакций как несколько имитационных экспериментов, в каждом из которых исследуется (наблюдается) только одна реакция. Кроме того, при исследовании системы S часто требуется иметь переменные реакции, связанные друг с другом, что практически приводит к усложнению планирования имитационного эксперимента. В этом случае рационально использовать интегральные оценки нескольких реакций, построенные с использованием весовых функций, функций полезности и т. д. [10, 18, 21, 46].

Существенное место при планировании экспериментов с имитационными моделями, реализуемыми методом статистического моделирования на ЭВМ, занимает проблема стохастической сходимости результатов машинного эксперимента. Эта проблема возникает вследствие того, что целью проведения конкретного машинного эксперимента при исследовании и проектировании системы S является получение на ЭВМ количественных характеристик процесса функционирования системы S с помощью машинной модели M_x . В качестве таких характеристик наиболее часто выступают средние некоторых распределений, для оценки которых применяют выборочные средние, найденные путем многократных прогонов модели на ЭВМ, причем чем больше выборка, тем больше вероятность того, что выборочные средние приближаются к средним распределений. Сходимость выборочных средних с ростом объема выборки называется *стохастической сходимостью*.

Основной трудностью при определении интересующих характеристик процесса функционирования системы S является медленная стохастическая сходимость. Известно, что мерой флуктуаций случайной величины служит ее нестандартное отклонение. Если σ — стандартное отклонение одного наблюдения, то стандартное отклонение среднего N наблюдений будет равно σ/\sqrt{N} . Таким образом, для уменьшения случайной ошибки в K раз требуется увеличить объем выборки в K^2 раз, т. е. для получения заданной точности оценки может оказаться, что объем необходимой выборки нельзя получить на ЭВМ из-за ограничения ресурса времени и памяти.

Медленная стохастическая сходимость в машинных имитационных экспериментах с заданной моделью M_m требует разработки специальных методов решения этой проблемы. Необходимо учитывать, что в машинном эксперименте после того, как модель сформулирована, включение дополнительных факторов для повышения точности невозможно, так как это потребует изменения конструкции модели M_m . Основная идея ускорения сходимости в машинных экспериментах со стохастическими моделями состоит в использовании априорной информации о структуре и поведении системы S , свойствах распределения входных переменных и наблюдаемых случайных воздействий внешней среды E . К методам ускорения сходимости относятся методы регрессионной выборки, дополняющей переменной, расслоенной выборки, значимой выборки [10, 18, 21, 46].

Переходя к рассмотрению проблемы ограниченности машинных ресурсов на проведение экспериментов с моделью системы S , необходимо помнить о том, что построение плана эксперимента с использованием различных подходов, рассмотренных в § 6.1, позволяет решить проблему стратегического планирования только с теоретической точки зрения. Но при планировании машинных экспериментов на практическую реализуемость плана существенное влияние оказывают имеющиеся в распоряжении экспериментатора ресурсы. Поэтому планирование машинного эксперимента представляет собой итерационный процесс, когда выбранная модель плана эксперимента проверяется на реализуемость, а затем, если это необходимо, вносятся соответствующие коррективы в исходную модель [46].

Этапы стратегического планирования. 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.

Структурная модель плана эксперимента характеризуется числом факторов и числом уровней для каждого фактора. Число элементов эксперимента

$$N_c = q_1 q_2 \dots q_k,$$

где k — число факторов эксперимента; q_i — число уровней i -го фактора, $i = \overline{1, k}$. При этом под элементом понимается структурный блок эксперимента, определяемый как простейший эксперимент в случае одного фактора и одного уровня, т. е. $k=1$, $q=1$, $N_c=1$.

Вопрос о виде и числе необходимых факторов следует рассматривать с различных точек зрения, причем основной является цель проводимого машинного эксперимента, т. е. в первую очередь решается вопрос о тех реакциях, которые надо оценить в результате эксперимента с машинной моделью M_m системы S . При этом надо

найти наиболее существенные факторы, так как из опыта известно, что для большинства систем 20% факторов определяют 80% свойств системы S , а остальные 80% факторов определяют лишь 20% ее свойств [46].

Следующий шаг в конструировании структурной модели плана состоит в определении уровней, на которых следует устанавливать и измерять каждый фактор, причем минимальное число уровней фактора, не являющегося постоянным, равно двум. Число уровней следует выбирать минимальным, но достаточным для достижения цели машинного эксперимента. При этом надо помнить, что каждый дополнительный уровень увеличивает затраты ресурсов на реализацию эксперимента на ЭВМ.

Анализ результатов существенно упрощается, если уровни равноотстоят друг от друга, т. е. ортогональное разбиение упрощает определение коэффициентов аппроксимации. Можно получить значительные аналитические упрощения, если принять число уровней всех факторов одинаковым. Тогда структурная модель будет симметричной и примет вид $N_c = q^k$, где $q = q_i$; $i = \overline{1, k}$.

Функциональная модель плана эксперимента определяет количество элементов структурной модели N_ϕ , т. е. необходимое число различных информационных точек. При этом функциональная модель может быть полной и неполной. Функциональная модель называется полной, если в оценке реакции участвуют все элементы, т. е. $N_\phi = N_c$, и неполной, если число реакций меньше числа элементов, т. е. $N_\phi < N_c$. Основная цель построения функциональной модели — нахождение компромисса между необходимыми действиями при машинном эксперименте (исходя из структурной модели) и ограниченными ресурсами на решение задачи методом моделирования.

Для более быстрого нахождения компромиссного решения можно при предварительном планировании машинного эксперимента использовать номограмму, построенную при варьировании числа факторов k , числа уровней факторов q , повторений эксперимента p , а также затрат времени на прогон модели τ и стоимости машинного времени c . Вид такой номограммы показан на рис. 6.4, причем при ее построении предполагалось, что полное число прогонов, необходимых при симметрично повторяемом эксперименте,

$$N = pq^k. \quad (6.4)$$

Рассмотрим особенности пользования такой номограммой на примере.

Пример 6.5. Пусть необходимо спланировать машинный эксперимент при наличии трех факторов $k = 3$, каждый из которых имеет три уровня $q = 3$, причем требуется $p = 15$ повторений с затратами $\tau = 120$ с машинного времени на один прогон при стоимости 1 ч машинного времени $c = 100$ руб. Кроме того, предполагается, что

в день на моделирование данной системы S выделяется 60 мин машинного времени, т. е. на моделирование требуется $k = N\tau/3600$ дней. Такой машинный эксперимент потребует около 400 прогонов, затрат примерно $\tau_k = 13$ ч машинного времени, около $T_k = 13$ дней на получение результатов моделирования и 1304 руб. для оплаты машинного времени.

Сравним случай, рассмотренный в примере, при условии, что число уровней факторов уменьшено до двух, т. е. $q=2$. Такой машинный эксперимент потребует только 135 прогонов; 4,5 ч машинного времени; 4,5 дня на получение результатов и всего 450 руб. затрат для оплаты машинного времени, т. е. имеет место сокращение затрат на 265%.

Такая номограмма (рис. 6.4) может быть использована и для других входов, например при фиксированной величине денежных средств, отводимых на машинный эксперимент.

Для более детального анализа имеющихся у экспериментатора возможностей при планировании эксперимента рассмотрим попарно относительное влияние числа факторов k , числа уровней q и числа повторений p на количество необходимых машинных прогонов модели N . Предполагая эти величины непрерывными, проанализируем, какая из трех величин дает наибольшее сокращение полного количества прогонов. Для этого продифференцируем уравнение (6.4):

$$\frac{\partial N}{\partial k} \frac{\partial N}{\partial q} = \frac{q \ln q}{k}, \quad \frac{\partial N}{\partial p} \frac{\partial N}{\partial q} = \frac{q}{kp}, \quad \frac{\partial N}{\partial p} \frac{\partial N}{\partial k} = \frac{1}{p \ln q} \quad (6.5)$$

Из этих уравнений видно, что: 1) если $kp > q$ и $k > q \ln q$, то доминирует (оказывает наибольшее влияние на число машинных прогонов N) изменение числа уровней q ; 2) если $kp > q$ и $k > q \ln q$, то доминирует число факторов k ; 3) если $p < q$ и $p \ln q < 1$, то доминирует число повторений p .

Такой анализ позволяет дать наглядную графическую интерпретацию определения доминирующей для данного машинного эксперимента с моделью системы S переменной: k , q или p . Графически изобразим уравнения (6.5). На рис. 6.5, а приведен график отношения $(q \ln q)/k$ как функции числа уровней q при изменении числа факторов k от 1 до 5. Если отношение $(q \ln q)/k > 1$ при данных k и q , то доминирует число факторов k . Если это отношение меньше 1, то доминирует число уровней q .

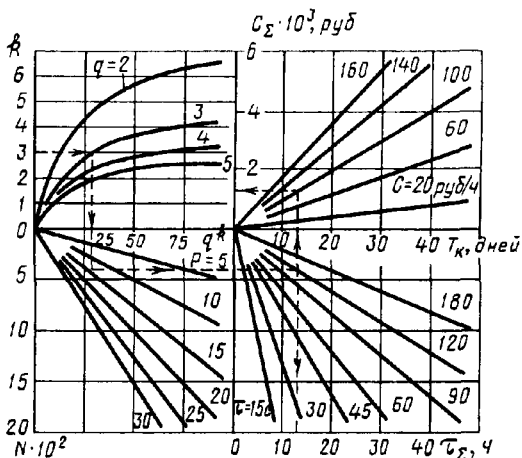


Рис. 6.4. Номограмма предварительного планирования машинного эксперимента

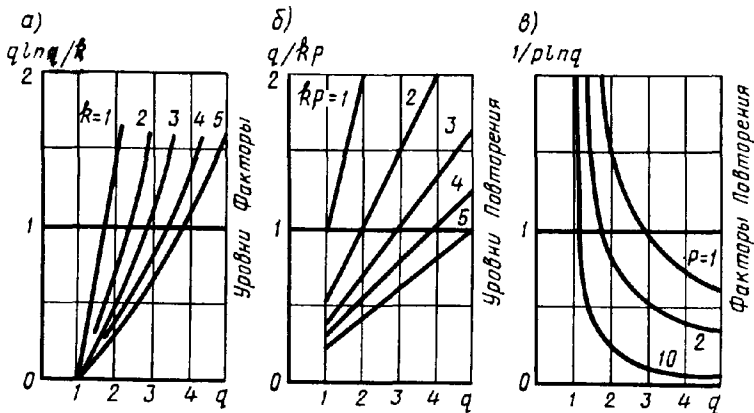


Рис. 6.5. Графическое изображение зависимостей:

а — $g \ln g/k$; б — $g/(kp)$; в — $1/(p \ln g)$ в функции

На рис. 6.5, б приведен график зависимости отношения $q/(kp)$ от числа уровней q для величин произведений kp в пределах от 1 до 5. Если в данном случае $q/(kp) > 1$, то доминирует число повторений p , а если $q/(kp) < 1$, то доминирует число уровней q .

На рис. 6.5, в показан график зависимости отношений $1/(p \ln q)$ от числа уровней q для числа повторений p , изменяющихся в пределах от 1 до 10. Если $1/(p \ln q) > 1$, то доминирует число повторений p , а если $1/(p \ln q) < 1$, то доминирует число факторов k .

Пример 6.6. Пусть при составлении плана машинного эксперимента требуется оценить, какая переменная играет доминирующую роль в сокращении полного числа машинных прогонов модели N при $k=4$, $q=3$, $p=3$. Воспользуемся рис. 6.5, а: для $q=3$ и $k=4$ отношение $(q \ln q)/k < 1$, т. е. число уровней q доминирует над числом факторов k . Исходя из рис. 6.5, б, для $q=3$, $kp=8$ имеем $q/(kp) < 1$, т. е. число уровней q доминирует над числом повторений p . И наконец, воспользовавшись рис. 6.5, в, видим, что для $q=3$ и $p=2$ отношение $1/(p \ln q) < 1$, т. е. число факторов k доминирует над числом повторений p .

Таким образом, использование при стратегическом планировании машинных экспериментов с M_n структурных и функциональных моделей плана позволяет решить вопрос о практической реализуемости модели на ЭВМ исходя из допустимых затрат ресурсов на моделирование системы S .

6.3. ТАКТИЧЕСКОЕ ПЛАНИРОВАНИЕ МАШИННЫХ ЭКСПЕРИМЕНТОВ С МОДЕЛЯМИ СИСТЕМ

Тактическое планирование эксперимента с машинной моделью M_n системы S связано с вопросами эффективного использования выделенных для эксперимента машинных ресурсов и определением конкретных способов проведения испытаний модели M_n , намеченных планом эксперимента, построенным при стратегическом плани-

ровании. Тактическое планирование машинного эксперимента связано прежде всего с решением следующих проблем: 1) определения начальных условий и их влияния на достижение установившегося результата при моделировании; 2) обеспечения точности и достоверности результатов моделирования; 3) уменьшения дисперсии оценок характеристик процесса функционирования моделируемых систем; 4) выбора правил автоматической остановки имитационного эксперимента с моделями систем [36, 37, 46].

Проблема определения начальных условий и их влияния на достижение установившегося результата при моделировании. Первая проблема при проведении машинного эксперимента возникает вследствие искусственного характера процесса функционирования модели M_m , которая в отличие от реальной системы S работает эпизодически, т. е. только когда экспериментатор запускает машинную модель и проводит наблюдения. Поэтому всякий раз, когда начинается очередной прогон модели процесса функционирования системы S , требуется определенное время для достижения условий равновесия, которые соответствуют условиям функционирования реальной системы. Таким образом, начальный период работы машинной модели M_m искажается из-за влияния начальных условий запуска модели. Для решения этой проблемы либо исключается из рассмотрения информация о модели M_m , полученная в начальной части периода моделирования $(0, T)$, либо начальные условия выбираются так, чтобы сократить время достижения установившегося режима. Все эти приемы позволяют только уменьшить, но не свести к нулю время переходного процесса при проведении машинного эксперимента с моделью M_m .

Проблема обеспечения точности и достоверности результатов моделирования. Решение второй проблемы тактического планирования машинного эксперимента связано с оценкой точности и достоверности результатов моделирования (при конкретном методе реализации модели, например, методе статистического моделирования на ЭВМ) при заданном числе реализаций (объеме выборки) или с необходимостью оценки необходимого числа реализаций при заданных точности и достоверности результатов моделирования системы S .

Как уже отмечалось, статистическое моделирование системы S — это эксперимент с машинной моделью M_m . Обработка результатов подобного имитационного эксперимента принципиально не может дать точных значений показателя эффективности E системы S ; в лучшем случае можно получить только некоторую оценку \tilde{E} такого показателя. При этом экономические вопросы затрат людских и машинных ресурсов, обосновывающие целесообразность статистического моделирования вообще, оказываются тесно связанными с вопросами точности и достоверности оценки показателя эффективности E системы S на ее модели M_m [4, 7, 11, 18, 21, 25].

Таким образом, количество реализаций N при статистическом моделировании системы S должно выбираться исходя из двух основных соображений: определения затрат ресурсов на машинный эксперимент с моделью M_m (включая построение модели и ее машинную реализацию) и оценки точности и достоверности результатов эксперимента с моделью системы S (при заданных ограничениях на ресурсы). Очевидно, что требования получения более хороших оценок и сокращения затрат ресурсов являются противоречивыми и при планировании машинных экспериментов на базе статистического моделирования необходимо решить задачу нахождения разумного компромисса между ними.

Из-за наличия стохастичности и ограниченности числа реализаций N в общем случае $\bar{E} \neq E$. При этом величина E называется *точностью* (абсолютной) *оценки*:

вероятность того, что неравенство

$$|E - \bar{E}| < \varepsilon, \quad (6.6)$$

выполняется, называется *достоверностью оценки*

$$Q = P \{ |E - \bar{E}| < \varepsilon \}. \quad (6.7)$$

Величина $\varepsilon_0 = \varepsilon/E$ называется *относительной точностью оценки*, а достоверность оценки соответственно будет иметь вид

$$Q = P \{ |(E - \bar{E})/E| < \varepsilon_0 \}.$$

Для того чтобы при статистическом моделировании системы S по заданным E (или E_0) и Q определить количество реализаций N или, наоборот, при ограниченных ресурсах (известном N) найти необходимые E и Q , следует детально изучить соотношение (6.7). Сделать это удастся не во всех случаях, так как закон распределения вероятностей величины $|E - \bar{E}|$ для многих практических случаев исследования систем установить не удастся либо в силу ограниченности априорных сведений о системе S , либо из-за сложности вероятностных расчетов. Основным путем преодоления подобных трудностей является выдвижение предположений о характере законов распределения случайной величины \bar{E} , т. е. оценки показателя эффективности системы S .

Рассмотрим взаимосвязь точности и достоверности результатов с количеством реализаций при машинном эксперименте, когда в качестве показателей эффективности E выступают вероятность p , математическое ожидание a и дисперсия σ^2 .

Пусть цель машинного эксперимента с моделью M_m некоторой системы S — получение оценки \bar{p} вероятности появления $p = P(A)$ некоторого события A , определяемого состояниями процесса функционирования исследуемой системы S . В качестве оценки вероятности p в данном случае выступает частость $\bar{p} = m/N$, где m — число положительных исходов.

Тогда соотношение (6.7), связывающее точность и достоверность оценок с количеством реализаций, будет иметь вид

$$P \{ |p - m/N| < \varepsilon \} = Q, \quad P \{ p - \varepsilon < m/N < p + \varepsilon \} = Q. \quad (6.8)$$

Для ответа на вопрос о законе распределения величины $\bar{p} = m/N$ представим эту частоту в виде $\bar{p} = m/N = (1/N) \sum_{i=1}^N x_i$, так как количество наступлений события A в данной реализации из N реализаций является случайной величиной ξ , принимающей значения $x_1 = 1$ с вероятностью p и $x_2 = 0$ с дополнительной вероятностью $1-p$. Математическое ожидание и дисперсия случайной величины ξ будут таковы:

$$\begin{aligned} M[\xi] &= x_1 p + x_2 (1-p) = 1p + 0(1-p) = p; \\ D[\xi] &= (x_1 - M[\xi])^2 p + (x_2 - M[\xi])^2 (1-p) = (1-p)^2 p + \\ &+ (0-p)^2 (1-p) = p(1-p). \end{aligned}$$

Тогда

$$M[\bar{p}] = M[m/N] = (1/N) M \left[\sum_{i=1}^N x_i \right] = (1/N) N M[\xi] = p.$$

Это соотношение говорит о несмещенности оценки \bar{p} для вероятности p . С учетом независимости значений величин x_i получим

$$D[\bar{p}] = D[m/N] = (1/N^2) D \left[\sum_{i=1}^N x_i \right] = (1/N^2) N D[\xi] = p(1-p)/N.$$

В силу центральной предельной теоремы теории вероятностей [или ее частного случая — теоремы Лапласа, см. (4.8)] частота m/N при достаточно больших N можно рассматривать как случайную величину, описываемую нормальным законом распределения вероятностей с математическим ожиданием p и дисперсией $p(1-p)/N$. Поэтому соотношение (6.8) с учетом (4.8) можно переписать так:

$$P \left\{ p - \varepsilon < \frac{m}{N} < p + \varepsilon \right\} = \Phi_0 \left(\frac{p + \varepsilon - p}{\sqrt{p(1-p)}} \sqrt{N} \right) - \Phi_0 \left(\frac{p - \varepsilon - p}{\sqrt{p(1-p)}} \sqrt{N} \right) = Q.$$

Учитывая, что $\Phi_0(-z) = 1 - \Phi_0(z)$, получим

$$2\Phi_0(\varepsilon \sqrt{N}/\sqrt{p(1-p)}) = 1 + Q; \quad \Phi_0(\varepsilon \sqrt{N}/\sqrt{p(1-p)}) = (1 + Q)/2 = \varphi.$$

Тогда

$$\varepsilon \sqrt{N}/\sqrt{p(1-p)} = t_\varphi,$$

где t_φ — квантиль нормального распределения вероятностей порядка $\varphi = (1+Q)/2$; находится из специальных таблиц [18, 21].

В результате точность оценки \bar{p} вероятности p можно определить как

$$\varepsilon = t_\varphi \sqrt{p(1-p)/N},$$

т. е. точность оценки вероятностей обратно пропорциональна \sqrt{N} .

Из соотношения для точности оценки ε можно вычислить количество реализаций

$$N = t_\varphi^2 p(1-p)/\varepsilon^2, \quad (6.9)$$

необходимых для получения оценки \bar{p} с точностью ε и достоверностью Q .

Пример 6.7. Необходимо рассчитать количество реализаций N при статистическом моделировании системы S , когда в качестве показателя эффективности используется вероятность p при достоверности $Q=0,95$ ($t_\varphi=1,96$) и точности $\varepsilon=0,01; 0,02; 0,05$. Так как значения p до проведения статистического моделирования системы S неизвестны, то вычислим множество оценок N для диапазона возможных значений p , т. е. от 0 до 1, с дискретом 0,1. Результаты расчетов с использованием выражения (6.9) представлены в табл. 6.4. Из таблицы видно, что при переходе от $p=0,1$ (0,9) к $p=0,5$ количество реализаций N возрастает примерно в три раза, а при переходе от $\varepsilon=0,05$ к $\varepsilon=0,01$ количество реализаций N возрастает примерно в 25 раз [4].

Таблица 6.4

Вероятность p	Точность ε		
	0,05	0,02	0,01
0,1 (0,9)	140	900	3600
0,2 (0,8)	250	1500	6200
0,3 (0,7)	330	2100	8400
0,4 (0,6)	380	2300	9400
0,5 (0,5)	390	2400	9800

При тактическом планировании машинного эксперимента, когда решается вопрос о выборе количества реализаций N , значение p неизвестно. Поэтому на практике проводят предварительное моделирование для произвольно выбранного значения N_0 , определяют $p_0 = m/N_0$, а затем по (6.9) вычисляют, используя вместо p значение p_0 , необходимое количество реализаций N . Такая процедура оценки N может выполняться несколько раз в ходе машинного эксперимента с некоторой системой S .

При отсутствии возможности получения каких-либо априорных сведений о вероятности p использование понятия абсолютной точности теряет смысл. Действительно, можно, например, предварительно задать точность результатов моделирования $\varepsilon=0,01$, а искомая p в результате окажется хотя бы на порядок ниже, т. е. $p \leq 0,001$. В таких случаях целесообразно задавать относительную

точность результатов моделирования ε_0 . Тогда соотношение (6.9) примет вид

$$N = t_\varphi^2 p(1-p)/(p^2 \varepsilon_0^2) = t_\varphi^2 (1-p)/(\varepsilon_0^2 p). \quad (6.10)$$

Соотношение (6.10) наглядно иллюстрирует специфику статистического моделирования систем, выражающуюся в том, что для оценивания малых вероятностей p с высокой точностью необходимо очень большое число реализаций N . В практических случаях для оценивания вероятностей порядка 10^{-k} целесообразно количество реализаций выбирать равным 10^{k+1} . Очевидно, что даже для сравнительно простых систем метод статистического моделирования приводит к большим затратам машинного времени.

Другим распространенным случаем в практике машинных экспериментов с моделью M_x является необходимость оценки показателя эффективности E системы S по результатам определения среднего значения некоторой случайной величины. Пусть случайная величина ξ имеет математическое ожидание a и дисперсию σ^2 . В реализации с номером i она принимает значение x_i . В качестве оценки математического ожидания a используется среднее арифметическое $\bar{x} = (1/N) \sum_{i=1}^N x_i$.

В силу центральной предельной теоремы теории вероятностей при больших значениях N среднее арифметическое \bar{x} будет иметь распределение, близкое к нормальному с математическим ожиданием a и дисперсией σ^2/N . Для математического ожидания a точность оценки $\varepsilon = t_\varphi \sigma / \sqrt{N}$, а количество реализаций

$$N = t_\varphi^2 \sigma^2 / \varepsilon^2, \text{ или } N = t_\varphi^2 \varepsilon^2 / (\varepsilon_0^2 a^2). \quad (6.11)$$

Аналогично, если в качестве показателя эффективности E системы S выступает дисперсия σ^2 , а в качестве ее оценки используется величина S^2 , то математическое ожидание и дисперсия соответственно будут

$$M[S^2] = (N-1)\sigma^2/N; \quad D[S^2] = (\mu_4 - \sigma^4)/N,$$

где μ_4 — центральный момент четвертого порядка случайной величины.

Для дисперсии σ^2 точность оценки $\varepsilon = t_\varphi \sqrt{(\mu_4 - \sigma^4)/N}$.

Отсюда количество реализаций будет

$$N = t_\varphi^2 (\mu_4 - \sigma^4) / \varepsilon^2, \text{ или } N = t_\varphi^2 [(\mu_4 / \sigma^4) - 1] / \varepsilon_0^2. \quad (6.12)$$

Для частного случая, когда случайная величина имеет нормальное распределение $\mu_4 = 3\sigma^4$, получим $N = t_\varphi^2 2\sigma^4 / \varepsilon^2 = 2t_\varphi^2 / \varepsilon_0^2$.

Таким образом, на основании соотношений (6.9) — (6.12) можно

сделать вывод, что количество реализаций при статистическом моделировании существенно зависит от дисперсии оцениваемой случайной величины. Поэтому выгодно выбирать такие оцениваемые показатели эффективности E системы S , которые имеют малые дисперсии.

Проблема уменьшения дисперсии оценок характеристик процесса функционирования моделируемых систем. Таким образом, с проблемой выбора количества реализаций при обеспечении необходимой точности и достоверности результатов машинного эксперимента тесно связана и третья проблема, а именно проблема уменьшения дисперсии. В настоящее время существуют методы, позволяющие при заданном числе реализаций увеличить точность оценок, полученных на машинной модели M_n , и, наоборот, при заданной точности оценок сократить необходимое число реализаций при статистическом моделировании. Эти методы используют априорную информацию о структуре и поведении моделируемой системы S и называются методами уменьшения дисперсии.

Рассмотрим в качестве иллюстрации метод коррелированных реализаций (выборок), используемый в задачах сравнения двух или более альтернатив. При исследовании и проектировании системы S всегда происходит сравнение вариантов S_i , $i = \overline{1, k}$, отличающихся друг от друга структурой, алгоритмами поведения и параметрами.

Независимо от того, как организуется выбор наилучшего варианта системы S (простым перебором результатов моделирования системы S_i или с помощью автоматизированной процедуры поиска), элементарной операцией при этом является равнение статистически усредненных критериев интерпретации [18, 21, 29, 33, 53].

Сравнимые статистические показатели E_i вариантов моделируемой системы S_i , $i = \overline{1, k}$, полученные на машинной модели M_n , можно записать в виде средних значений $E_i = M[q_i]$, $i = \overline{1, k}$, критериев η_i , характеризующих систему S_i , или в виде средних значений функции этих критериев $f_j(q_i)$, $i = \overline{1, k}$, $j = \overline{1, L}$. Например, если

$$f_j(q_i) = \begin{cases} 1 & \text{система } S_i \text{ удовлетворяет требованиям;} \\ 0 & \text{в противном случае,} \end{cases}$$

то показатели E_i являются вероятностями нормальной работы системы S_i . Если

$$f_j(q_i) = (\Delta q_i - M[\Delta q_i])^2, \quad i = \overline{1, k}, \quad j = \overline{1, L},$$

то показатель E_i является дисперсией значения контролируемой величины и т. д. Здесь $\Delta q_i = |q_i - q_{ni}|$ — отклонение значения контролируемой для системы S_i величины q_i от истинной q_{ni} .

В дальнейшем, поскольку при сравнении характеристик, полученных на машинной модели M_m , всегда рассматриваются два конкурирующих варианта моделируемой системы, будем сопоставлять только две системы: S_1 и S_2 . Существенной особенностью операции сравнения вариантов систем S_1 и S_2 является повышение требований к точности статистических оценок \bar{E}_1, \bar{E}_2 показателей E_1, E_2 при уменьшении разности $\Delta E = |E_1 - E_2|$. Это обстоятельство требует разработки специальных приемов получения статистически независимых оценок для уменьшения дисперсии.

Рассмотрим наиболее характерные случаи, имеющие место при имитационных экспериментах, когда в качестве оценок выступают средние значения, вероятности и дисперсии [29, 53].

Если полученные в результате имитационного эксперимента с вариантами модели системы S_1 и S_2 оценки \bar{a}_1, \bar{a}_2 средних значений критериев $q_1, q_2, a_1 = M[q_1], a_2 = M[q_2]$ имеют дисперсии $D[\bar{a}_1], D[\bar{a}_2]$ и коэффициент корреляции оценок \bar{a}_1, \bar{a}_2 равен $R[\bar{a}_1, \bar{a}_2]$, то дисперсию погрешности оценки $\bar{d} = \bar{a}_1 - \bar{a}_2$ разности $d = a_1 - a_2$ можно найти из соотношения

$$D[\bar{d}] = D[\bar{a}_1 - \bar{a}_2] = D[\bar{a}_1] + D[\bar{a}_2] - 2R[\bar{a}_1, \bar{a}_2]\sigma_1\sigma_2, \quad (6.13)$$

где $\sigma_1 = \sqrt{D[\bar{a}_1]}, \sigma_2 = \sqrt{D[\bar{a}_2]}$ — средние квадратические отклонения оценок.

При независимом моделировании вариантов системы с использованием различных реализаций псевдослучайных последовательностей коэффициент корреляции оценок $R[\bar{a}_1, \bar{a}_2] = 0$ и $D_n[\bar{d}] = D[\bar{a}_1] + D[\bar{a}_2]$.

При моделировании удается получить положительный коэффициент корреляции $R[\bar{a}_1, \bar{a}_2] > 0$, т. е. $D[\bar{d}] < D_n[\bar{d}]$, когда при имитационных экспериментах с вариантами системы S_1 и S_2 используются, например, одни и те же псевдослучайные последовательности. Рассмотренные соотношения для дисперсии $D[\bar{d}]$ не связаны со специальными предположениями о способе получения оценок \bar{a}_1, \bar{a}_2 .

Пример 6.8. Рассмотрим полученные при машинном моделировании реализации $y_1(r\Delta t), y_2(r\Delta t), r = \overline{0, N}$, критериев q_1, q_2 как выборку из двумерного векторного стационарного процесса $\vec{q}(t) = \|q_1(t), q_2(t)\|$ со средним значением $\|a_1, a_2\|$ и матричной корреляционной функцией

$$B(\tau) = \begin{pmatrix} B_{11}(\tau) & B_{12}(\tau) \\ B_{21}(\tau) & B_{22}(\tau) \end{pmatrix},$$

где $B_{ii}(\tau)$ — автокорреляционная функция $i = 1, 2; B_{ij}(\tau) = B_{ji}(\tau)$ — взаимная корреляционная функция компонент $\vec{q}(t), i, j = 1, 2, i \neq j$

Тогда можно показать, что

$$D[\bar{d}] = [1/(N+1)](B_{11}(0) + B_{22}(0) - 4B_{12}(0) + \\ + \sum_{r=1}^{N+1} [1-r/(N+1)](B_{11}(r\Delta t) + \{B_{22}(r\Delta t) - 2[B_{12}(r\Delta t) + B_{21}(r\Delta t)]\})).$$

Эту формулу применяют для расчета точности оценки \vec{d} при заданной матричной корреляционной функции $B(\tau)$.

Вероятности p_1, p_2 событий A_1, A_2 , характеризующих сравниваемые варианты модели систем S_1 и S_2 , можно представить как средние значения двоичных случайных величин q_1, q_2 с распределением вероятностей $P\{q_1=1\}=p_1; P\{q_1=0\}=1-p_1; P\{q_2=1\}=p_2; P\{q_2=0\}=1-p_2$.

Поэтому для оценки разности вероятностей $\Delta p = p_1 - p_2 = M[q_1] - M[q_2]$ можно использовать все выражения, полученные ранее при сравнении средних значений, видоизменив в них обозначения с учетом того, что двумерное распределение вектора (q_1, q_2) , описывающее зависимость между событиями A_1, A_2 , имеет вид $P\{q_1=1, q_2=1\}=P(A_1, A_2)=p_A; P\{q_1=0, q_2=0\}=P(\bar{A}_1, \bar{A}_2)=p_B; P\{q_1=1, q_2=0\}=P(A_1, \bar{A}_2)=p_C; P\{q_1=0, q_2=1\}=P(\bar{A}_1, A_2)=p_D$, причем $p_A + p_C = p_1, p_A + p_D = p_2$. В частности, для повторной выборки объемом N получим, что оценка

$$\Delta \bar{p} = \bar{p}_1 - \bar{p}_2 = m_1/N - m_2/N,$$

где m_1, m_2 — количества наступлений событий A_1, A_2 , полученных при независимых прогнозах модели. Учитывая, что между q_1, q_2 ковариация $B_{12} = p_A - p_1 p_2$, найдем дисперсию оценки

$$D[\Delta \bar{p}] = [p_C + p_D - (p_C - p_D)^2]/N,$$

что следует из (6.13).

Если в процессе проведения имитационных экспериментов с моделью фиксируются эмпирические частоты \bar{p}_C, \bar{p}_D событий C, D , то для дисперсии $D[\Delta \bar{p}]$ при достаточно большом N можно воспользоваться несмещенной оценкой

$$\tilde{D}[\Delta \bar{p}] = [\bar{p}_C + \bar{p}_D - (\bar{p}_C - \bar{p}_D)^2]/(N-1).$$

И наконец, рассмотрим случай, когда в качестве оценки вариантов систем S_1 и S_2 выступает дисперсия. В этом случае оценка $\Delta \tilde{D}$ разности $\Delta D = D_1 - D_2$ дисперсией критериев q_1, q_2 вычисляется по независимым реализациям вектора (q_1, q_2) с помощью формулы $\Delta \tilde{D} = \tilde{D}_1 - \tilde{D}_2$, где \tilde{D}_1, \tilde{D}_2 — эмпирические дисперсии критериев q_1, q_2 , рассчитываемые по формуле

$$\tilde{D}_q = \sum_{i=1}^N (q_i - M[q])^2 / (N-1).$$

Для оценки ΔD дисперсия

$$D[\Delta\tilde{D}] = D[\tilde{D}_1] + D[\tilde{D}_2] - 2B[\tilde{D}_1, \tilde{D}_2],$$

где дисперсии эмпирических дисперсий $D[\tilde{D}_1]$, $D[\tilde{D}_2]$ вычисляются по формуле

$$D[\tilde{D}_q] = N[(\mu_4 - D_4^2)/N - 2(\mu_4 - 2D_4^2)/N^2 + (\mu_4 - 3D_4^2)/N^3]/(N-1),$$

где $\mu_4 = M_4[q] = M[(q - M[q])^4]$ — центральный момент распределения четвертого порядка.

Ковариация

$$B[\tilde{D}_1, \tilde{D}_2] \approx M[(q_1 - M[q_1])^2 (q_2 - M[q_2])^2]/N.$$

Пример 6.9. Пусть сравниваемые критерии q_1 , q_2 имеют нормальное распределение

$$f(y) = e^{-\frac{(y-a)^2}{2\sigma^2}} / \sqrt{2\pi\sigma}, \quad a = M[q], \quad \sigma^2 = D[q],$$

тогда

$$D[\Delta\tilde{D}] = 2(D_1^2 + D_2^2 - 2B_{12}^2)/(N-1),$$

где B_{12} — ковариация. Использование зависимых испытаний дает выигрыш в точности сравнения дисперсий ($B_{12} \neq 0$) независимо от знака корреляции. Воспользовавшись оценкой

$$\tilde{B}[\Delta\tilde{D}] = 2(\tilde{D}_1^2 + \tilde{D}_2^2 - 2\tilde{B}_{12})/(N-1),$$

можно организовать последовательную процедуру сравнения дисперсий для вариантов системы S_1 и S_2 .

Таким образом, при таком подходе к уменьшению дисперсии задача состоит в специальном построении моделирующего алгоритма системы S , позволяющего получить положительную корреляцию, например, за счет управления генерацией случайных величин. Вопрос об эффективности использования метода уменьшения дисперсии может быть решен только с учетом необходимости дополнительных затрат машинных ресурсов (времени и памяти) на реализацию подхода, т. е. теоретическое уменьшение затрат машинного времени на моделирование вариантов системы (при той же точности результатов) должно быть проверено на сложность машинной реализации модели.

Проблема выбора правил автоматической остановки имитационного эксперимента с моделями системы. И наконец, последней из проблем, возникающих при тактическом планировании имитационных экспериментов, рассмотрим проблему выбора правил автоматической остановки имитационного эксперимента. Простейший способ решения проблемы — задание требуемого количества реализаций N (или длины интервала моделирования T). Однако такой детерминированный подход неэффективен, так как в его основе лежат достаточно грубые предположения о распределении

выходных переменных, которые на этапе тактического планирования являются неизвестными. Другой способ — задание доверительных интервалов для выходных переменных и остановка прогона машинной модели M_n при достижении заданного доверительного интервала, что позволяет теоретически приблизить время прогона к оптимальному. При практической реализации введение в модель M_n правил остановки и операций вычисления доверительных интервалов увеличивает машинное время, необходимое для получения одной выборочной точки при статистическом моделировании.

Правила автоматической остановки могут быть включены в машинную модель такими способами: 1) путем двухэтапного проведения прогона, когда сначала делается пробный прогон из N^* реализаций, позволяющий оценить необходимое количество реализаций N (причем если $N^* \geq N$, то прогон можно закончить, в противном случае необходимо набрать еще $N - N^*$ реализаций); 2) путем использования последовательного анализа для определения минимально необходимого количества реализаций N , которое рассматривается при этом как случайная величина, зависящая от результатов $N - 1$ предыдущих реализаций (наблюдений, испытаний) машинного эксперимента.

Рассмотрим особенности последовательного планирования машинных экспериментов, построенных на последовательном анализе. В последовательном анализе объем выборки не фиксирован, а после i -го наблюдения принимается одно из следующих решений: принять данную гипотезу, отвергнуть гипотезу, продолжить испытания, т. е. повторить наблюдения еще раз. Благодаря такому подходу можно объем выборки существенно уменьшить по сравнению со способами остановки, использующими фиксированный объем выборки. Таким образом, последовательное планирование машинного эксперимента позволяет минимизировать объем выборки в эксперименте, необходимой для получения требуемой при исследовании системы S информации. Построив критерий, можно на каждом шаге решать вопрос либо о принятии нулевой гипотезы H_0 , либо о принятии альтернативной гипотезы H_1 , либо о продолжении машинного эксперимента. Последовательное планирование машинного эксперимента использует принцип максимального правдоподобия и последовательные проверки статистических гипотез [18, 21, 33].

Пусть распределение генеральной совокупности характеризуется функцией плотности вероятностей с неизвестным параметром $Y = f(y, \theta)$. Определяются нулевая и альтернативная гипотезы $H_0: \theta = \theta_0$ и $H_1: \theta = \theta_1$. Гипотезы проверяют на основании выборки нарастающего объема m . Можно записать: вероятность получения данной выборки $P_{0m} = f(y_1, \theta_0) f(y_2, \theta_0) \dots f(y_m, \theta_0)$ при условии, что верна гипотеза H_0 (правдоподобная выборка); вероятность получения выборки $P_{1m} = f(y_1, \theta_1) f(y_2, \theta_1) \dots f(y_m, \theta_1)$ при условии верности

гипотезы H_1 . Процедура проверки строится на отношении правдоподобия P_{1m}/P_{0m} .

Последовательный критерий отношения вероятностей строится следующим образом. На каждом шаге машинного эксперимента определяются P_{1m} и P_{0m} , а также проверяется условие:

$$\text{если } \frac{P_{1m}}{P_{0m}} \begin{cases} \leq B, \text{ то принимается } H_0; \\ < A \\ > B, \\ \geq A, \text{ то принимается } H_1; \end{cases} \text{ то эксперимент продолжается;}$$

где $0 < B < 1$, $A > 1$, $m = \overline{1, N}$.

Для сходимости критерия необходимо, чтобы $A \leq (1 - \beta)/\alpha$, $B \geq \beta/(1 - \alpha)$, где α — вероятность ошибки первого рода; β — вероятность ошибки второго рода.

Данный метод позволяет снизить среднее число реализаций в машинных экспериментах по сравнению с использованием фиксированных объемов выборки (при одинаковых вероятностях ошибок). Примером применения метода может служить проверка гипотезы о среднем значении величины, распределенной по нормальному закону.

Пример 6.10. Пусть для случайной величины y известна дисперсия σ^2 и неизвестно среднее μ . При этом нулевая гипотеза $H_0: \mu = \theta_0$, альтернативная $H_1: \mu = \theta_1$. Если H_0 верна, то вероятность ее отвергнуть равна α . Если верна гипотеза H_1 , то вероятность принять ее равна β . В случае $\theta_0 < \mu < \theta_1$ ни одна из гипотез не принимается.

Для нормального распределения

$$P_{1m} = e^{-\frac{1}{2\sigma^2} \sum_{i=1}^m (y_i - \theta_1)^2} / (\sqrt{2\pi\sigma})^m,$$

$$P_{0m} = e^{-\frac{1}{2\sigma^2} \sum_{i=1}^m (y_i - \theta_0)^2} / (\sqrt{2\pi\sigma})^m,$$

$$P_{1m}/P_{0m} = e^{-\frac{1}{2\sigma^2} \left[\sum_{i=1}^m (y_i - \theta_1)^2 - \sum_{i=1}^m (y_i - \theta_0)^2 \right]}.$$

Критерий проверки гипотезы строится по следующему правилу: если

$$\ln B = b < \frac{1}{2\sigma^2} \left[\sum_{i=1}^m (y_i - \theta_1)^2 - \sum_{i=1}^m (y_i - \theta_0)^2 \right] < a = \ln A,$$

то наблюдение продолжается.

Можно упростить процедуру, если использовать логарифмическую функцию правдоподобия. В этом случае

$$\ln(P_{1m}/P_{0m}) = [(\theta_1 - \theta_0)/\sigma^2] \left[\sum_{i=1}^m y_i - 0,5m(\theta_1 + \theta_0) \right],$$

$$a = \ln A = \ln [(1 - \beta)/\alpha],$$

$$b = \ln B = \ln [\beta/(1 - \alpha)].$$

Тогда на каждом шаге m проверяется выполнение неравенств: если

$$\sum_{i=1}^m y_i \geq a\sigma^2/(\theta_1 - \theta_0) + 0,5m(\theta_1 + \theta_0),$$

то принимается H_0 ;
если

$$\sum_{i=1}^m y_i \leq b\sigma^2/(\theta_1 - \theta_0) + 0,5m(\theta_1 + \theta_0),$$

то принимается H_1 ;
если

$$\frac{b\sigma^2}{\theta_1 + \theta_0} + 0,5m(\theta_1 + \theta_0) < \sum_{i=1}^m y_i < \frac{a\sigma^2}{\theta_1 + \theta_0} + 0,5m(\theta_1 + \theta_0),$$

то машинный эксперимент продолжается.

Для математического ожидания числа наблюдений при условии верности H_1 и H_2 соответственно можно записать

$$M[N/H_0] = [b(1 - \alpha) + a\alpha]/M[z/H_0],$$

$$M[N/H_1] = [b\beta + a(1 - \beta)]/M[z/H_1],$$

где N — число наблюдений; $z = \ln [f(y, \theta_1)/f(y, \theta_0)] = -[(y - \theta_1)^2 + (y - \theta_0)^2]/(2\sigma^2)$.

Можно записать $M[z/H_0] = (\theta_1 - \theta_0)^2/(2\sigma^2)$, $M[z/H_1] = (\theta_1 - \theta_0)^2/(2\sigma^2)$, так как $M[y] = \theta_0$ для гипотезы H_0 и $M[y] = \theta_1$ для гипотезы H_1 .

Тогда

$$M[N/H_0] = -[b(1 - \alpha) + a\alpha] 2\sigma^2/(\theta_1 - \theta_0)^2,$$

$$M[N/H_1] = [b\beta + a(1 - \beta)] 2\sigma^2/(\theta_1 - \theta_0)^2.$$

Применение данного метода по сравнению с фиксированным объемом выборки N дает уменьшение числа реализаций при статистическом моделировании более чем в два раза.

Для проверки гипотезы о среднем для случайных величин с нормальным законом распределения, неизвестным средним μ и неизвестной дисперсией σ можно использовать следующую процедуру. Проверяют гипотезы $H_0: \mu < \mu_0$ и $H_1: \mu > \mu_0$. Необходимо, чтобы вероятность отвергнуть H_0 при $\mu \leq \mu_0$ была $P \leq \alpha$ и вероятность принять H_0 при $\mu > \mu + \Delta$ была $P \leq \beta$.

На первом шаге берут выборку размером m и вычисляют выборочную дисперсию

$$S^2 = \sum_{j=1}^m (y_j - \bar{y})^2 / (m-1);$$

здесь число m выбрано таким, чтобы выполнялось условие

$$a(\delta) \leq 1,25 \log(1/\delta),$$

где $a(\delta) = [(1/\delta)^{2f} - 1]^{1/2}$, $\delta = \min(\alpha, \beta)$, $f = m - 1$.

Затем последовательно проводят по одному эксперименту. При выполнении условия

$$\sum_{j=1}^m (y_j - \mu_0 - \Delta/2) > S^2 a(\alpha) / (2d) - N(\Delta/2 - d)$$

эксперимент прекращают и гипотезу H_0 отвергают.

Гипотезу H_0 принимают, если

$$\sum_{j=1}^m (y_j - \mu_0 - \Delta/2) < N(\Delta/2 - d) - S^2 a(\beta) / (2d),$$

где $d = 3\Delta/8$.

Таким образом, чем сложнее машинная модель M_m , тем важнее этап тактического планирования машинного эксперимента, выполняемый непосредственно перед моделированием на ЭВМ системы S . Процесс планирования машинных экспериментов с моделью M_m итерационен, т. е. при уточнении некоторых свойств моделируемой системы S этапы стратегического и тактического планирования экспериментов могут чередоваться.

Контрольные вопросы

- 6.1. Каковы характерные особенности машинного эксперимента по сравнению с другими видами экспериментов?
- 6.2. Какие виды факторов бывают в имитационном эксперименте с моделями систем?
- 6.3. Что называется полным факторным экспериментом?
- 6.4. Какова цель стратегического планирования машинных экспериментов?
- 6.5. Какие проблемы стратегического планирования машинных экспериментов с моделями систем являются основными?
- 6.6. Какова цель тактического планирования машинных экспериментов?
- 6.7. Что называется точностью и достоверностью результатов моделирования систем на ЭВМ?
- 6.8. Как повысить точность результатов статистического моделирования системы в условиях ограниченности ресурсов инструментальной ЭВМ?

ГЛАВА 7

ОБРАБОТКА И АНАЛИЗ РЕЗУЛЬТАТОВ МОДЕЛИРОВАНИЯ СИСТЕМ

Концепция статистического моделирования систем в реализационном плане неразрывно связана с ограниченностью ресурсов инструментальных ЭВМ. Поэтому при рассмотрении теоретических проблем машинной имитации, относящихся в основном к разделу математической статистики, необходимо учитывать особенности и возможности текущей обработки экспериментальной информации на ЭВМ. Успех имитационного эксперимента с моделью системы существенным образом зависит от правильного решения вопросов обработки и последующего анализа и интерпретации результатов моделирования. Особенно важно решить проблему текущей обработки экспериментальной информации при использовании модели для целей автоматизации проектирования систем.

7.1. ОСОБЕННОСТИ ФИКСАЦИИ И СТАТИСТИЧЕСКОЙ ОБРАБОТКИ РЕЗУЛЬТАТОВ МОДЕЛИРОВАНИЯ СИСТЕМ НА ЭВМ

После того как машинный эксперимент спланирован, необходимо предусмотреть меры по организации эффективной обработки и представления его результатов. Вообще, проблема статистической обработки результатов эксперимента с моделью тесно связана с рассмотренными в гл. 6 проблемами стратегического и тактического планирования. Но важность этой проблемы и наличие специфики в машинной обработке результатов моделирования выделяют ее в самостоятельную проблему. При этом надо иметь в виду, что применяемые на практике методы обработки результатов моделирования составляют только небольшую часть арсенала математической статистики [7, 11, 18, 21 25, 33].

Особенности машинных экспериментов. При выборе методов обработки существенную роль играют три особенности машинного эксперимента с моделью системы S .

1. Возможность получать при моделировании системы S на ЭВМ большие выборки позволяет количественно оценить характеристики процесса функционирования системы, но превращает в серьезную проблему хранение промежуточных результатов моделирования. Эту проблему можно решить, используя рекуррентные алгоритмы обработки, когда оценки вычисляются по ходу моделирования, причем большой объем выборки дает возможность пользоваться при этом достаточно простыми для расчетов на ЭВМ асимптотическими формулами.

2. Сложность исследуемой системы S при ее моделировании на ЭВМ часто приводит к тому, что априорное суждение о характеристиках процесса функционирования системы, например о типе ожидаемого распределения выходных переменных, является невозможным. Поэтому при моделировании систем широко используются непараметрические оценки и оценки моментов распределения.

3. Блочность конструкции машинной модели M_m и раздельное исследование блоков связаны с программной имитацией входных переменных для одной частичной модели по оценкам выходных переменных, полученных на другой частичной модели. Если ЭВМ, используемая для моделирования, не позволяет воспользоваться переменными, записанными на внешние носители, то следует представить эти переменные в форме, удобной для построения алгоритма их имитации.

Методы оценки. Рассмотрим наиболее удобные для программной реализации методы оценки распределений и некоторых их моментов при достаточно большом объеме выборки (числе реализаций N). Математическое ожидание и дисперсия случайной величины ξ соответственно имеют вид

$$\begin{aligned}\mu_{\xi} &= M[\xi] = \int_{-\infty}^{\infty} xf(x) dx; \quad \sigma_{\xi}^2 = D[\xi] = M[(x - \mu_{\xi})^2] = \\ &= \int_{-\infty}^{\infty} (x - \mu_{\xi})^2 f(x) dx,\end{aligned}$$

где $f(x)$ — плотность распределения случайной величины ξ , принимающей значения x .

При проведении имитационного эксперимента со стохастической моделью системы S определить эти моменты нельзя, так как плотность распределения, как правило, априори неизвестна. Поэтому при обработке результатов моделирования приходится довольствоваться лишь некоторыми оценками моментов, полученными на конечном числе реализаций N . При независимых наблюдениях значений случайной величины ξ в качестве таких оценок используются

$$\bar{x} = \tilde{\mu}_{\xi} = (1/N) \sum_{i=1}^N x_i; \quad S_b^2 = \tilde{\sigma}_{\xi}^2 = (1/N) \sum_{i=1}^N (x_i - \bar{x})^2,$$

где \bar{x} и S_b^2 — выборочное среднее и выборочная дисперсия соответственно. Знак \sim над $\tilde{\mu}_{\xi}$ и $\tilde{\sigma}_{\xi}^2$ означает, что эти выборочные моменты используются в качестве оценок математического ожидания μ_{ξ} и дисперсии σ_{ξ}^2 .

К качеству оценок, полученных в результате статистической обработки результатов моделирования, предъявляются следующие требования [7, 11, 25]:

1) несмещенность оценки, т. е. равенство математического ожидания оценки определяемому параметру $M[\tilde{g}] = g$, где \tilde{g} — оценка переменной (параметра) g ;

2) эффективность оценки, т. е. минимальность среднего квадрата ошибки данной

оценки $M[\bar{g}_1 - g] \leq M[(\bar{g}_1 - g)^2]$, где \bar{g}_1 — рассматриваемая оценка; g_1 — любая другая оценка;

3) состоятельность оценки, т. е. сходимость по вероятности при $N \rightarrow \infty$ к оцениваемому параметру

$$\lim_{N \rightarrow \infty} P\{|\bar{g} - g| \geq \varepsilon\} = 0, \varepsilon > 0,$$

либо, учитывая неравенство Чебышева, достаточное (но не обязательно необходимое) условие выполнения этого неравенства заключается в том, чтобы $\lim_{N \rightarrow \infty} M[(\bar{g} - g)^2] = 0$.

$N \rightarrow \infty$ Рассмотрим оценку выборочного среднего значения \bar{x} . Математическое ожидание выборочного среднего значения \bar{x} составит

$$M[\bar{x}] = M\left[\frac{1}{N} \sum_{i=1}^N x_i\right] = \frac{1}{N} M\left[\sum_{i=1}^N x_i\right] = \frac{N\mu_x}{N} = \mu_x,$$

т. е. оценка $\bar{\mu}_x = \bar{x}$ является несмещенной.

С учетом независимости значений x_i средний квадрат ошибки

$$M[(\bar{x} - \mu_x)^2] = M\left[\frac{1}{N^2} \sum_{i=1}^N (x_i - \mu_x)^2\right] = \frac{1}{N^2} M\left[\left(\sum_{i=1}^N (x_i - \mu_x)^2\right)\right] = \frac{N\sigma_x^2}{N^2} = \frac{\sigma_x^2}{N},$$

т. е. оценка $\bar{\mu}_x = \bar{x}$ состоятельна. Можно показать, что эта оценка также и эффективна.

Рассмотрим оценку выборочной дисперсии S_x^2 . Математическое ожидание выборочной дисперсии

$$M[S_x^2] = M\left[\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2\right] = \frac{1}{N} M\left[\sum_{i=1}^N (x_i - \bar{x})^2\right].$$

Учитывая, что

$$\sum_{i=1}^N (x_i - \bar{x})^2 = \sum_{i=1}^N (x_i - \mu_x)^2 - N(\bar{x} - \mu_x)^2,$$

$$M[(x_i - \mu_x)^2] = \sigma_x^2, \quad M[(\bar{x} - \mu_x)^2] = \sigma_x^2/N,$$

получим $M[S_x^2] = (N\sigma_x^2 - \sigma_x^2)/N = (N-1)\sigma_x^2/N$, т. е. оценка $\bar{\sigma}_x^2 = S_x^2$ является смещенной. Можно показать, что эта оценка состоятельна и эффективна.

Несмещенную оценку дисперсии σ_x^2 можно получать, вычисляя выборочную дисперсию вида

$$S = \sigma_x^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2.$$

Эта оценка также удовлетворяет условиям эффективности и состоятельности.

Статистические методы обработки. Рассмотрим некоторые особенности статистических методов, используемых для обработки результатов моделирования системы S . Для случая исследования сложных систем при большом числе реализаций N в результате моделирования на ЭВМ получается значительный объем информации о состояниях процесса функционирования системы. Поэтому необходимо так организовать в процессе вычислений фиксацию и обработку результатов моделирования, чтобы оценки для ис-

комых характеристик формировались постепенно по ходу моделирования, т. е. без специального запоминания всей информации о состояниях процесса функционирования системы S .

Если при моделировании процесса функционирования конкретной системы S учитываются случайные факторы, то и среди результатов моделирования присутствуют случайные величины. В качестве оценок для искомых характеристик рассчитывают средние значения, дисперсии, корреляционные моменты и т. д.

Пусть в качестве искомой величины фигурирует вероятность некоторого события A . В качестве оценки для искомой вероятности $p = P(A)$ используется частота наступления события m/N , где m — число случаев наступления события A ; N — число реализаций. Такая оценка вероятности появления события A является состоятельной, несмещенной и эффективной. В случае необходимости получения оценки вероятности в памяти ЭВМ при обработке результатов моделирования достаточно накапливать лишь число m (при условии, что N задано заранее).

Аналогично при обработке результатов моделирования можно подойти к оценке вероятностей возможных значений случайной величины, т. е. закона распределения. Область возможных значений случайной величины η разбивается на n интервалов. Затем накапливается количество попаданий случайной величины в эти интервалы m_k , $k = \overline{1, n}$. Оценкой для вероятности попадания случайной величины в интервал с номером k служит величина m_k/N . Таким образом, при этом достаточно фиксировать n значений m_k при обработке результатов моделирования на ЭВМ.

Для оценки среднего значения случайной величины η накапливается сумма возможных значений случайной величины y_k , $k = \overline{1, N}$, которые она принимает при различных реализациях. Тогда среднее значение

$$\bar{y} = (1/N) \sum_{k=1}^N y_k.$$

При этом ввиду несмещенности и состоятельности оценки

$$M[\bar{y}] = M[\eta] = \mu_\eta; D[\bar{y}] = D[\eta]/N = \sigma_\eta^2/N.$$

В качестве оценки дисперсии случайной величины η при обработке результатов моделирования можно использовать

$$S_\eta^2 = \sum_{k=1}^N (y_k - \bar{y})^2/N.$$

Непосредственное вычисление дисперсии по этой формуле нерационально, так как среднее значение \bar{y} изменяется в процессе накопления значений y_k . Это приводит к необходимости запоминания всех N значений y_k . Поэтому более рационально организовать фик-

саию результатов моделирования для оценки дисперсии с использованием следующей формулы:

$$D[\eta] = \sigma_{\eta}^2 = \left[\sum_{k=1}^N y_k^2 - \left(\sum_{k=1}^N y_k \right)^2 / N \right] / (N-1).$$

Тогда для вычисления дисперсии достаточно накапливать две суммы: значений y_k и их квадратов y_k^2 .

Для случайных величин ξ и η с возможными значениями x_k и y_k корреляционный момент

$$k_{\xi\eta} = \left[\sum_{k=1}^N (x_k - \bar{x})(y_k - \bar{y}) \right] / N \text{ или}$$

$$k_{\xi\eta} = \left(\sum_{k=1}^N x_k y_k - (1/N) \sum_{k=1}^N x_k \sum_{k=1}^N y_k \right) / (N-1).$$

Последнее выражение вычисляется при запоминании в процессе моделирования небольшого числа значений.

Если при моделировании системы S искомыми характеристиками являются математическое ожидание и корреляционная функция случайного процесса $y(t)$ [в интервале моделирования $(0, T)$], то для нахождения оценок этих величины указанный интервал разбивают на отрезки с постоянным шагом Δt и накапливают значения процесса $y_k(t)$ для фиксированных моментов времени $t = t_m = m\Delta t$.

При обработке результатов моделирования математическое ожидание и корреляционную функцию запишем так:

$$\bar{y}(t_m) = \sum_{k=1}^N y_k(t_m) / N; \quad \tilde{B}(u, z) = \sum_{k=1}^N (y_k(u) - \bar{y}(u))(y_k(z) - \bar{y}(z)) / (N-1),$$

где u и z пробегает все значения t_m .

Для уменьшения затрат машинных ресурсов на хранение промежуточных результатов последнее выражение также целесообразно привести к следующему виду:

$$\tilde{B}(u, z) = \left(\sum_{k=1}^N y_k(u) y_k(z) - (1/N) \sum_{k=1}^N y_k(u) \sum_{k=1}^N y_k(z) \right) / (N-1).$$

Отметим особенности фиксации и обработки результатов моделирования, связанные с оценкой характеристик стационарных случайных процессов, обладающих эргодическим свойством. Пусть рассматривается процесс $y(t)$. Тогда с учетом этих предположений поступают в соответствии с правилом: среднее по времени равно среднему по множеству. Это означает, что для оценки искомым характеристик выбирается одна достаточно продолжительная ре-

ализация процесса $y(t)$, для которой целесообразно фиксировать результаты моделирования. Для рассматриваемого случая запишем математическое ожидание и корреляционную функцию процесса:

$$\bar{y} = \lim_{T \rightarrow \infty} (1/T) \int_0^T y(t) dt; B(\tau) = \lim_{T \rightarrow \infty} [1/(T-\tau)] \int_0^{T-\tau} y(t)y(t+\tau) dt - \bar{y}^2.$$

На практике при моделировании на ЭВМ системы S интервал $(0, T)$ оказывается ограниченным и, кроме того, значения $y(t)$ удается определить только для конечного набора моментов времени t_m . При обработке результатов моделирования для получения оценок \bar{y} и $B(\tau)$ используем приближенные формулы

$$\bar{y} \approx (\Delta t/T) \sum_{m=1}^{T/\Delta t} y(t_m); B(\tau) \approx [\Delta t/(T-\tau)] \sum_{m=1}^{(T-\tau)/\Delta t} y(t_m)y(t_m+\tau) - \bar{y}^2,$$

которые целесообразно преобразовать к виду, позволяющему эффективно организовать порядок фиксации и обработки результатов моделирования на ЭВМ [4].

Задачи обработки результатов моделирования. При обработке результатов машинного эксперимента с моделью M_m наиболее часто возникают следующие задачи: определение эмпирического закона распределения случайной величины, проверка однородности распределений, сравнение средних значений и дисперсий переменных, полученных в результате моделирования, и т. д. Эти задачи с точки зрения математической статистики являются типовыми задачами по проверке статистических гипотез.

Задача определения эмпирического закона распределения случайной величины наиболее общая из перечисленных, но для правильного решения требует большого числа реализаций N . В этом случае по результатам машинного эксперимента находят значения выборочного закона распределения $F_n(y)$ (или функции плотности $f_n(y)$) и выдвигают нулевую гипотезу H_0 , что полученное эмпирическое распределение согласуется с каким-либо теоретическим распределением. Проверяют эту гипотезу H_0 с помощью статистических критериев согласия Колмогорова, Пирсона, Смирнова и т. д., причем необходимую в этом случае статистическую обработку результатов ведут по возможности в процессе моделирования системы S на ЭВМ.

Для принятия или опровержения гипотезы выбирают некоторую случайную величину U , характеризующую степень расхождения теоретического и эмпирического распределения, связанную с недостаточностью статистического материала и другими случайными причинами. Закон распределения этой случайной величины зависит от закона распределения случайной величины η и числа реализаций N при статистическом моделировании системы S . Если вероятность

расхождения теоретического и эмпирического распределений $P\{U_i \geq U\}$ велика в понятиях применяемого критерия согласия, то проверяемая гипотеза о виде распределения H_0 не опровергается. Выбор вида теоретического распределения $F(y)$ (или $f(y)$) проводится по графикам (гистограммам) $F_s(y)$ (или $f_s(y)$), выведенным на печать или на экран дисплея.

Рассмотрим особенности использования при обработке результатов моделирования системы S на ЭВМ ряда критериев согласия [7, 11, 18, 21, 25].

Критерий согласия Колмогорова. Основан на выборе в качестве меры расхождения U величины $D = \max [F_s(y) - F(y)]$.

Из теоремы Колмогорова следует, что $\delta = D\sqrt{N}$ при $N \rightarrow \infty$ имеет функцию распределения

$$F(z) = P\{\delta < z\} = \sum_{k=-\infty}^{\infty} (-1)^k e^{-2k^2 z^2}, \quad z > 0.$$

Если вычисленное на основе экспериментальных данных значение δ меньше, чем табличное значение при выбранном уровне значимости γ , то гипотезу H_0 принимают, в противном случае расхождение между $F_s(y)$ и $F(y)$ считается неслучайным, и гипотеза H_0 отвергается.

Критерий Колмогорова для обработки результатов моделирования целесообразно применять в тех случаях, когда известны все параметры теоретической функции распределения. Недостаток использования этого критерия связан с необходимостью фиксации в памяти ЭВМ для определения D всех статистических частот с целью их упорядочения в порядке возрастания.

Критерий согласия Пирсона. Основан на определении в качестве меры расхождения U величины

$$\chi^2 = \sum_{i=1}^d (m_i - Np_i) / (Np_i),$$

где m_i — количество значений случайной величины η , попавших в i -й подынтервал; p_i — вероятность попадания случайной величины η в i -й подынтервал, вычисленная из теоретического распределения; d — количество подынтервалов, на которые разбивается интервал измерения в машинном эксперименте.

При $N \rightarrow \infty$ закон распределения величины U , являющейся мерой расхождения, зависит только от числа подынтервалов и приближается к закону распределения χ^2 (хи-квадрат) с $(d-r-1)$ степенями свободы, где r — число параметров теоретического закона распределения.

Из теоремы Пирсона следует, что, какова бы ни была функция распределения $F(y)$ случайной величины η , при $N \rightarrow \infty$ распределение величины χ^2 имеет вид

$$F_k(z) = P\{\chi^2 < z\} = 1 / [2^{k/2} \Gamma(k/2)] \int_0^z e^{-t/2} t^{(k/2)-1} dt, \quad z > 0,$$

где $\Gamma(k/2)$ — гамма-функция; z — значение случайной величины χ^2 ; $k = d - r - 1$ — число степеней свободы. Функции распределения $F_k(z)$ табулированы.

По вычисленному значению $U = \chi^2$ и числу степеней свободы k с помощью таблиц находится вероятность $P\{\chi^2_i \geq \chi^2\}$. Если эта вероятность превышает некоторый уровень значимости γ , то считается, что гипотеза H_0 о виде распределения не опровергается результатами машинного эксперимента.

Критерий согласия Смирнова. При оценке адекватности машинной модели M_m реальной системе S возникает необходимость проверки гипотезы H_0 , заключающейся в том, что две выборки принадлежат той же генеральной совокупности. Если

выборки независимы и законы распределения совокупностей $F(u)$ и $F(z)$, из которых извлечены выборки, являются непрерывными функциями своих аргументов v и ξ , то для проверки гипотезы H_0 можно использовать критерий согласия Смирнова, применение которого сводится к следующему. По имеющимся результатам вычисляют эмпирические функции распределения $F_3(u)$ и $F_3(z)$ и определяют

$$D = \max |F_3(u) - F_3(z)|.$$

Затем при заданном уровне значимости γ находят допустимое отклонение

$$D_\gamma = \sqrt{\ln \gamma (1/N_1 + 1/N_2)/2},$$

где N_1 и N_2 — объемы сравниваемых выборок для $F_3(u)$ и $F_3(z)$, и проводят сравнение значений D и D_γ : если $D > D_\gamma$, то нулевую гипотезу H_0 о тождественности законов распределения $F(u)$ и $F(z)$ с доверительной вероятностью $\beta = 1 - \gamma$ отвергают.

Критерий согласия Стьюдента. Сравнение средних значений двух независимых выборок, взятых из нормальных совокупностей с неизвестными, но равными дисперсиями $D[v] = D[\xi]$, сводится к проверке нулевой гипотезы $H_0: \Delta = u - z = 0$ на основании критерия согласия Стьюдента (t -критерия). Проверка по этому критерию сводится к выполнению следующих действий. Вычисляют оценку

$$t = [(\bar{u} - \bar{z}) / \sqrt{(N_1 - 1)\bar{\sigma}_1^2 + (N_2 - 1)\bar{\sigma}_2^2}] / \sqrt{N_1 N_2 (N_1 + N_2 - 2) / (N_1 + N_2)},$$

где N_1 и N_2 — объемы выборок для оценки \bar{u} и \bar{z} соответственно; $\bar{\sigma}_1^2$ и $\bar{\sigma}_2^2$ — оценки дисперсий соответствующих выборок.

Затем определяют число степеней свободы $k = N_1 + N_2 - 2$, выбирают уровень значимости γ и по таблицам находят значение t_γ . Расчетное значение t сравнивается с табличным t_γ и если $|t| < t_\gamma$, то гипотеза H_0 не опровергается результатами машинного эксперимента.

Критерий согласия Фишера. Задача сравнения дисперсий сводится к проверке нулевой гипотезы H_0 , заключающейся в принадлежности двух выборок к одной и той же генеральной совокупности. Пусть необходимо сравнить две дисперсии $\bar{\sigma}_1^2$ и $\bar{\sigma}_2^2$, полученные при обработке результатов моделирования и имеющие k_1 и k_2 степеней свободы соответственно, причем $\bar{\sigma}_1^2 > \bar{\sigma}_2^2$. Для того чтобы опровергнуть нулевую гипотезу $H_0: \sigma_1^2 = \sigma_2^2$, необходимо при уровне значимости γ указать значимость расхождения между $\bar{\sigma}_1^2$ и $\bar{\sigma}_2^2$. При условии независимости выборок, взятых из нормальных совокупностей, в качестве критерия значимости используется распределение Фишера (F -критерий) $F = \sigma_1^2 / \sigma_2^2$, которое зависит только от числа степеней свободы $k_1 = N_1 - 1$, $k_2 = N_2 - 1$, где N_1 и N_2 — объемы выборок для оценки σ_1^2 и σ_2^2 соответственно.

Алгоритм применения критерия Фишера следующий: 1) вычисляется выборочное отношение $F = \bar{\sigma}_1^2 / \bar{\sigma}_2^2$; 2) определяется число степеней свободы $k_1 = N_1 - 1$ и $k_2 = N_2 - 1$; 3) при выбранном уровне значимости γ по таблицам F -распределения находятся значения границ критической области $F_1 = 1/[F_{1-\gamma/2}(k_1, k_2)]$; $F_2 = F_{1-\gamma/2}(k_1, k_2)$; 4) проверяется неравенство $F_1 \leq 1 \leq F_2$; если это неравенство выполняется, то с доверительной вероятностью β нулевая гипотеза $H_0: \sigma_1^2 = \sigma_2^2$ может быть принята.

Хотя рассмотренные оценки искомых характеристик процесса функционирования системы S , полученные в результате машинного эксперимента с моделью M_n , являются простейшими, но охватывают большинство случаев, встречающихся в практике обработки результатов моделирования системы для целей ее исследования и проектирования.

7.2. АНАЛИЗ И ИНТЕРПРЕТАЦИЯ РЕЗУЛЬТАТОВ МАШИННОГО МОДЕЛИРОВАНИЯ

Возможность фиксации при моделировании системы S на ЭВМ значений переменных (параметров) и их статистическая обработка для получения интересующих экспериментатора характеристик позволяют провести объективный анализ связей между этими величинами. Для решения этой задачи существуют различные методы, зависящие от целей исследования и вида получаемых при моделировании характеристик. Рассмотрим особенности использования методов корреляционного, регрессионного и дисперсионного анализа для результатов моделирования систем [7, 11, 18, 21, 25, 46].

Корреляционный анализ результатов моделирования. С помощью корреляционного анализа исследователь может установить, насколько тесна связь между двумя (или более) случайными величинами, наблюдаемыми и фиксируемыми при моделировании конкретной системы S . Корреляционный анализ результатов моделирования сводится к оценке разброса значений η относительно среднего значения \bar{y} , т. е. к оценке силы корреляционной связи. Существование этих связей и их тесноту можно для схемы корреляционного анализа $y = M[\eta/\xi = x]$ выразить при наличии линейной связи между исследуемыми величинами и нормальности их совместного распределения с помощью коэффициента корреляции

$$\begin{aligned} r_{\xi\eta} &= M[\xi - M[\xi]] M[\eta - M[\eta]] / \sqrt{D[\xi] D[\eta]} = \\ &= M[\xi - \mu_\xi] M[\eta - \mu_\eta] / (\sigma_\xi \sigma_\eta), \end{aligned}$$

т. е. второй смешанный центральный момент делится на произведение средних квадратичных отклонений, чтобы иметь безразмерную величину, инвариантную относительно единиц измерения рассматриваемых случайных переменных.

Пример 7.1. Пусть результаты моделирования получены при N реализациях, а коэффициент корреляции

$$\tilde{r}_{\xi\eta} = \frac{\sum_{k=1}^N (x_k - \bar{x})(y_k - \bar{y})}{\left[\sum_{k=1}^N (x_k - \bar{x})^2 \sum_{k=1}^N (y_k - \bar{y})^2 \right]^{1/2}} = \frac{\sum_{k=1}^N x_k y_k - N \bar{x} \bar{y}}{\left[\left(\sum_{k=1}^N x_k^2 - N \bar{x}^2 \right) \left(\sum_{k=1}^N y_k^2 - N \bar{y}^2 \right) \right]^{1/2}}$$

Очевидно, что данное соотношение требует минимальных затрат машинной памяти на обработку результатов моделирования. Получаемый при этом коэффициент корреляции $|r_{\xi\eta}| \leq 1$. При сделанных предположениях $r_{\xi\eta} = 0$ свидетельствует о взаимной независимости случайных переменных ξ и η , исследуемых при моделировании (рис. 7.1, а). При $|r_{\xi\eta}| = 1$ имеет место функциональная (т. е. нестохастическая) линейная зависимость вида $y = b_0 + b_1 x$, причем если $r_{\xi\eta} > 0$, то говорят о положительной корреляции, т. е. большие значения одной случайной величины соответствуют большим значениям другой (рис. 7.1, б). Случай $0 < r_{\xi\eta} < 1$ соответствует либо наличию линейной корреляции с рассеянием (рис. 7.1, в), либо наличию нелинейной корреляции результатов моделирования (рис. 7.1, г).

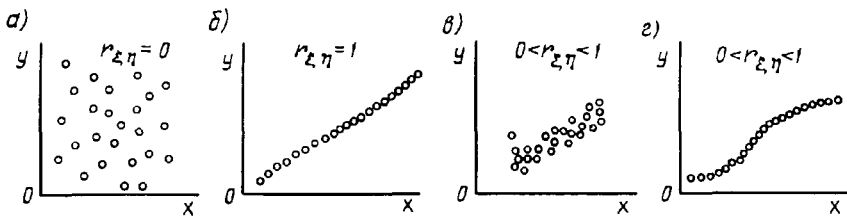


Рис. 7.1. Различные случаи корреляции переменных

Для того чтобы оценить точность полученной при обработке результатов моделирования системы S оценки $r_{\xi\eta}$, целесообразно ввести в рассмотрение коэффициент $w = \ln [(1 + r_{\xi\eta}) / (1 - r_{\xi\eta})] / 2$, причем w приближенно подчиняется гауссовскому распределению со средним значением и дисперсией:

$$\mu_w = \ln [(1 + r_{\xi\eta}) / (1 - r_{\xi\eta})] / 2, \quad \sigma_w^2 = 1 / (N - 3).$$

Из-за влияния числа реализаций при моделировании N на оценку коэффициента корреляции необходимо убедиться в том, что $0 \leq r_{\xi\eta} \leq 1$ действительно отражает наличие статистически значимой корреляционной зависимости между исследуемыми переменными модели M_u . Это можно сделать проверкой гипотезы $H_0: r_{\xi\eta} = 0$. Если гипотеза H_0 при анализе отвергается, то корреляционную зависимость признают статистически значимой. Очевидно, что выборочное распределение введенного в рассмотрение коэффициента w при $r_{\xi\eta} = 0$ является гауссовским с нулевым средним $\mu_w = 0$ и дисперсией $\sigma_w^2 = (N - 3)^{-1}$. Следовательно, область принятия гипотезы H_0 определяется неравенством

$$-z_{\alpha/2} \leq \sqrt{N - 3} \ln [(1 + r_{\xi\eta}) / (1 - r_{\xi\eta})] / 2 \leq z_{\alpha/2},$$

где $z_{\alpha/2}$ подчиняется нормированному гауссовскому распределению. Если $r_{\xi\eta}$ лежит вне приведенного интервала, то это означает наличие корреляционной зависимости между переменными модели на уровне значимости γ .

При анализе результатов моделирования системы S важно отметить то обстоятельство, что даже если удалось установить тесную зависимость между двумя переменными, то отсюда еще непосредственно не следует их причинно-следственная взаимообусловленность. Возможна ситуация, когда случайные ξ и η стохастически зависимы, хотя причинно они являются для системы S независимыми. При статистическом моделировании наличие такой зависимости может иметь место, например, из-за коррелированности последовательностей псевдослучайных чисел, используемых для имитации событий, положенных в основу вычисления значений x и y .

Таким образом, корреляционный анализ устанавливает связь между исследуемыми случайными переменными машинной модели

и оценивает тесноту этой связи. Однако в дополнение к этому желательно располагать моделью зависимости, полученной после обработки результатов моделирования.

Регрессионный анализ результатов моделирования. *Регрессионный анализ* дает возможность построить модель, наилучшим образом соответствующую набору данных, полученных в ходе машинного эксперимента с системой S . Под наилучшим соответствием понимается минимизированная функция ошибки, являющаяся разностью между прогнозируемой моделью и данными эксперимента. Такой функцией ошибки при регрессионном анализе служит сумма квадратов ошибок.

Пример 7.2. Рассмотрим особенности регрессионного анализа результатов моделирования при построении линейной регрессионной модели. На рис. 7.2, а показаны точки $x_i, y_i, i = \overline{1, N}$, полученные в машинном эксперименте с моделью M_M системы S .

Делаем предположение, что модель результатов машинного эксперимента графически может быть представлена в виде прямой линии

$$\hat{y} = \varphi(x) = b_0 + b_1 x,$$

где \hat{y} — величина, предсказываемая регрессионной моделью.

Требуется получить такие значения коэффициентов b_0 и b_1 , при которых сумма квадратов ошибок является минимальной. На рисунке ошибка $e_i, i = \overline{1, N}$, для каждой

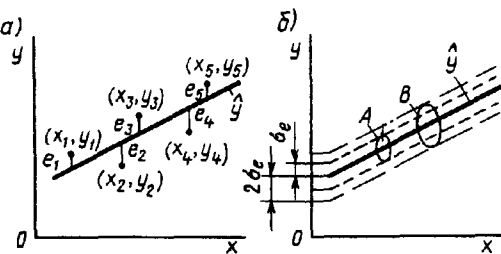


Рис. 7.2. Построение линейной регрессионной модели

экспериментальной точки определяется как расстояние по вертикали от этой точки до линии регрессии $y = \varphi(x)$.

Обозначим $\hat{y}_i = b_0 + b_1 x_i, i = \overline{1, N}$. Тогда выражение для ошибок будет иметь вид

$$e_i = \hat{y}_i - y_i = b_0 + b_1 x_i - y_i, \text{ а функция ошибки } F_0 = \sum_{i=1}^N (b_0 + b_1 x_i - y_i)^2.$$

Для получения b_0 и b_1 , при которых функция F_0 является минимальной, применяются обычные методы математического анализа. Условием минимума является $\partial F_0 / \partial b_0 = 0; \partial F_0 / \partial b_1 = 0$.

Дифференцируя F_0 , получаем

$$\begin{aligned} \partial F_0 / \partial b_0 &= \partial \sum_{i=1}^N (b_0 + b_1 x_i - y_i)^2 / \partial b_0 = 2 \left(N b_0 + b_1 \sum_{i=1}^N x_i - \sum_{i=1}^N y_i \right) = 0, \\ \partial F_0 / \partial b_1 &= \partial \sum_{i=1}^N (b_0 + b_1 x_i - y_i)^2 / \partial b_1 = 2 b_0 \sum_{i=1}^N x_i + 2 b_1 \sum_{i=1}^N x_i^2 - 2 \sum_{i=1}^N x_i y_i = 0 \end{aligned}$$

Решая систему этих двух линейных алгебраических уравнений, можно получить значения b_0 и b_1 . В матричном представлении эти уравнения имеют вид

$$\begin{pmatrix} N & \sum_{i=1}^N x_i \\ \sum_{i=1}^N x_i & \sum_{i=1}^N x_i^2 \end{pmatrix} \cdot \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^N y_i \\ \sum_{i=1}^N x_i y_i \end{pmatrix}.$$

Решая это уравнение, получаем

$$b_0 = \left(\sum_{i=1}^N y_i \sum_{i=1}^N x_i^2 - \sum_{i=1}^N x_i \sum_{i=1}^N x_i y_i \right) / \left[N \sum_{i=1}^N x_i^2 - \left(\sum_{i=1}^N x_i \right)^2 \right],$$

$$b_1 = \left(N \sum_{i=1}^N x_i y_i - \sum_{i=1}^N x_i \sum_{i=1}^N y_i \right) / \left[N \sum_{i=1}^N x_i^2 - \left(\sum_{i=1}^N x_i \right)^2 \right],$$

где N — число реализаций при моделировании системы.

Соотношения для вычисления b_0 и b_1 требуют минимального объема памяти ЭВМ для обработки результатов моделирования. Обычно мерой ошибки регрессионной модели служит среднее квадратичное отклонение

$$\sigma_e = \left[\left(\sum_{i=1}^N e_i^2 \right) / (N-2) \right]^{1/2} = \left\{ \left[\sum_{i=1}^N (b_0 - b_1 x_i - y_i)^2 \right] / (N-2) \right\}^{1/2}.$$

Для нормально распределенных процессов приблизительно 67% точек находится в пределах одного отклонения σ_e от линии регрессии и 95% — в пределах $2\sigma_e$ (трубки A и B соответственно на рис. 7.2, б). Для проверки точности оценок b_0 и b_1 регрессионной модели могут быть использованы, например, критерии Фишера (F -распределение) и Стьюдента (t -распределение). Аналогично могут быть оценены коэффициенты уравнения регрессии и для случая нелинейной аппроксимации.

Дисперсионный анализ результатов моделирования. При обработке и анализе результатов моделирования часто возникает задача сравнения средних выборок. Если в результате такой проверки окажется, что математическое ожидание совокупностей случайных переменных $\{y^{(1)}\}$, $\{y^{(2)}\}$, ..., $\{y^{(n)}\}$ отличается незначительно, то статистический материал, полученный в результате моделирования, можно считать однородным (в случае равенства двух первых моментов). Это дает возможность объединить все совокупности в одну и позволяет существенно увеличить информацию о свойствах исследуемой модели M_m , а следовательно, и системы S . Попарное использование для этих целей критериев Смирнова и Стьюдента для проверки нулевой гипотезы затруднено в связи с наличием большого числа выборок при моделировании системы. Поэтому для этой цели используется *дисперсионный анализ*.

Пример 7.3. Рассмотрим решение задачи дисперсионного анализа при обработке результатов моделирования системы в следующей постановке. Пусть генеральные совокупности случайной величины $\{y^{(1)}\}$, $\{y^{(2)}\}$, ..., $\{y^{(n)}\}$ имеют нормальное распределение и одинаковую дисперсию. Необходимо по выборочным средним значениям при некотором уровне значимости γ проверить нулевую гипотезу H_0 о равенстве математических ожиданий. Выявим влияние на результаты моделирования только одного фактора, т. е. рассмотрим однофакторный дисперсионный анализ.

Допустим, изучаемый фактор x привел к выборке значений неслучайной

величины Y следующего вида: y_1, y_2, \dots, y_k , где k — количество уровней фактора x . Влияние фактора будем оценивать неслучайной величиной D_x , называемой факторной дисперсией:

$$D_x = \sigma_x^2 = \sum_{i=1}^k (y_i - \bar{y})^2 / k,$$

где \bar{y} — среднее арифметическое значение величины Y .

Если генеральная дисперсия $D[y]$ известна, то для оценки случайности разброса наблюдений необходимо сравнить $D[y]$ с выборочной дисперсией S_y^2 , используя критерий Фишера (F -распределение). Если эмпирическое значение F_y попадает в критическую область, то влияние фактора x считается значимым, а разброс значений x — неслучайным. Если генеральная дисперсия $D[x]$ до проведения машинного эксперимента с моделью M_M неизвестна, то необходимо при моделировании найти ее оценку.

Пусть серия наблюдений на уровне y_i имеет вид $y_{i1}, y_{i2}, \dots, y_{in}$, где n — число повторных наблюдений на i -м уровне. Тогда на i -м уровне среднее значение наблюдений

$$\bar{y}_i = \frac{1}{n} \sum_{j=1}^n y_{ij}$$

а среднее значение наблюдений по всем уровням

$$\bar{y} = \frac{1}{kn} \sum_{j=1}^k \sum_{i=1}^n y_{ij} = \frac{1}{k} \sum_{i=1}^k \bar{y}_i.$$

Общая выборочная дисперсия всех наблюдений

$$S_y^2 = \frac{1}{kn-1} \left[\sum_{i=1}^k \sum_{j=1}^n y_{ij}^2 - \frac{1}{kn} \left(\sum_{i=1}^k \sum_{j=1}^n y_{ij} \right)^2 \right].$$

При этом разброс значений y определяется суммарным влиянием случайных причин и фактора x . Задача дисперсионного анализа состоит в том, чтобы разложить общую дисперсию $D[y]$ на составляющие, связанные со случайными и неслучайными причинами.

Оценка генеральной дисперсии, связанной со случайными факторами,

$$\bar{D}_0[y] = \frac{1}{k(n-1)} \left[\sum_{i=1}^k \sum_{j=1}^n y_{ij}^2 - \frac{1}{n} \sum_{i=1}^k \left(\sum_{j=1}^n y_{ij} \right)^2 \right],$$

а оценка факторной дисперсии

$$\bar{D}_x = \bar{D}[y] - \bar{D}_0[y].$$

Учитывая, что факторная дисперсия наиболее заметна при анализе средних значений на i -м уровне фактора, а остаточная дисперсия (дисперсия случайности) для средних значений в n раз меньше, чем для отдельных измерений, найдем более точную оценку выборочной дисперсии:

$$\bar{D}_x + \frac{1}{n} \bar{D}_0[y] = \frac{f}{k-f} \sum_{i=1}^k (y_i - \bar{y})^2.$$

Умножив обе части этого выражения на n , получим в правой части выборочную дисперсию S_y^2 , имеющую $(k-1)$ -ю степень свободы. Влияние фактора x будет значимым, если при заданном γ выполняется неравенство $\bar{S}_y^2 / \bar{D}_0[y] > F_{1-\gamma}$. В противном случае влиянием фактора x на результаты моделирования можно пренебречь и счи-

тать нулевую гипотезу H_0 о равенстве средних значений на различных уровнях справедливой.

Таким образом, дисперсионный анализ позволяет вместо проверки нулевой гипотезы о равенстве средних значений выборок проводить при обработке результатов моделирования проверку нулевой гипотезы о тождественности выборочной и генеральной дисперсий.

Возможны и другие подходы к анализу и интерпретации результатов моделирования, но при этом необходимо помнить, что их эффективность существенно зависит от вида и свойств конкретной моделируемой системы S .

7.3. ОБРАБОТКА РЕЗУЛЬТАТОВ МАШИННОГО ЭКСПЕРИМЕНТА ПРИ СИНТЕЗЕ СИСТЕМ

При синтезе системы S на базе машинной модели M_m задача поиска оптимального варианта системы при выбранном критерии оценки эффективности и заданных ограничениях решается путем анализа характеристик процесса функционирования различных вариантов системы, их сравнительной оценки и выбора наилучшего варианта. Независимо от того, как организуется выбор наилучшего варианта системы — простым перебором всех проанализированных при машинных экспериментах результатов или с помощью специальных процедур поиска оптимального варианта, например методов математического программирования, — элементарной операцией является сравнение статистически усредненных критериев оценки эффективности вариантов систем [9, 29, 33, 53].

Особенности машинного синтеза. Учитывая то обстоятельство, что конкурирующие варианты системы S отличаются друг от друга структурой, алгоритмами поведения, параметрами, число таких вариантов достаточно велико. Поэтому при синтезе оптимального варианта системы S_{opt} особенно важно минимизировать затраты ресурсов на получение в результате моделирования характеристик каждого варианта системы. Исходя из этих особенностей, при синтезе системы S обработку и анализ результатов моделирования каждого варианта системы S следует рассматривать не автономно, а в их тесной взаимосвязи. Очевидно, что задача синтеза оптимального варианта моделируемой системы S_{opt} должна быть уже поставлена при планировании машинного эксперимента с моделью M_m .

В предыдущей главе было показано, что искусственная организация статистической зависимости между выходными характеристиками сравниваемых вариантов S_1 и S_2 системы дает выигрыш в точности оценки средних значений, вероятностей и дисперсий при положительно коррелированных критериях q_1 и q_2 . Корреляция между критериями q_1 и q_2 возникает в силу того, что случайные векторы

$$\vec{v} = (v_1, \dots, v_k, v_{k+1}^{(1)}, \dots, v_n^{(1)}), \vec{v}_2 = (v_1, \dots, v_k, v_{k+1}^{(2)}, \dots, v_m^{(2)}),$$

описывающие воздействие внешней среды E на варианты S_1 и S_2 системы, имеют общие составляющие $\vec{v} = (v_1, \dots, v_k)$, в то время как составляющие $(v_{k+1}^{(1)}, \dots, v_n^{(1)})$ и $(v_{k+1}^{(2)}, \dots, v_m^{(2)})$ статистически независимы.

Если через $\vec{v} = (v_1, \dots, v_k)$ обозначить фиксированные значения составляющих $\vec{v} = (v_1, \dots, v_k)$, то условные средние значения q_1 и q_2 будут такими:

$$\mu_1(\vec{v}) = M[q_1/\vec{v}] = M[q_2/\vec{v}],$$

т. е. являются функциями переменных $\vec{v} = (v_1, \dots, v_k)$.

Рассмотрим особенности обработки результатов моделирования, когда сравниваемые в ходе проведения имитационных экспериментов полные средние значения критериев q_1 и q_2 примут вид

$$\mu_1 = M[q_1] = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \mu_1(\vec{v}) f(\vec{v}) d\vec{v};$$

$$\mu_2 = M[q_2] = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \mu_2(\vec{v}) f(\vec{v}) d\vec{v},$$

где $d\vec{v} = dv_1, \dots, dv_k$; $f(\vec{v}) = f(v_1, \dots, v_k)$ — совместная плотность вероятностей составляющих v_1, \dots, v_k .

Ковариация

$$B_{12} = B[q_1, q_2] = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \mu_1(\vec{v}) \mu_2(\vec{v}) f(\vec{v}) d\vec{v} - \mu_1 \mu_2.$$

Достаточным условием неотрицательности ковариации, дающим выигрыш в оценке разности средних, является одинаковая упорядоченность условных средних $\mu_1(\vec{v})$, $\mu_2(\vec{v})$ относительно векторного аргумента $\vec{v} = (v_1, \dots, v_k)$, т. е. выполнение неравенства

$$[\mu_1(\vec{z}) - \mu_1(\vec{u})][\mu_2(\vec{z}) - \mu_2(\vec{u})] \geq 0 \quad (7.1)$$

для любых значений векторных аргументов

$$\vec{z} = (z_1, \dots, z_k), \vec{u} = (u_1, \dots, u_k).$$

Действительно, учитывая, что $\int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} f(\vec{v}) d\vec{v} = 1$, находим

$$B_{12} = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \mu_1(\vec{v}) \mu_2(\vec{v}) f(\vec{v}) d\vec{v} - \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \mu_1(\vec{v}) f(\vec{v}) d\vec{v} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \mu_2(\vec{v}) f(\vec{v}) d\vec{v} =$$

$$\begin{aligned} \times \mu_2(\vec{v}) f(\vec{v}) d\vec{v} = 0,5 \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} f(\vec{z}) f(\vec{u}) [\mu_1(\vec{u}) \mu_2(\vec{u}) - \mu_1(\vec{z}) \mu_2(\vec{u}) + \\ + \mu_1(\vec{z}) \mu_2(\vec{z}) - \mu_1(\vec{u}) \mu_2(\vec{z})] d\vec{z} d\vec{u} = 0,5 \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} f(\vec{z}) f(\vec{u}) \times \\ \times [\mu_1(\vec{z}) - \mu_1(\vec{u})] [\mu_2(\vec{z}) - \mu_2(\vec{u})] d\vec{z} d\vec{u}. \end{aligned}$$

Так как $f(\vec{v}) \geq 0$ для всех \vec{v} , то при выполнении (7.1) имеем $B_{12} \geq 0$, что и требовалось доказать.

Когда в качестве результатов моделирования выступают вероятности событий A_1, A_2 для вариантов S_1 и S_2 системы, то условные значения

$$\mu_1(\vec{v}) = P(A_1/\vec{v}); \mu_2(\vec{v}) = P(A_2/\vec{v}),$$

где $P(A_i/\vec{v})$ — условная вероятность, $i=1, 2$.

Тогда достаточное условие неотрицательности ковариации запишется в виде

$$P(A_1/\vec{z}) - P(A_2/\vec{u}) [P(A_2/\vec{z}) - P(A_2/\vec{u})] \geq 0, \quad (7.2)$$

что соответствует одинаковой упорядоченности условных вероятностей $P(A_1/\vec{v})$ и $P(A_2/\vec{v})$ относительно векторного аргумента \vec{v} .

Одинаково упорядоченными являются монотонно возрастающие или монотонно убывающие функции $\mu_1(v)$ и $\mu_2(v)$ скалярного аргумента v , а также одинаковые функции $\mu_1(v) = \mu_2(v)$ независимо от их монотонности. Пример одинаково упорядоченных возрастающих (а) и убывающих (б) функций $\mu(v)$ показан на рис. 7.3.

Если положительные функции $\mu_j(\vec{v}), j = \overline{1, n}$, одинаково упорядочены, то произведение любой комбинации этих функций $\mu_k(\vec{v}) \mu_s(\vec{v}) \dots \mu_m(\vec{v})$ одинаково упорядочено с произведением любой комбинации $\mu_i(\vec{v}) \mu_q(\vec{v}) \dots \mu_p(\vec{v})$. Это же можно сказать и об условных вероятностях $P(A_j/\vec{v}), j = \overline{1, n}$.

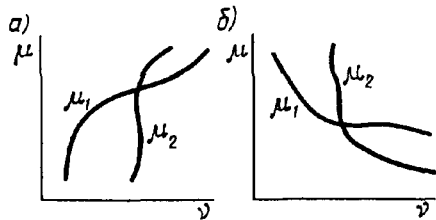


Рис. 7.3. Пример одинаково упорядоченных функций

Пример 7.4. Пусть методом статистического моделирования на ЭВМ необходимо сравнить результаты моделирования двух вариантов S_1 и S_2 системы, составленных из одинаковых блоков $B_1 - B_4$ (структура системы показана на рис. 7.4) и сравниваемых по критерию надежности с учетом случайных изменений внешней температуры. События A_1 и A_2 соответствуют безотказной работе вариантов S_1 и S_2 системы в течение заданного времени T .

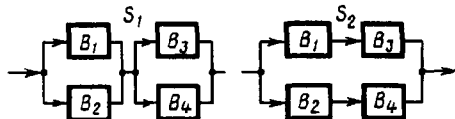


Рис. 7.4. Структуры сравниваемых вариантов систем S_1 и S_2

Вероятность безотказной работы B_i при заданной температуре $v = v$ можно определить как

$$P(B_i/v) = e^{-\lambda_i/vT}, \quad i = \overline{1, 4},$$

где $\lambda_i(v)$ — интенсивности отказов, являющиеся возрастающими функциями температуры.

Таким образом, функции $P(B_i/v)$ являются одинаково упорядоченными убывающими функциями. Можно показать, что функции $P(A_1/v) = \{1 - [1 - P(B_1/v)][1 - P(B_2/v)]\} \{1 - [1 - P(B_3/v)][1 - P(B_4/v)]\}$, $P(A_2/v) = 1 - [1 - P(B_1/v) \times P(B_3/v)][1 - P(B_2/v) \times P(B_4/v)]$ также одинаково упорядочены и убывают с ростом температуры v . Поэтому, используя при машинном эксперименте с вариантами S_1 и S_2 системы одни и те же реализации v случайной температуры v , получим в результате моделирования большую точность сравнения вероятностей $P(A_1)$ и $P(A_2)$, чем при раздельном моделировании S_1 и S_2 системы с использованием независимых реализаций v .

Рассмотренный пример можно обобщить и на случай векторного аргумента, например для набора таких переменных, как температура, давление, ускорение и т. п.

Когда независимые компоненты в воздействиях внешней среды E отсутствуют, т. е. $\vec{v}_1 = \vec{v}_2 = \vec{v}$, условные средние $\mu_1(\vec{v}) = M[q_1/\vec{v}]$, $\mu_2(\vec{v}) = M[q_2/\vec{v}]$ преобразуются в детерминированные зависимости критериев от случайных воздействий $q_1 = f_1(\vec{v})$, $q_2 = f_2(\vec{v})$.

При этом условия одинаковой упорядоченности становятся еще более жесткими. Так, например, условия (7.2) выполняются лишь тогда, когда для всех значений исключено одно из состояний: $A_1\overline{A}_2$ или \overline{A}_1A_2 . Другими словами, положительная корреляция B_{12} и связанные с ней преимущества гарантируются лишь тогда, когда вариант системы S_1 равномерно лучше (хуже) варианта S_2 . В принятых в § 6.3 обозначениях это соответствует $p_C = 0$ или $p_D = 0$.

Состояния $C = A_1\overline{A}_2$ или $D = \overline{A}_1A_2$ вариантов систем S_1 и S_2 возможны лишь при наличии двух неисправных блоков B_i , $i = \overline{1, 4}$, состояние $A = A_1A_2$ возможно при отсутствии неисправностей или при одной неисправности, а состояние $B = \overline{A}_1\overline{A}_2$ — при трех или четырех неисправностях. Обозначив через $B_i\overline{B}_j$ ситуацию с неисправностями блоков B_i и B_j , находим соответствие между состояниями

$$\begin{aligned} \overline{B}_1\overline{B}_2 \rightarrow B, \quad \overline{B}_1\overline{B}_3 \rightarrow A, \quad \overline{B}_1\overline{B}_4 \rightarrow C, \\ \overline{B}_2\overline{B}_3 \rightarrow C, \quad \overline{B}_2\overline{B}_4 \rightarrow A, \quad \overline{B}_3\overline{B}_4 \rightarrow B \end{aligned}$$

и убеждаемся в отсутствии состояния D .

Следует помнить, что условия одинаковой упорядоченности (7.1) и (7.2) являются достаточными, но не необходимыми и достаточными условиями неотрицательности корреляции. Поэтому, обнаружив в конкретной схеме проведения имитационного эксперимента нарушение этих условий при некоторых реализациях входных воздействий \vec{v} , следует более детально рассмотреть процедуру сравнения средних значений или вероятностей. Например, при сравнении вероятностей, задаваясь значениями $\Delta p = p_1 - p_2$, p_A и p_D , необходимо рассчитать значения $p_2 = p_A + p_D$, $p_1 = p_2 + \Delta p$, $p_C = p_D + \Delta p$ и вычислить коэффициенты корреляции и «выигрыша» соответственно:

$$\begin{aligned} R_{12} &= (p_A - p_1 p_2) / \sqrt{p_1(1-p_1)p_2(1-p_2)}; \\ \gamma &= N_n/N_s = [p_1(1-p_1) + p_2(1-p_2)] / [p_C + p_D - (p_C - p_D)^2], \end{aligned}$$

где N_n и N_1 — объемы выборки, необходимые для получения заданной точности оценки Δp при использовании независимых и зависимых реализаций.

Таким образом, использование зависимых испытаний дает возможность значительно сократить затраты машинного времени на моделирование. Рассмотренная методика сравнения характеристик вариантов при синтезе системы с учетом их корреляции является формальной. Однако основа для получения с помощью этой методики практических преимуществ — неформальная операция выбора такой схемы имитации, при которой искусственно создавалась бы требуемая корреляция.

Оценка результатов моделирования системы. Рассмотрим возможность оценки при обработке результатов моделирования абсолютных значений характеристик процесса функционирования системы S . Пусть исследование одного из вариантов системы, например S_2 , выполнено аналитическим методом и определено среднее значение μ_2 критерия q_2 . Тогда оценка $\mu'_1 = \mu_2 - \bar{d}$ среднего значения μ_1 имеет дисперсию

$$D[\tilde{\mu}'_1] = D[\bar{d}] = (D[\tilde{\mu}_1] + D[\tilde{\mu}_2])/\gamma_\mu = (1 + \alpha)D[\tilde{\mu}_1]/\gamma_\mu,$$

где γ_μ — коэффициент выигрыша, получаемого при оценке разности средних значений $\bar{d} = \mu_2 - \mu_1$ за счет зависимости испытаний; $\alpha = D[\tilde{\mu}_2]/D[\tilde{\mu}_1]$. Оценка $\tilde{\mu}'_1$ точнее $\tilde{\mu}_1$, если $(1 + \alpha)/\gamma_\mu < 1$.

Однако затраты машинного времени для получения оценки $\tilde{\mu}'_1$, которые обозначим как t_{12} , превышают при заданном N затраты машинного времени t_1 , необходимого для автономной оценки μ_1 . Поэтому при заданной точности оценки среднего μ_1 оценка $\tilde{\mu}'_1$ дает выигрыш по затратам машинного времени на имитацию только в том случае, если $(1 + \alpha)t_{12}/(\gamma_\mu t_1) < 1$.

Для нормально распределенных критериев q_1 и q_2 оценка дисперсии $D'_1 = D_2 + \Delta\tilde{D}$. Выигрыш в затратах машинного времени на имитационное моделирование по сравнению с автономной оценкой \tilde{D}_1 будет лишь при условии $(1 + \alpha)/t_{12}/\gamma_D t_1 < 1$, где γ_D — коэффициент выигрыша, получаемого при оценке разности дисперсий $\Delta\tilde{D}$ за счет зависимых испытаний.

Рассмотренные методы сравнения вариантов S_1 и S_2 моделируемой системы можно использовать в алгоритмах оптимизации на этапе проектирования системы S , т. е. при ее синтезе, по результатам имитационного эксперимента с ее машинной моделью M_m .

При синтезе системы S на основе проведения машинных экспериментов с моделью M_m возникает задача анализа чувствительности модели к вариациям ее параметров. Под анализом чувствительности машинной модели M_m понимают проверку устойчивости результатов моделирования, т. е. характеристик процесса функционирования системы S , полученных при проведении

имитационного эксперимента, по отношению к возможным отклонениям параметров машинной модели $\Delta \vec{h} = (\Delta h_1, \dots, \Delta h_n)$ от истинных их значений $\vec{h} = (h_1, \dots, h_n)$ [29, 33, 53].

Анализ чувствительности позволяет сравнивать методические погрешности, полученные при построении машинной модели M_m , с неточностями задания исходных данных, что особенно важно при практической реализации для целей синтеза системы S .

Малым отклонениям $\Delta \vec{h}$ будут соответствовать изменения характеристик $q(\vec{h})$, которые в практических расчетах можно оценить величиной $\Delta q = \vec{q}'(\vec{h}) \Delta \vec{h} + r_0$, где $\vec{q}'(\vec{h}) = (\partial q(\vec{h})/\partial h_1, \dots, \partial q(\vec{h})/\partial h_n)$; r_0 — остаточный член второго порядка малости относительно вариации, который используется для проверки точности решения.

Частная производная $\vec{q}'(\vec{h})$ определяется в точках, соответствующих номинальным значениям параметров $\vec{h}_{\text{ном}}$. Если $\vec{h}_{\text{ном}} = \vec{h}^*$, где \vec{h}^* — оптимальные параметры системы по показателю $\vec{q}(\vec{h})$, то $\vec{q}'(\vec{h}_{\text{ном}}) = 0$ и необходимо проводить оценку с использованием второй производной $\vec{q}''(\vec{h}_{\text{ном}})$. Таким образом, частные производные $\vec{q}'(\vec{h})$, $\vec{q}''(\vec{h})$ количественно характеризуют чувствительность машинной модели M_m к изменениям ее параметров.

Большие отклонения характеристик $q(\vec{h})$ при малых вариациях $\Delta \vec{h}$ свидетельствуют о неустойчивости модели M_m по отношению к этим вариациям. Для получения оценок $\vec{q}(\vec{h})$ показателя $q(\vec{h})$ удобно рассматривать зависимые реализации внешних воздействий при различных \vec{h} и проводить соответствующую обработку результатов машинного эксперимента с моделью M_m .

Чувствительность можно оценить и на более простой модели, чем модель для определения характеристик процесса функционирования системы S . Кроме того, универсальные оценки производных $\vec{q}'(\vec{h})$ и $\vec{q}''(\vec{h})$, выполняемые при моделировании по зависимым испытаниям, в ряде частных случаев можно заменить более удобными непосредственными вычислениями.

Таким образом, результаты машинного эксперимента с моделью M_m обрабатываются с учетом целей моделирования системы S , которые находятся в тесной связи с вопросами, решаемыми при планировании экспериментов. При синтезе системы S на базе ее машинной модели M_m необходимо принять меры по организации зависимых испытаний анализируемых вариантов системы и оценке чувствительности модели к вариации ее параметров, что позволит упростить работу с моделью на каждом шаге оптимизации.

Контрольные вопросы

7.1. Каковы особенности имитационного эксперимента на ЭВМ с точки зрения обработки результатов?

7.2. В чем сущность методов фиксации и обработки результатов при статистическом моделировании систем на ЭВМ?

7.3. Какие методы математической статистики используются для анализа результатов имитационного моделирования систем?

7.4. Какое место занимают имитационные модели при машинном синтезе систем?

7.5. Какова цель организации зависимых испытаний модели системы на ЭВМ?

ГЛАВА 8

МОДЕЛИРОВАНИЕ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ ТИПОВЫХ МАТЕМАТИЧЕСКИХ СХЕМ

Объекты информационных систем характеризуются сложностью структуры, алгоритмов поведения, многопараметричностью, что, естественно, приводит к сложности их машинных моделей; это требует при их разработке построения иерархических модульных конструкций, а также использования формального описания внутрисистемных процессов. Типовые математические схемы являются связующим звеном в цепочке «концептуальная модель — машинная модель», позволяя эффективно решать при моделировании проблемы взаимодействия заказчика (постановщика задачи) и исполнителя (разработчика модели). Наиболее характерными для информационных систем являются объекты дискретного типа (дискретные производственные процессы, вычислительные комплексы, каналы передачи данных, информационные сети и т. д.), что предопределяет необходимость детального ознакомления с машинным моделированием на базе дискретных, вероятностных, а также универсальных типовых математических схем.

8.1. ИЕРАРХИЧЕСКИЕ МОДЕЛИ ПРОЦЕССОВ ФУНКЦИОНИРОВАНИЯ СИСТЕМ

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

Блочная конструкция модели. Рассмотрим машинную модель M_m системы S как совокупность блоков $\{m_i\}$, $i = \overline{1, n}$. Каждый блок модели можно охарактеризовать конечным набором возможных состояний $\{z_0\}$, в которых он может находиться. Пусть в течение рассматриваемого интервала времени $(0, T)$, т. е. времени прогона модели, блок изменяет состояния в моменты времени $t_i^{(j)} \leq T$, где j — номер момента времени. Вообще моменты времени смены состояния блока m , можно условно разделить на три группы: 1) случайные моменты, связанные с внутренними свойствами части системы S , соответствующей данному блоку; 2) случайные моменты, связанные с изменением состояний других блоков (включая блоки, имитирующие воздействия внешней среды E); 3) детерминированные моменты, связанные с заданным расписанием функционирования блоков модели [29, 36, 37, 53].

Моментами смены состояний модели M_m в целом $t^{(k)} \leq T$ будем считать все моменты изменения состояний блоков $\{m_i\}$, т. е. $\{t_i^{(j)}\} \in \{t^k\}$, $i = \overline{1, n}$. Пример для модели с тремя блоками m_1 , m_2 и m_3 показан на рис. 8.1.

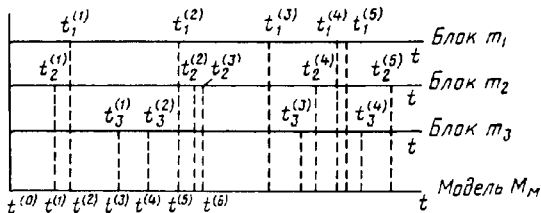


Рис. 8.1. Смена состояний модели для случая трех блоков

При этом моменты $t_i^{(j)}$ и t^k являются моментами системного времени, т. е. времени, в котором функционирует моделируемая система S , но не моментами машинного времени. Мгновенные изменения состояний модели во время дискретного события (особого состояния) возможны только при моделировании в системном времени.

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

При машинной реализации модели M_m ее блоки, имеющие аналогичные функции, могут быть представлены в виде отдельных

программных модулей. Работа каждого такого модуля имитирует работу всех однотипных блоков. В общем случае при числе блоков модели n можно получить набор машинных модулей $l \leq n$. Таким образом, каждому блоку или элементу модели будет соответствовать некоторый модуль или «стандартная подпрограмма», число которых не будет превосходить числа блоков модели.

Моделирующий алгоритм. Типовая укрупненная схема моделирующего алгоритма, построенного по блочному принципу, для систем с дискретными событиями приведена на рис. 8.2.

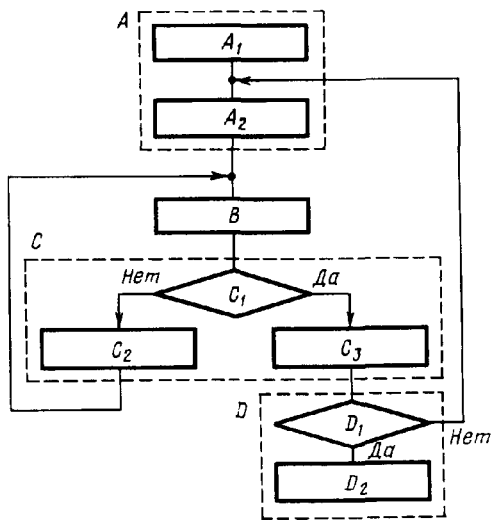


Рис. 8.2. Типовая укрупненная схема моделирующего алгоритма

Эта схема содержит следующие укрупненные модули: A — модуль задания начальных значений состояний, содержащий два подмодуля (A_1 — для задания начальных состояний моделируемого варианта и A_2 — для задания начальных состояний для одного прогона модели); B — модуль определения очередного момента смены состояния, осуществляющий просмотр массива состояний и выбирающий блок модели m_i , $i = \overline{1, n}$, с минимальным временем смены состояния $\min t_i^0$; C — модуль логического переключения, содержащий три подмодуля (C_1 — для логического перехода по номеру блока модели i или по времени T , т. е. для решения вопроса о завершении прогона; C_2 — для фиксации информации о состояниях, меняющихся при просмотре блока, а также для определения момента следующей смены состояния блока m_i и номера следующего особого состояния s_0 ; C_3 — для завершения прогона в случае, когда $t_i^0 \geq T$, фиксации и предварительной обработки результатов моделирования); D — модуль управления и обработки, содержащий два подмодуля (D_1 — для проверки окончания исследования варианта модели M_n по заданному числу прогонов или по точности результатов моделирования; D_2 — для окончательной обработки информации, полученной на модели M_n , и выдачи результатов моделирования).

Данная укрупненная схема моделирующего алгоритма соответствует статике моделирования. При необходимости организации моделирования последовательностей вариантов модели M_n и проведении оптимизации моделируемой системы S , например, на этапе ее проектирования, т. е. для решения вопросов, относящихся к динамике моделирования, следует добавить внешний цикл для варьирования структуры, алгоритмов и параметров модели M_n .

Пример 8.1. Рассмотрим модульный принцип реализации модели S , формализованной в виде Q -схемы. Пусть имеется L^Φ -фазная многоканальная Q -схема без потерь с $L^И$ -входными потоками заявок. В каждой фазе имеется L_j^K , $j = \overline{1, L^\Phi}$, каналов обслуживания. Определить распределения времени ожидания заявок в каждой фазе и времени простоя каждого обслуживающего канала.

В качестве блоков модели M_n будем рассматривать: $m^И$ — блоки источников заявок, имитирующие $L^И$ входных потоков; m^K — блоки каналов обслуживания, имитирующие функционирование каналов; $m^В$ — блок взаимодействия, отражающий взаимосвязь всех блоков машинной модели M_n . При этом в массиве состояний будем фиксировать моменты поступления заявок, освобождения каналов и окончания моделирования, т. е. количество элементов этого массива будет равно

$$L^И + \sum_{j=1}^{L^\Phi} L_j^K + 1.$$

Схема моделирующего алгоритма для данного примера приведена на рис. 8.3. Как видно из схемы, в подмодуле C_2 предусмотрены три вида процедур C_2 , C_2' и C_2'' . Первая процедура C_2' работает при поступлении заявки из любого входного потока, вторая процедура C_2'' работает в момент освобождения канала любой фазы об-

служивания, кроме последней, третья процедура C''_1 работает при освобождении канала последней фазы, т. е. при окончании обслуживания заявки Q -схемой.

Рассмотрим более подробно операторы процедур C'_2 , C''_2 и C''_1 . Оператор C'_{21} определяет принадлежность заявки к одному из L^I входных потоков, генерируемых модулем B . Оператор C'_{22} проверяет, есть ли на первой фазе очередь свободных каналов обслуживания. Если очередь есть, то управление передается оператору C'_{23} , в противном случае — оператору C'_{24} . Оператор C'_{23} фиксирует момент поступления заявки в массиве очереди заявок первой фазы. Оператор C'_{24} выбирает номер канала из массива очереди канала первой фазы, уменьшая ее длину на единицу, вычисляет и фиксирует длительность простоя канала, определяет длительность обслуживания и засылает новый момент освобождения канала в массив состояний. Оператор C'_{25} определяет новый момент поступления заявки и засылает его в соответствующую ячейку массива состояний.

Оператор C''_{21} служит для определения j -й фазы и k -го канала,

$j=2, L^{\Phi}-1, k=1, L^F$. Оператор C''_{22} проверяет наличие очереди заявок на выбранной j -й фазе. При отсутствии очереди управление передается оператору C''_{23} , а при ее наличии — оператору C''_{24} . Оператор C''_{23} засылает момент освобождения канала в массив очереди каналов j -й фазы, уменьшает длину очереди на единицу и фиксирует время ожидания выбранной заявкой начала ее обслуживания. Далее определяется длительность обслуживания этой заявки освободившимся каналом, вычисляется и засылается в массив состояний новый момент освобождения канала. Операторы C''_{25} , C''_{26} и C''_{27} выполняют те же действия с заявкой, обслуживаемой на j -й фазе, что и операторы C'_{22} , C'_{23} и C'_{24} с заявкой, которая поступила в первую фазу Q -схемы.

Оператор C''_{24} настраивает операторы этой процедуры C''_{22} , C''_{23} и C''_{24} на выбранный канал обслуживания последней, L^{Φ} -й, фазы. Работа операторов C''_{22} , C''_{23} и C''_{24} аналогична работе операторов C'_{22} , C'_{23} и C'_{24} .

Назначение остальных подмодулей алгоритма не отличается от рассмотренного ранее для моделирующего алгоритма, приведенного на рис. 8.2.

Построение моделирующего алгоритма по блочному принципу позволяет за счет организации программных модулей уменьшить затраты времени на моделирование системы S , так как машинное время в этом случае не тратится на просмотр повторяющихся ситуаций. Кроме того, данная схема моделирующего алгоритма получается проще, чем в случае, когда модули не выделяются.

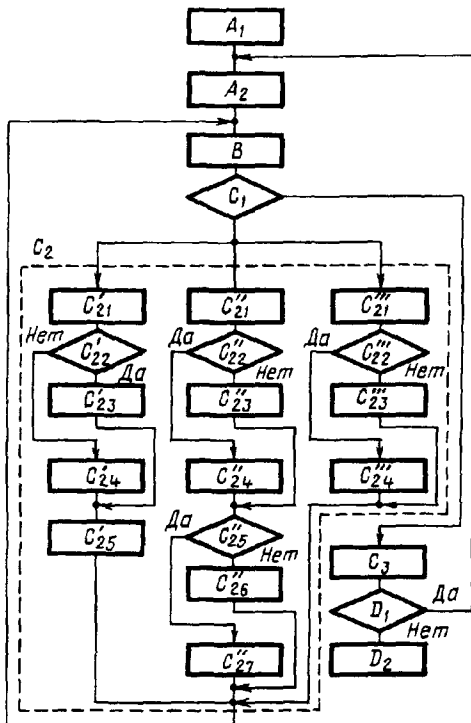


Рис. 8.3. Схема моделирующего алгоритма многофазной многоканальной Q -схемы

Автономность процедур подмодуля S_2 позволяет проводить их параллельное программирование и отладку, причем описанные процедуры могут быть стандартизованы, положены в основу разработки соответствующего математического обеспечения моделирования и использованы для автоматизации процесса моделирования систем.

Если говорить о перспективах, то блочный подход создает хорошую основу для автоматизации имитационных экспериментов с моделями систем, которая может полностью или частично охватывать этапы формализации процесса функционирования системы S , подготовки исходных данных для моделирования, анализа свойств машинной модели M_m системы, планирования и проведения машинных экспериментов, обработки и интерпретации результатов моделирования системы. Такие машинные эксперименты должны носить научный, а не эмпирический характер, т. е. в результате должны предлагаться не только методы решения конкретной поставленной задачи, но и указываться границы эффективного использования этих методов, оцениваться их возможности. Лишь только автоматизация процесса моделирования создаст перспективы использования моделирования в качестве инструмента повседневной работы системного специалиста.

8.2. МОДЕЛИРОВАНИЕ ПРОЦЕССОВ ФУНКЦИОНИРОВАНИЯ СИСТЕМ НА БАЗЕ Q -СХЕМ

Особенности использования при моделировании систем непрерывно-стохастического подхода, реализуемого в виде Q -схем, и основные понятия массового обслуживания были даны в § 2.5. Рассмотрим возможности использования Q -схем для формального описания процесса функционирования некоторой системы S . Характерная ситуация в работе таких систем — появление заявок (требований) на обслуживание и завершение обслуживания в случайные моменты времени, т. е. стохастический характер процесса их функционирования. В общем случае моменты поступления заявок в систему S из внешней среды E образуют входящий поток, а моменты окончания обслуживания образуют выходящий поток обслуженных заявок [6, 13, 39, 51, 53].

Формализация на базе Q -схем. Формализуя какую-либо реальную систему с помощью Q -схем, необходимо построить структуру такой системы. В качестве элементов структуры Q -схем будем рассматривать элементы трех типов: И — источники; Н — накопители; К — каналы обслуживания заявок.

Пример структуры системы S , представленной в виде Q -схемы, приведен на рис. 8.4. Кроме связей, отражающих движение заявок в Q -схеме (сплошные линии), можно говорить о различных управляющих связях. Примером таких связей являются различные блоки-

ровки обслуживающих каналов (по входу и по выходу): «клапаны» изображены в виде треугольников, а управляющие связи — пунктирными линиями. Блокировка канала по входу означает, что этот канал отключается от входящего потока заявок, а блокировка канала по выходу указывает, что заявка, уже обслуженная блокированным каналом, остается в этом канале до момента снятия блокировки (открытия «клапана»). В этом случае, если перед накопителем нет «клапана», при его переполнении будут иметь место потери заявок. Помимо выходящего потока обслуженных заявок можно говорить о потоке потерянных заявок.

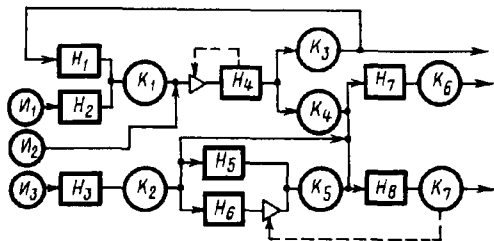


Рис. 8.4. Структура системы, представленной в виде Q -схемы

Как отмечалось выше, Q -схему можно считать заданной, если определены: потоки событий (входящие потоки заявок и потоки обслуживаний для каждого H и K); структура системы S (число фаз L^{Φ} , число каналов обслуживания L^K , число накопителей L^H каждой из L^{Φ} фаз обслуживания заявок и связи I , H и K); алгоритмы функционирования системы (дисциплины ожидания заявок в H и выбора на обслуживание K , правила ухода заявок из H и K).

Рассмотрим возможности формализации воздействий внешней среды E , представляемых в Q -схемах в виде источников (I). Формирование однородных потоков событий, заданных в общем виде многомерным интегральным законом или плотностью распределения вероятностей, т.е.

$$F(y_1, y_2, \dots, y_k) = P\{\tau_1 < y_1, \tau_2 < y_2, \dots, \tau_k < y_k\},$$

$$f(y_1, y_2, \dots, y_k) = F^k(y_1, y_2, \dots, y_k) / (\partial y_1 \partial y_2 \dots \partial y_k),$$

сводится к рассмотренным ранее методам машинной имитации k -мерных векторных величин, требующих больших затрат машинных ресурсов. При моделировании систем, формализуемых в виде Q -схем, часто возникают задачи имитации потоков заявок с некоторыми ограничениями, позволяющими упростить как математическое описание, так и программную реализацию генераторов потоков заявок.

Так, для ординарных потоков с ограниченным последствием интервалы между моментами поступления заявок являются независимыми и совместная плотность распределения может быть представлена в виде произведения частных законов распределения: $f(y_1, y_2, \dots, y_k) = f_1(y_1) f_2(y_2) \dots f_k(y_k)$, где $f_i(y_i)$, $i = \overline{1, k}$, при $i > 1$ являются условными функциями плотности величин y_i при условии, что в момент начала i -го интервала поступит заявка. Относительно начального момента времени t_0 никаких предположений не делается, поэтому функция $f_1(y_1)$ безусловная.

Если поток с ограниченным последствием удовлетворяет условию стационарности, т.е. вероятность появления k событий на интервале $(t_0, t_0 + \Delta t)$ зависит только от длины интервала Δt , то при $i > 0$ интервалы τ_i распределены одинаково, т.е. $f_2(y_2) = f_3(y_3) = \dots = f_k(y_k)$.

Плотность распределения первого интервала $f_1(y_1)$ может быть найдена с использованием соотношения Пальма

$$f_1(y_1) = \lambda \left(1 - \int_0^{y_1} f(y) dy\right), \quad (8.1)$$

где λ — интенсивность потока событий.

Порядок моделирования моментов появления заявок в стационарном потоке с ограниченным последствием следующий. Из последовательности случайных чисел, равномерно распределенных на интервале $(0, 1)$, выбирается случайная величина и формируется первый интервал y_1 в соответствии с (8.1) любым из рассмотренных выше способов формирования случайной величины. Момент наступления первого события $t_1 = t_0 + y_1$, следующие моменты появления событий определяются как

$$t_2 = t_1 + y_2, \dots, t_k = t_{k-1} + y_k, \quad (8.2)$$

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

Пример 8.2. Пусть при моделировании некоторой системы необходимо сформировать на ЭВМ простейший поток заявок. Распределение длин интервалов между заявками является экспоненциальным, т. е. $f(y) = \lambda e^{-\lambda y}$, $y > 0$.

Используем формулу Пальма для определения первого интервала y , т. е.

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

Из этого выражения следует, что $f_1(y_1) = f(y)$, т. е. первый интервал распределен так же, как и остальные. Этого и следовало ожидать ввиду отсутствия последствия в простейшем потоке. Формируя на ЭВМ равновероятностные случайные числа x_i на интервале $(0, 1)$, будем преобразовывать их в соответствии с выражением $\int_0^{y_1} f(y) dy = x_i$. Тогда длина интервала между $(i-1)$ -м и i -м событиями $y_i = -(1/\lambda) \ln x_i$, а моменты появления заявок в потоке определяются согласно (8.2).

Пример 8.3. Пусть при моделировании некоторой системы требуется сформировать на ЭВМ поток событий, равномерно распределенных на интервале (a, b) . Функция плотности интервалов между событиями $f(y) = 1/(b-a)$, $a \leq y \leq b$.

Распределение первого интервала между началом отсчета и первым событием

$$f_1(y_1) = \lambda \left(1 - \int_0^{y_1} f(y) dy\right) = \lambda \left[1 - \int_0^{y_1} dy/(b-a)\right].$$

Интенсивность потока

$$\lambda = 1/M[y] = 1/\int_a^b y f(y) dy = 2/(a+b).$$

Тогда $f_1(y_1) = 2[1 - y_1/(b-a)]/(a+b)$.

Заметим, что математическое ожидание первого интервала $M[y_1]$ отличается от математического ожидания интервалов при $i > 1$:

$$M[y_1] = \int_a^{y_1} y_1 f(y_1) dy_1 = [1/(a+b)] \int_a^{y_1} [y_1 - y^2/(b-a)] dy_1.$$

Длины интервалов между событиями будут

$$\int_a^{y_1} f_1(y_1) dy_1 = x_1, \quad \int_a^{y_1} f(y) dy = x_1.$$

Так, например, при $i > 1$ получим

$$\int_a^{y_i} [1/(b-a)] dy = x_i, \quad y_i = a + (b-a)x_i,$$

где x_i — случайная величина, равномерно распределенная на интервале (0, 1).

Пример 8.4. Рассмотрим формирование на ЭВМ потока Эрланга, в котором между последовательными событиями закон распределения интервалов

$$f_k(t) = \frac{\lambda(\lambda t)^{k-1}}{(k-1)!} e^{-\lambda t}, \quad t > 0.$$

Пусть $k=2$ (поток Эрланга второго порядка). Тогда распределение первого интервала находится по формуле Пальма:

$$f_1(y_1) = (\lambda/2)(e^{-\lambda y_1} + \lambda y_1 e^{-\lambda y_1}) = (\lambda(\lambda + \lambda y_1)/2) e^{-\lambda y_1},$$

$$F_1(y_1) = 1 - (\lambda y_1/2 + 1) e^{-\lambda y_1}.$$

Для определения y_1 решают трансцендентное уравнение вида

$$e^{-2z}(z+1) = x_1, \quad z = \lambda y_1/2.$$

При $i > 1$ интервалы y_i между последовательными событиями в потоке Эрланга второго порядка формируются с учетом того, что y_i представляет собой сумму двух случайных величин y_i' и y_i'' , одинаково распределенных по показательному закону с интенсивностью λ . Для нахождения y_i необходимо определить y_i' и y_i'' и вычислить сумму $y_i' + y_i''$.

Тогда

$$y_i' = -(1/\lambda) \ln x_i', \quad y_i'' = -(1/\lambda) \ln x_i'',$$

$$y_i = -(1/\lambda) (\ln x_i' + \ln x_i'') = -(1/\lambda) \ln (x_i' x_i'').$$

Если реализация моделируемого случайного процесса оказывается достаточно длинной, то можно положить $f_1(y_1) = f_i(y_1)$, $i > 1$, т. е. считать, что все интервалы одинаково распределены. Влияние такого допущения на результаты моделирования системы S будет незначительным.

Проведем анализ принципов формирования потока событий, описываемого нестационарным распределением Пуассона с мгновенной плотностью потока $\lambda(t)$.

С учетом выражения $f(t_0, t) = \lambda(t_0 + t) e^{-\int_{t_0}^{t_0+t} \lambda(t) dt}$ плотность распределения длины первого интервала

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

где $a = \int_{t_0}^{t_0+\Delta t} \lambda(t) dt$ — математическое ожидание числа событий на интервале $(t_0, t_0 + \Delta t)$.

На основании соотношения $\int_0^{y_1} f_1(y_1) dy = x_1$ запишем уравнение

$$a(t_0, y_1) = -\ln x_1. \quad (8.3)$$

Из (8.3) аналитически или любым приближенным способом определяется y_i . Дальнейшая методика моделирования случайной величины y_i при $i > 1$ аналогична формированию y_i с использованием условной функции распределения

$$F_i(y_i/t_{i-1}) = 1 - e^{-a(t_{i-1}, y_i)},$$

где t_{i-1} — момент наступления $(i-1)$ -го события.

Уравнение для нахождения очередного значения интервала имеет вид $a(t_{i-1}, y_i) = -\ln x_i$.

Чтобы описать неординарные потоки событий, для которых $\lim_{t \rightarrow \infty} P_m(t_0, t_0 + \Delta t) \neq 0$, кроме задания законов распределения моментов появления t_i необходимо определять распределение количества событий в рассматриваемый момент времени. Если количество событий, поступающих в систему S в момент времени t_i , не зависит от t_i , то достаточно задать вероятность того, что в произвольный момент времени наступает ровно m событий, т. е. величину $P_m(t_0, t_i)$.

Вопросы построения и машинной реализации программных генераторов, имитирующих потоки событий, были рассмотрены в гл. 4, поэтому более подробно остановимся на особенностях построения моделирующих алгоритмов процесса функционирования таких элементов Q -схем, как накопители (Н) и каналы (К).

Способы построения моделирующих алгоритмов Q -схем. Моделирующий алгоритм должен адекватно отражать процесс функционирования системы S и в то же время не создавать трудностей при машинной реализации модели M_m . При этом моделирующий алгоритм должен отвечать следующим основным требованиям: обладать универсальностью относительно структуры, алгоритмов функционирования и параметров системы S ; обеспечивать одновременную (в один и тот же момент системного времени) и независимую работу необходимого числа элементов системы S ; укладываться в приемлемые затраты ресурсов ЭВМ (машинного времени и памяти) для реализации машинного эксперимента; проводить разбиение на достаточно автономные логические части, т. е. возможность построения блочной структуры алгоритма; гарантировать выполнение рекуррентного правила — событие, происходящее в момент времени t_k , может моделироваться только после того, как промоделированы все события, произошедшие в момент времени $t_{k-1} < t_k$.

При этом необходимо иметь в виду, что появление одной заявки входящего потока в некоторый момент времени t_i может вызвать изменение состояния не более чем одного из элементов Q -схемы, а окончание обслуживания заявки в момент t_i в некотором канале (К) может привести в этот момент времени к последовательному изменению состояний нескольких элементов (Н и К), т. е. будет иметь место процесс распространения смены состояний в направлении, противоположном движению заявок в системе S .

Известно, что существует два основных принципа построения моделирующих алгоритмов: «принцип Δt » и «принцип δz ». При построении моделирующего алгоритма Q -схемы по «принципу Δt », т. е. алгоритма с детерминированным шагом, необходимо для

построения адекватной модели M_m определить минимальный интервал времени между соседними событиями $\Delta t' = \min \{u_i\}$ (во входящих потоках и потоках обслуживания) и принять, что шаг моделирования равен $\Delta t'$. В моделирующих алгоритмах, построенных по «принципу δz », т. е. в алгоритмах со случайным шагом, элементы Q -схемы просматриваются при моделировании только в моменты особых

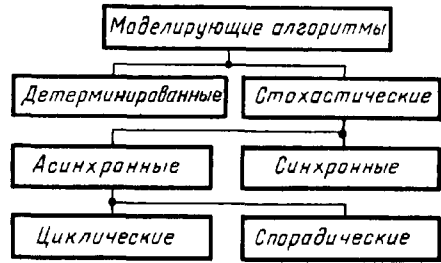


Рис. 8.5. Классификация способов построения моделирующих алгоритмов Q -схем

состояний (в моменты появления заявок из I или изменения состояний K). При этом длительность шага $\Delta t = \text{var}$ зависит как от особенностей самой системы S , так и от воздействий внешней среды E . Моделирующие алгоритмы со случайным шагом могут быть реализованы синхронным и асинхронным способами. При синхронном способе один из элементов Q -схемы (I , N или K) выбирается в качестве ведущего и по нему «синхронизируется» весь процесс моделирования. При асинхронном способе построения моделирующего алгоритма ведущий (синхронизирующий) элемент не используется, а очередному шагу моделирования (просмотру элементов Q -схемы) может соответствовать любое особое состояние всего множества элементов I , N и K . При этом просмотр элементов Q -схемы организован так, что при каждом особом состоянии либо циклически просматриваются все элементы, либо спорадически — только те, которые могут в этом случае изменить свое состояние (просмотр с прогнозированием) [4, 36, 37].

Классификация возможных способов построения моделирующих алгоритмов Q -схем приведена на рис. 8.5.

Более подробно особенности построения и реализации перечисленных разновидностей моделирующих алгоритмов будут рассмотрены при моделировании конкретных вариантов систем.

Особенности моделирования на базе Q -схем. Математическое обеспечение и ресурсные возможности современных ЭВМ позволяют достаточно эффективно провести моделирование различных систем, формализуемых в виде Q -схем, используя либо пакеты прикладных программ, созданные на базе алгоритмических языков общего назначения, либо специализированные языки имитационного моделирования. Но прежде чем применять эти средства автоматизации моделирования, необходимо глубже вникнуть в суть процесса построения и реализации моделирующих алгоритмов [4, 7, 17, 23, 32, 46].

Пример 8.5. Для более детального ознакомления с технологией машинной имитации остановимся на рассмотрении Q -схемы достаточно общего вида, показанной на рис. 8.6. В частности, разберем на данном примере, какое влияние оказывает на

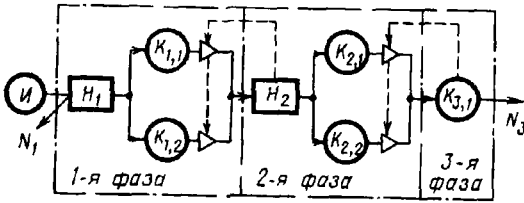


Рис. 8.6. Пример Q -схемы общего вида

могут быть рассмотрены поток потерянных заявок из N_1 и поток обслуженных заявок из $K_{3,1}$ (N_1 и N_3 на рис. 8.6).

Для имитационной модели рассматриваемой Q -схемы можно записать следующие переменные и уравнения: эндогенная переменная P — вероятность потери заявки; экзогенные переменные: t_m — время появления очередной заявки из И; $t_{k,j}$ — время окончания обслуживания каналом $K_{k,j}$ очередной заявки, $k=1, 2, 3; j=1, 2$; вспомогательные переменные: z_i и $z_{k,j}$ — состояния N_i и $K_{k,j}$; $i=1, 2; k=1, 2, 3; j=1, 2$; параметры: L_i — емкость i -го N_i ; L_k^j — число каналов в k -й фазе; $L_1^k=2, L_2^k=2, L_3^k=1$; переменные состояния: N_1 — число потерянных заявок в N_1 , N_3 — число обслуженных заявок, т. е. вышедших из 3-й фазы; уравнение модели: $P=N_1/(N_1+N_3)=N_1/N$.

При имитации процесса функционирования Q -схемы на ЭВМ требуется организовать массив состояний. В этом массиве должны быть выделены: подмассив K для запоминания текущих значений $z_{k,j}$ соответствующих каналов $K_{k,j}$ и времени окончания обслуживания очередной заявки $t_{k,j}$, $j=1, 2, L_k^k$, подмассив N для записи текущего значения z_i соответствующих накопителей N_i , $i=1, 2$; подмассив I , в который записывается время поступления очередной заявки t_m из источника (И).

Процедура моделирования процесса обслуживания каждым элементарным каналом $K_{k,j}$ сводится к следующему. Путем обращения к генератору случайных чисел с законом распределения, соответствующим обслуживанию данных $K_{k,j}$, получается длительность времени обслуживания и вычисляется время окончания обслуживания $t_{k,j}$, а затем фиксируется состояние $z_{k,j}=1$; при освобождении $z_{k,j}=0$; в случае блокировки $K_{k,j}$ записывается $z_{k,j}=2$. При поступлении заявки в N_i к его содержимому добавляется единица, т. е. $z_i=z_i+1$, а при уходе заявки из N_i на обслуживание вычитается единица, т. е. $z_i=z_i-1$, $i=1, 2$.

Детерминированный моделирующий алгоритм. Укрупненная схема детерминированного моделирующего алгоритма Q -схемы, т. е. алгоритма, построенного по «принципу Δt », представлена на рис. 8.7. Специфика наличия постоянного шага Δt позволяет оформить «часы» системного времени в виде автономного блока 10. Этот блок служит для отсчетов системного времени, т. е. для вычисления $t_n=t_{n-1}+\Delta t$. Для определения момента остановки при моделировании Q -схемы (по числу реализаций N или по длине интервала времени моделирования T) проводится проверка соответствующих

особенности построения схемы моделирующего алгоритма принципа, положенный в основу его машинной реализации.

На рисунке представлена трехфазная Q -схема ($L^0=3$) с блокировкой каналов по выходу в 1-й и 2-й фазах обслуживания (пунктирные линии на рисунке). В качестве выходящих потоков такой Q -схемы

условий (блок 3). Работа вспомогательных блоков — ввода исходных данных 1, установки начальных условий 2, обработки 11 и вывода результатов моделирования 12 — не отличается по своей сути от аналогичных блоков, используемых в алгоритмах вычислений на ЭВМ. Поэтому остановимся более детально на рассмотрении работы той части моделирующего алгоритма, которая отражает специфику детерминированного подхода (блоки 4 — 9). Детализированные схемы алгоритмов этих блоков приведены на рис. 8.8, а — е. На этих и последующих схемах моделирующих алгоритмов *Q*-схем приняты следующие обозначения: $ZN(I) \equiv z_i$; $Z(K, J) \equiv z_{k, j}$; $TM \equiv t_m$, $TN \equiv t_n$, $T(K, J) \equiv t_{k, j}$, $LO(I) \equiv L_i$, $PO \equiv P$, $NO1 \equiv N_1$, $NO3 \equiv N_3$, $NO \equiv N$.

Процедура обслуживания заявок каналами $K_{k, j}$ оформлена в виде подпрограммы $WORK [K(K, J)]$, позволяющей обратиться к генератору случайных чисел с соответствующим данному каналу $K_{k, j}$ законом распределения, генерирующему длительность интервала обслуживания очередной заявки $t_{k, j}$. Процедура генерации заявок источником (И) оформлена в виде подпрограммы $D(TM)$, которая определяет момент поступления очередной t_m в *Q*-схему.

Окончание обслуживания заявки в некотором канале K_k в момент времени t_n может вызвать процесс распространения изменений состояний элементов («особых состояний») системы в направлении, противоположном движению заявок в системе, поэтому все N и K системы должны просматриваться при моделировании

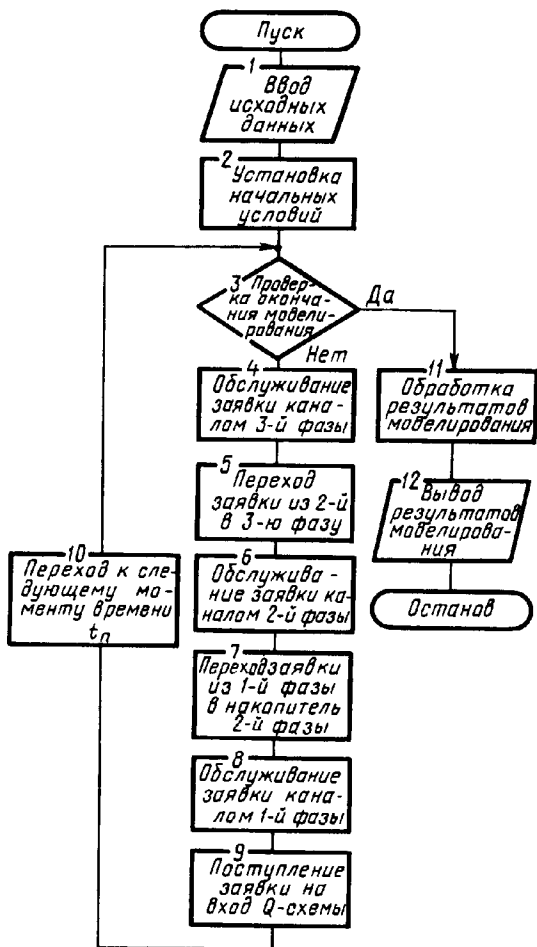


Рис. 8.7. Укрупненная схема детерминированного моделирующего алгоритма *Q*-схемы

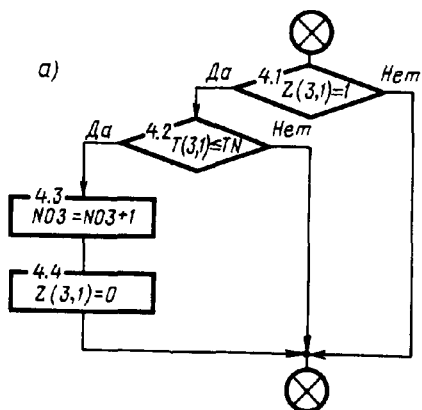
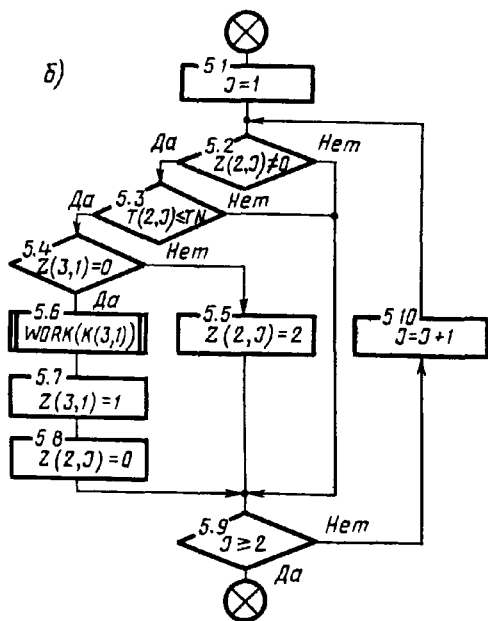


Рис. 8.8. Схемы алгоритмов блока 3 (а), блока 5 (б), блока 6 (в), блока 7 (г), блока 8 (д), блока 9 (е) (рис. 8.7)



Продолжение рис. 8.8

5.6), фиксируются занятость канала 3-й фазы (оператор 5.7) и освобождение канала 2-й фазы (оператор 5.8). Если канал $K_{3,1}$ занят (оператор 5.4), то фиксируется блокировка канала 2-й фазы (оператор 5.5).

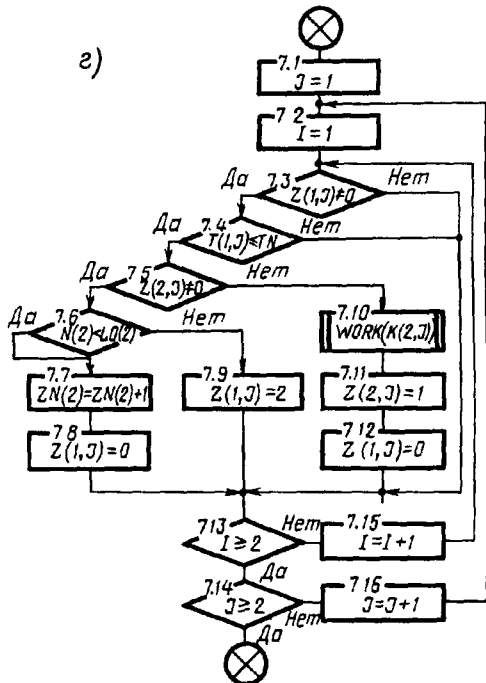
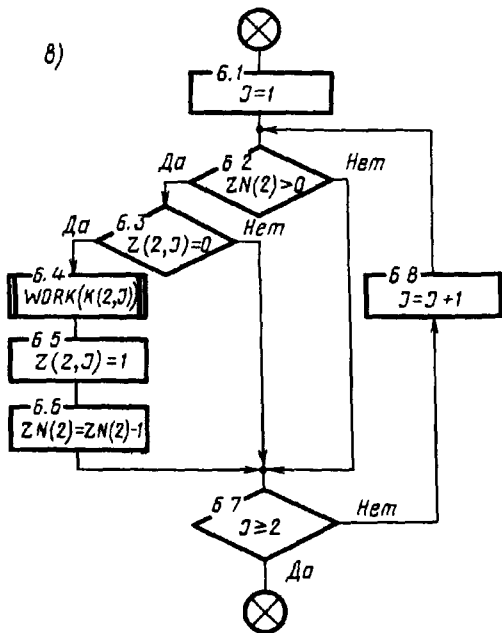
начиная с обслуживающего канала последней фазы по направлению к накопителю 1-й фазы (см. рис. 8.6).

После пуска, ввода исходных данных и установки начальных условий (блоки 1 и 2 на рис. 8.7) проверяется условие окончания моделирования системы (блок 3). Затем переходят к имитации обслуживания заявок каналом $K_{3,1}$ 3-й фазы Q -схемы (рис. 8.8, а). Если закончилось обслуживание в $K_{3,1}$ (операторы 4.1 и 4.2), то фиксируется выход из системы очередной обслуженной заявки (оператор 4.3) и проводится освобождение канала $K_{3,1}$ (оператор 4.4).

Далее реализуется переход к моделированию работы каналов 2-й фазы Q -схемы (рис. 8.8, б). При этом проводится последовательный просмотр каналов этой фазы (операторы 5.1, 5.9 и 5.10). Затем определяется, имеются ли в каналах 2-й фазы заявки, ожидающие обслуживания в канале $K_{3,1}$ (операторы 5.2 и 5.3). Если в момент времени t_n имеются заявки, требующие обслуживания в $K_{3,1}$, и этот канал свободен (оператор 5.4), то выбирается в соответствии с дисциплиной обслуживания одна из заявок и имитируется ее обслуживание $K_{3,1}$ (оператор

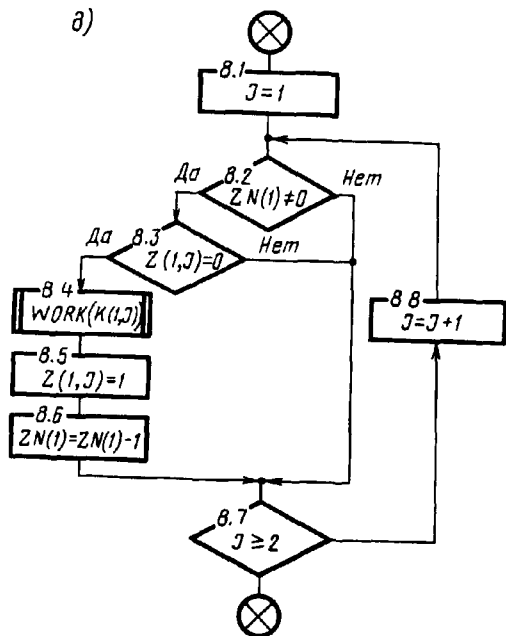
Затем имитируется взаимодействие в процессе обслуживания заявок в накопителе и каналов 2-й фазы последовательно для каждого из каналов (операторы 6.1, 6.7 и 6.8 на рис. 8.8, а). Далее, если в накопителе H_2 имеются заявки (оператор 6.2) и свободные каналы 2-й фазы (оператор 6.3), то имитируется обслуживание заявки одним из свободных каналов (операторы 6.4, 6.5) и освобождение места в накопителе H_2 (оператор 6.6).

Потом имитируется взаимодействие конкретного канала 1-й фазы и накопителя 2-й фазы H_2 (операторы 7.1, 7.2, 7.13 — 7.16 на рис. 8.8, з). Для $K_{1,j}$ проверяется наличие в них заявок, требующих обслуживания в t_n (операторы 7.3 и 7.4). Если нет свободных каналов 2-й фазы (оператор 7.5), но в накопителе имеются свободные места (оператор 7.6), то моделируются запись заявки в H_2 (оператор 7.7) и освобождение конкретного канала 1-й фазы (оператор 7.8). Если свободных мест в H_2 нет, то фиксируется блокировка канала 1-й фазы (оператор 7.9). При наличии свободных каналов 2-й фазы осуществляется обслуживание заявки (оператор 7.10) и фиксируются занятость одного из каналов 2-й фазы (оператор 7.11) и освобождение одного из каналов 1-й фазы



Продолжение рис. 8.8

д)



Продолжение рис. 8.8

(оператор 7.12). Затем операторы 7.3 и 7.4 повторяются, так как одновременно из 1-й фазы во 2-ю могут переместиться две заявки. При третьем выполнении операторов 7.13 и 7.14 управление будет передано по условию «Да» следующему блоку 8 (см. рис. 8.7).

Затем имитируется взаимодействие в процессе обслуживания заявок в накопителе и каналов 1-й фазы (операторы 8.1, 8.7 и 8.8 на рис. 8.8, д). Проверяется необходимость и возможность обслуживания каналами $K_{1,j}$ заявок из накопителя N_1 (операторы 8.2 и 8.3). Если в N_1 имеются заявки и один из $K_{1,j}$ свободен, то имитируется обслуживание заявки в 1-й

фазе (оператор 8.4), фиксируются занятость конкретного канала (оператор 8.5) и освобождение одного места в N_1 (оператор 8.6).

Далее имитируется взаимодействие источника (И) и накопителя 1-й фазы N_1 с учетом занятости каналов этой фазы (рис. 8.8, е). В блоке 9 (см. рис. 8.7) вспомогательными операторами циклов являются операторы 9.2, 9.6 — 9.9 (рис. 8.8, е). Если в t_n поступила заявка из И (оператор 9.1), то она при наличии свободного канала (оператор 9.3) может быть обслужена $K_{1,j}$ (операторы 9.4 и 9.5), при наличии места в N_1 поставлена в очередь (операторы 9.10 и 9.11) либо при отсутствии места в N_1 (его переполнении) потеряна (оператор 9.12). После этого определяется время поступления в Q -схему очередной заявки из источника t_m (оператор 9.13) и управление передается блоку 10, который определяет момент очередного шага t_n (см. рис. 8.7).

Затем управление снова передается блоку 3 (рис. 8.7), который при наборе необходимой статистики проводит обработку и выдачу результатов моделирования, а затем остановку моделирования (блоки 11 и 12).

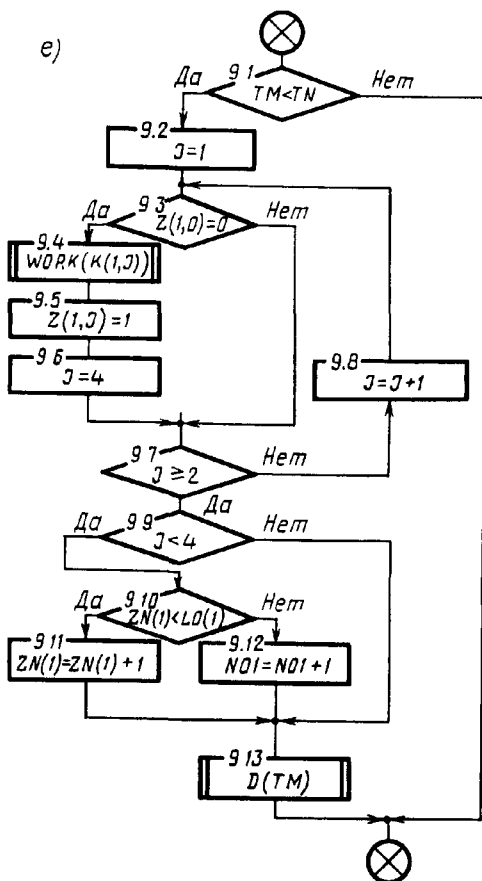
Синхронный моделирующий алгоритм. Рассмотрим особенности построения моделирующих алгоритмов той же Q -схемы, структура

которой приведена на рис. 8.6, по «принципу δz ». Сначала построим синхронный моделирующий алгоритм, причем для определенности примем в качестве синхронизирующего элемента источник (И), т. е. $t_n = t_m$. В момент t_n , т. е. на n -м шаге моделирования, на вход 1-й фазы Q -схемы поступает очередная заявка из И. С момента t_{n-1} до момента t_n в Q -схеме могли произойти изменения состояний N_1 и $K_{1,j}$, если в интервале (t_{n-1}, t_n) либо могло закончиться обслуживание в $K_{1,j}$, либо могли освободиться $K_{2,j}$. Эти изменения необходимо промоделировать раньше, чем поступление заявок в эту фазу в t_n . Это справедливо и для остальных фаз Q -схемы: необходимо моделировать все изменения состояний k -й фазы до поступления в k -ю фазу заявок из $(k-1)$ -й фазы (в этом случае 0-я фаза эквивалентна И).

Каналом $K^*_{k,j}$, имеющим минимальное время окончания обслуживания, является тот, для которого

$$i^*_{k,j} = \min_j \bar{i}_{k,j}, \text{ где } \bar{i}_{k,j} = \begin{cases} t_{k,j}, & \text{если } z_{k,j} = 1; \\ 0, & \text{если } z_{k,j}(t_{n-1}) = 2. \end{cases}$$

Укрупненная схема синхронного моделирующего алгоритма представлена на рис. 8.9. Работа большинства блоков этой схемы аналогична детально рассмотренной схеме детерминированного моделирующего алгоритма (см. рис. 8.7). Поэтому остановимся более подробно только на взаимодействии синхронизирующего элемента, т. е. источника (И) с остальной частью Q -схемы, т. е. рассмотрим работу блока б, имитирующего запись заявки из вход-



Продолжение рис. 8.8

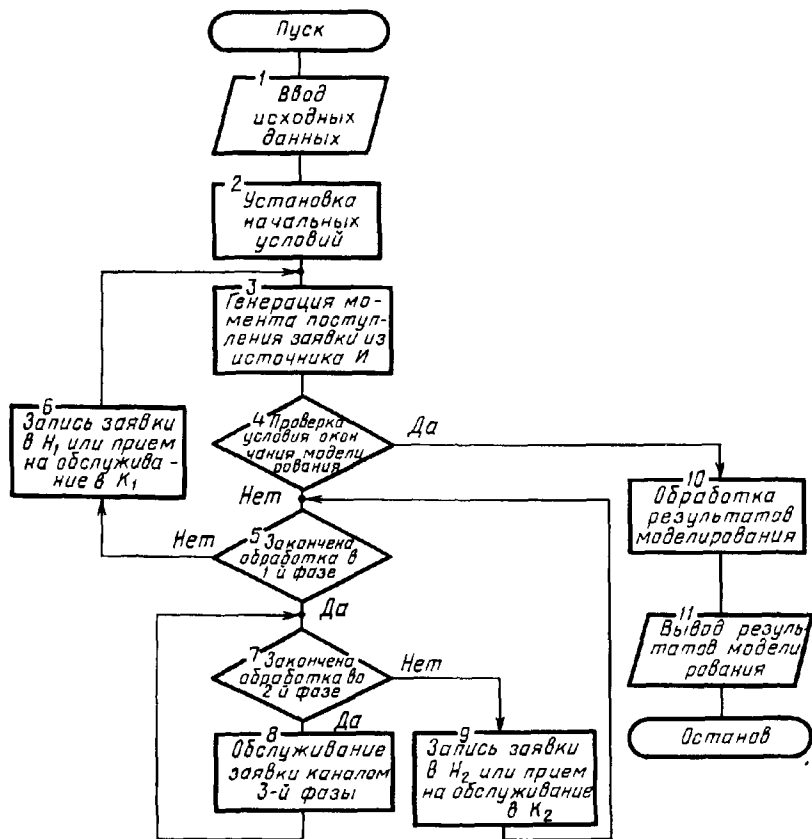


Рис. 8.9. Укрепленная схема синхронного моделирующего алгоритма Q-схемы

ного потока в накопитель H_1 или прием на обслуживание в один из каналов 1-й фазы (рис. 8.10).

На этой схеме вспомогательными являются операторы 6.1, 6.5 — 6.8. Проверяется наличие свободных каналов 1-й фазы (оператор 6.2). Если среди каналов 1-й фазы $K_{1,j}$ есть свободные, то выбирается один из них и имитируется обслуживание, т. е. определяется время окончания обслуживания в этом канале (оператор 6.3), затем фиксируется его новое состояние (оператор 6.4) и осуществляется переход к следующему шагу. Если же оба канала 1-й фазы заняты, то проверяется, есть ли свободные места в накопителе H_1 этой фазы (оператор 6.9). Если свободные места есть, то имитируется запись заявки в H_1 (оператор 6.10), а в противном случае фиксируется потеря заявки (оператор 6.11).

Асинхронный моделирующий алгоритм. Рассмотрим особенности построения асинхронного моделирующего алгоритма, который от-

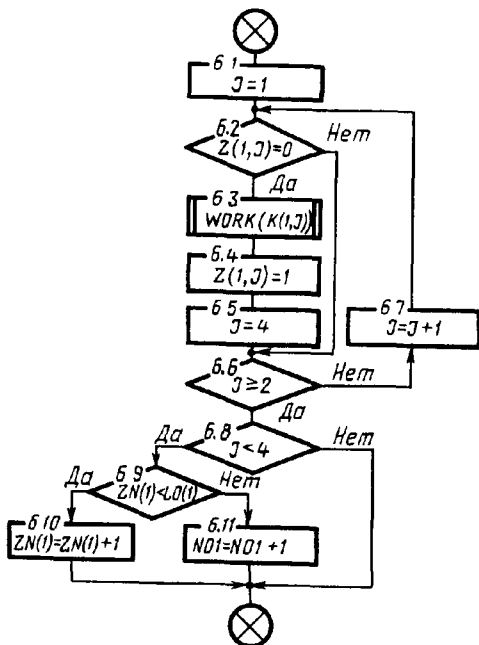


Рис. 8.10. Схема алгоритма блока 6 (рис. 8. 10)

личается от синхронного отсутствием ведущего (синхронизирующего) элемента, причем очередному шагу моделирования соответствует особое состояние, т. е. момент окончания обслуживания одной из заявок любым каналом или момент поступления заявки из источника. При использовании такого принципа построения моделирующего алгоритма целесообразно процесс изменения состояний элементов *Q*-схемы рассматривать в направлении, противоположном направлению движения заявок в системе. Это можно сделать, циклически просматривая на каждом шаге моделирования все элементы *Q*-схемы и определяя, какие переходы заявок из одного элемента в другой могут иметь место в данный момент системного времени. Такой асинхронный циклический моделирующий алгоритм в плане просмотра состояний элементов *Q*-схемы тождествен детерминированному моделирующему алгоритму, который приведен на рис. 8.7. Отличие заключается лишь в том, что отсчет системного времени проводится следующим образом:

$$t_n = \min \left(\min_{k, j} t_{k, j}; \min_m t_m \right),$$

т. е. время очередного шага определяется как минимум из минимальных времен окончания начатого обслуживания всеми каналами

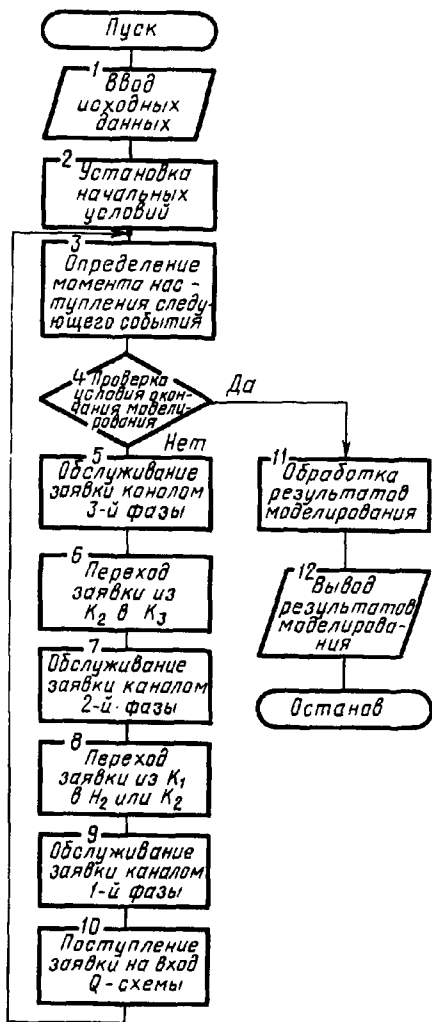


Рис. 8.11. Укрупненная схема асинхронного циклического моделирующего алгоритма Q -схемы

Блок 3 служит для определения временного интервала до ближайшего момента изменения состояния каким-либо элементом Q -схемы (И, Н или К). Системное время

$$t_n = \min \left(\min_{k, j} t_{k, j}; t_m \right)$$

всех фаз Q -схемы и минимального времени поступления очередных заявок из источника. В силу указанных причин не будем подробно останавливаться на рассмотрении асинхронного циклического моделирующего алгоритма Q -схемы, а рассмотрим только укрупненную схему, приведенную на рис. 8.11.

В асинхронных спорадических моделирующих алгоритмах в отличие от циклических для каждого момента системного времени t_n просматриваются только те элементы Q -схемы, которые изменяют свое состояние в этот момент времени. Для моделирования процесса распространения изменений состояний элементов Q -схемы в направлении, противоположном направлению движения заявок в системе, необходимо проследивать цепочку разблокировок в случае освобождения каналов, т. е. рассматривать, вызовет ли освобождение K_k , разблокирование $K_{k-1, j}$, освобождение $K_{k-1, j}$, разблокирование $K_{k-2, j}$, и т. д. Рассмотрим случай, когда эта цепочка просматривается за один шаг моделирования.

Укрупненная схема асинхронного спорадического моделирующего алгоритма, реализующего «принцип δz », показана на рис. 8.12. Рассмотрим подробно отсутствующий в предыдущих схемах блок 3 (рис. 8.13).

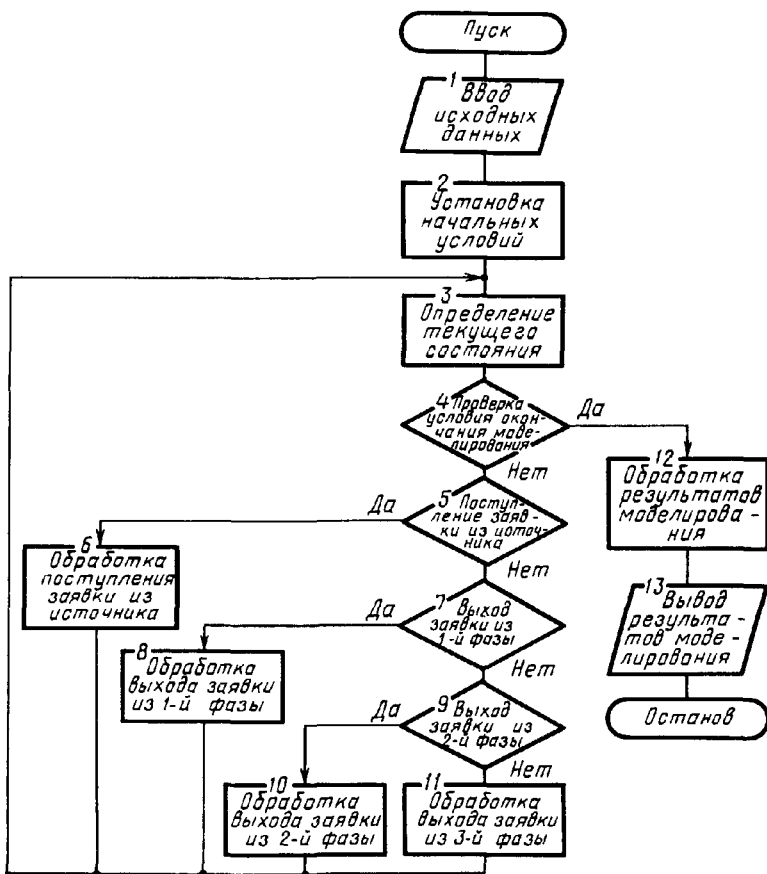


Рис. 8.12. Укрупненная схема асинхронного спорадического моделирующего алгоритма Q -схемы

— это минимальное время освобождения канала $K_{k,j}$, или время до поступления новой заявки из И (операторы 3.13 и 3.14). Поиск минимального времени освобождения канала $K_{k,j}$ реализуется с помощью операторов 3.1 — 3.12. В момент осуществления ближайшего события продвижение состояний реализуется операторами 3.15 и 3.16. Таким образом, в результате работы блока 3 $t_m = 0$, если ближайшим событием является поступление из И, и $t_{k,j} = 0$ и определены k и j , если ближайшим событием является освобождение k -го канала j -й фазы Q -схемы.

Рассмотренные алгоритмы моделирования многофазовой многоканальной Q -схемы, конечно, по своей общности не охватывают всех тех разновидностей Q -схем, которые применяют в практике

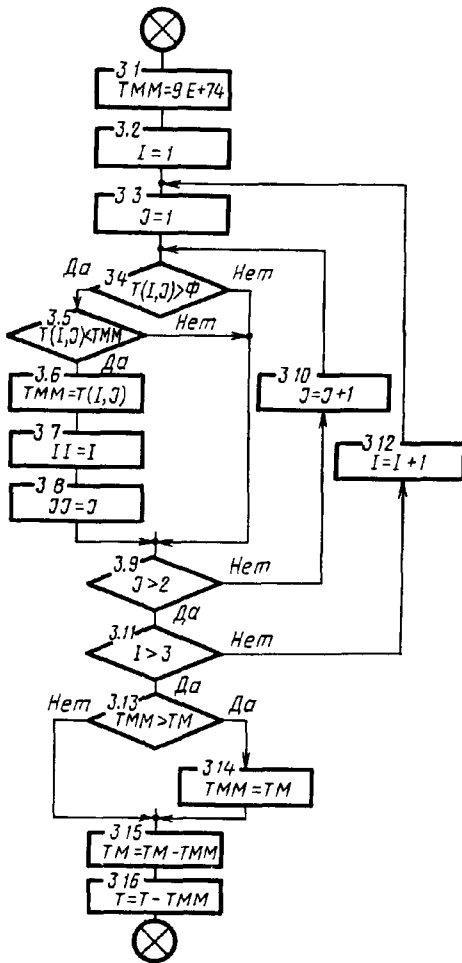


Рис. 8.13. Схема алгоритма блока 3 (рис 8.12)

возможности модификации моделирующих алгоритмов Q -схемы. В плане усложнения машинных моделей M_m при исследовании вариантов системы S можно рассмотреть следующие модификации:

1. Наличие потоков заявок нескольких типов. В этом случае необходимо иметь несколько источников (генераторов) заявок и фиксировать признак принадлежности заявки к тому или иному потоку тогда, когда накопители и каналы рассматриваемой Q -схемы критичны к этому признаку или требуется определить характеристики обслуживания заявок каждого из потоков в отдельности.

2. Наличие приоритетов при постановке заявок в очередь в накопитель. В зависимости от класса приоритета заявок может быть рассмотрен случай, когда заявки одного класса имеют приоритет по записи в накопитель (при отсутствии свободных мест вытесняют из накопителя заявки с более низким классом приоритета, которые при этом считаются потерянными). Этот фактор может быть учтен в моделирующем алгоритме соответствующей Q -схемы путем фиксации для каждого накопителя признаков заявок, которые в нем находятся (путем организации соответствующего массива признаков).

3. Наличие приоритетов при выборе заявок на обслуживание каналов. По отношению к каналу могут быть рассмотрены заявки

анализа и синтеза систем. Однако эти конкретные примеры моделирования позволяют детально ознакомиться с основными принципами построения моделирующих алгоритмов таких систем, причем эти принципы инвариантны к виду и сложности моделируемой системы S .

Возможности модификации моделирующих алгоритмов Q -схемы. В плане усложнения машинных моделей M_m при исследовании вариантов системы S можно рассмотреть следующие модификации:

1. Наличие потоков заявок нескольких типов. В этом случае необходимо иметь несколько источников (генераторов) заявок и фиксировать признак принадлежности заявки к тому или иному потоку тогда, когда накопители и каналы рассматриваемой Q -схемы критичны к этому признаку или требуется определить характеристики обслуживания заявок каждого из потоков в отдельности.

2. Наличие приоритетов при постановке заявок в очередь в накопитель. В зависимости от класса приоритета заявок может быть рассмотрен случай, когда заявки одного класса имеют приоритет по записи в накопитель (при отсутствии свободных мест вытесняют из накопителя заявки с более низким классом приоритета, которые при этом считаются потерянными). Этот фактор может быть учтен в моделирующем алгоритме соответствующей Q -схемы путем фиксации для каждого накопителя признаков заявок, которые в нем находятся (путем организации соответствующего массива признаков).

3. Наличие приоритетов при выборе заявок на обслуживание каналов. По отношению к каналу могут быть рассмотрены заявки

с абсолютным и относительным приоритетами. Заявки с абсолютным приоритетом при выборе из очереди в накопитель вытесняют из канала заявки с более низким классом приоритета, которые при этом снова поступают в накопитель (в начало или конец очереди) или считаются потерянными, а заявки с относительным приоритетом дожидаются окончания обслуживания каналом предыдущей заявки. Эти особенности учитываются в моделирующих алгоритмах приоритетных Q -схем, при определении времени освобождения канала и выборе претендентов на его занятие. Если наличие абсолютных приоритетов приводит к потере заявок, то необходимо организовать фиксацию потерянных заявок.

4. Ограничение по времени пребывания заявок в системе. В этом случае возможно ограничение как по времени ожидания заявок в накопителях, так и по времени обслуживания заявок каналами, а также ограничение по сумме этих времен, т. е. по времени пребывания заявок в обслуживающем приборе. Причем эти ограничения могут рассматриваться как применительно к каждой фазе, так и к Q -схеме в целом. При этом необходимо в качестве особых состояний Q -схемы рассматривать не только моменты поступления новых заявок и моменты окончания обслуживания заявок, но и моменты окончания допустимого времени пребывания (ожидания, обслуживания) заявок в Q -схеме.

5. Выход элементов системы из строя и их дальнейшее восстановление. Такие события могут быть рассмотрены в Q -схеме, как потоки событий с абсолютными приоритетами, приводящими к потере заявок, находящихся в обслуживании в канале или ожидающих начала обслуживания в накопителе в момент выхода соответствующего элемента из строя. В этом случае в моделирующем алгоритме Q -схемы должны быть предусмотрены датчики (генераторы) отказов и восстановлений, а также должны присутствовать операторы для фиксации и обработки необходимой статистики.

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

Дадим краткую сравнительную оценку сложности различных моделирующих алгоритмов Q -схем, в основу построения которых положены перечисленные принципы. Детерминированный и асинхронный циклический алгоритмы наиболее просты с точки зрения логики их построения, так как при этом используется перебор всех элементов Q -схемы на каждом шаге. Трудности возникают с машинной реализацией этих алгоритмов вследствие увеличения затрат машинного времени на моделирование, так как просматриваются все состояния элементов Q -схемы (по «принципу Δt » или по «принципу δz »). Затраты машинного времени на моделирование существенно увеличиваются при построении

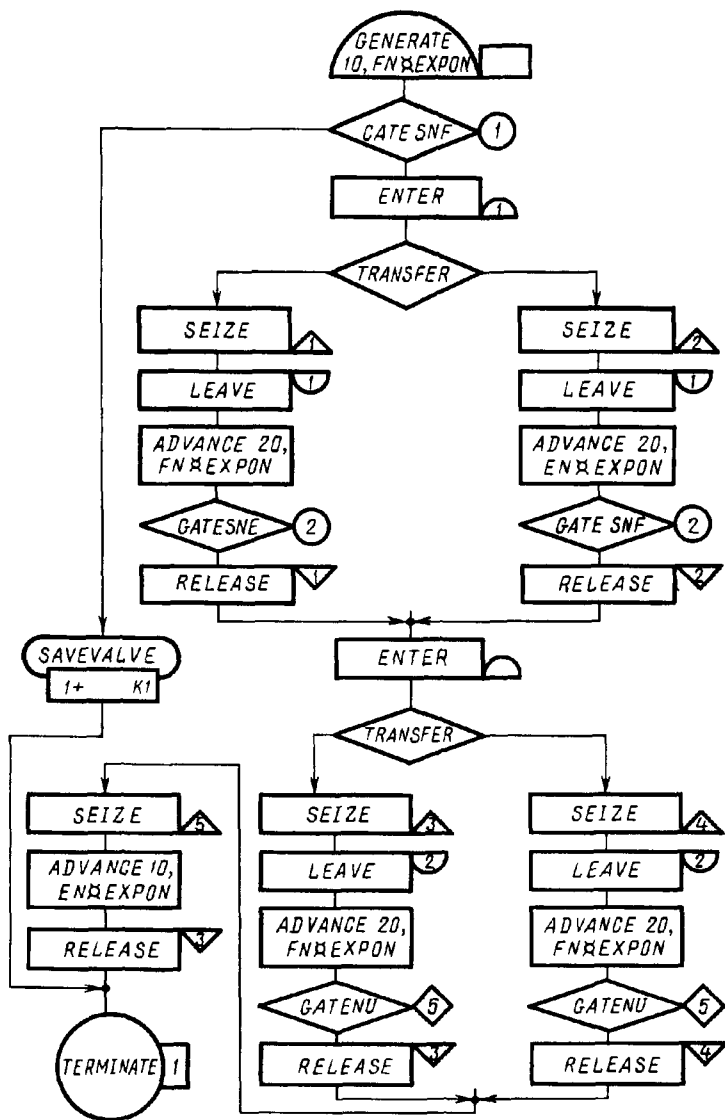


Рис. 8 14 Блок-диаграмма моделирующего алгоритма в символике языка GPSS

детерминированных моделирующих алгоритмов *Q-схем*, элементы которых функционируют в различных масштабах времени, например когда длительности обслуживания заявок каналами многоканальной *Q-схемы* значительно отличаются друг от друга.

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

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

Затраты необходимой оперативной памяти ЭВМ на проведение имитации могут быть значительно уменьшены при построении блочных моделей, когда отдельные блоки (модули) *Q-схемы* реализуются в виде процедур (подпрограмм).

К настоящему времени накоплен значительный опыт моделирования *Q-схем* (при их классическом рассмотрении или в различных приложениях). Рассмотренные моделирующие алгоритмы

```

SIMULATE      Программа имитации многофазной Q-схемы;
1 STORAGE    10
2 STORAGE    10
EXPON FUNCTION RN1 C24
0 0 1 104 2 222 3 355 4 509 5 69
6 915 7 12 75 138 8 16 84 83 88 212
9 23 92 252 94 281 95 299 96 32 97 35
98 39 99 46 995 53 996 62 999 70 9997 80
GENERATE     10 FNMEXPON
GATE SNF     1 OTK
ENTER        1
TRANSFER     BOTH KAN11 KAN12
KAN11 SEIZE  1
LEAVE        1
ADVANCE      20 FNMEXPON
GATE SNF     2
RELEASE      1
TRANSFER     NAK2
KAN12 SEIZE  2
LEAVE        1
ADVANCE      20 FNMEXPON
GATE SNF     2
RELEASE      2
NAK2 ENTER   2
TRANSFER     BOTH KAN21 KAN22
KAN21 SEIZE  3
LEAVE        2
ADVANCE      20 FNMEXPON
GATE NU      5
RELEASE      3
TRANSFER     ,KAN31
KAN22 SEIZE  4
LEAVE        2
ADVANCE      20 FNMEXPON
GATE NU      5
RELEASE      4
KAN31 SEIZE  5
ADVANCE      10 FNMEXPON
RELEASE      5
TRANSFER     END
OTK SAVEVALVE 1+ K1
END TERMINATE 1

```

Рис. 8.15. Программа реализации моделирующего алгоритма на языке GPSS

позволяют практически отразить всевозможные варианты многофазных и многоканальных Q -схем, а также провести исследование всего спектра их вероятностно-временных характеристик, различных выходных характеристик, интересующих исследователя или разработчика системы S .

Рассмотрим особенности моделирования систем, формализуемых в виде Q -схем, с использованием языка имитационного моделирования GPSS. В этом случае отпадает необходимость выбора принципа построения моделирующего алгоритма, так как механизм системного времени и просмотра состояний уже заложен в систему имитации дискретных систем, т. е. в язык GPSS [33, 47].

Пример 8.5. Использование языка GPSS рассмотрим при моделировании Q -схемы, схема которой приведена на рис. 8.6. Блок-диаграмма моделирующего алгоритма в символике языка GPSS представлена на рис. 8.14. Условные обозначения отдельных блоков были приведены в табл. 5.2. Как уже отмечалось, блок-диаграмма языка GPSS позволяет генерировать адекватные программы имитации. Пример программы на языке GPSS показан на рис. 8.15. Действия операторов блок-диаграммы (и программы) GPSS для данного примера приведены в табл. 8.1. При этом приняты следующие обозначения: $NAK1 \equiv N_1$, $KANKJ \equiv K_{k,j}$.

Таблица 8.1

№ карты	№ блока	Назначение оператора (карты)
1	—	Сообщает, что после ассемблирования необходимо начать счет по программе
2	—	Задает емкость накопителя 1
3	—	Задает емкость накопителя 2
4 — 8	—	Описывают функцию экспоненциального распределения с именем EXPON, номером FN1 и значениями в интервале (0, 1)
9	1	Генерирует транзакты с интервалами, распределенными по экспоненциальному закону и средним значением 10 условных единиц
10	2	Проверяет, есть ли свободные места в накопителе 1
11	3	Занимает одно место в накопителе 1
12	4	Направляет транзакт в один из свободных каналов 1 и 2
13	5	Занимает канал 1, 1
14	6	Освобождает одно место в накопителе 1
15	7	Задерживает транзакт на случайный интервал времени в соответствии с экспоненциальным законом со средним значением 20
16	8	Блокирует продвижение транзактов при занятости накопителя 2
17	9	Освобождает канал 1, 1
18	10	Направляет транзакты к блоку с меткой NAK2
19 — 23	11 — 15	Выполняют функции, аналогичные блокам 5 — 9 по отношению к каналу 1, 2
24	16	Занимает одно место в накопителе 2
25	17	Направляет транзакт в один из свободных каналов 2,1 и 2,2
26 — 30	18 — 22	Выполняют функции, аналогичные блокам 5 — 9 по отношению к каналу 2,1

№ карты	№ блока	Назначение оператора (карты)
31	23	Направляет транзакты к блоку с меткой KAN31
32 — 36	24 — 28	Выполняет функции, аналогичные блокам 5 — 9 по отношению к каналу 2,2
37	29	Занимает канал 3,1
38	30	Задерживает транзакт на случайный интервал времени в соответствии с экспоненциальным законом со средним значением 10 условных единиц
39	31	Освобождает канал 3,1
40	32	Направляет транзакты к блоку с меткой END
41	33	Подсчитывает число транзактов, получивших отказ в обслуживании
42	34	Уничтожает транзакт
43	—	Означает конец набора входных карт модели и устанавливает начальное значение счетчика числа завершений, равное 1000
44	—	Передаёт управление операционной системе

Из приведенных примеров моделирования системы S , формализуемой в виде Q -схемы, видно, что при использовании для реализации как универсального алгоритмического языка, так и языка имитационного моделирования $GPSS$ возможности по оценке в процессе имитации вероятностно-временных характеристик исследуемых систем существенно расширяются по сравнению с применением аналитического подхода, когда получение оценок в явном виде ограничено результатами, полученными в теории массового обслуживания.

8.3. МОДЕЛИРОВАНИЕ ПРОЦЕССОВ ФУНКЦИОНИРОВАНИЯ СИСТЕМ НА БАЗЕ N -СХЕМ

Особенности использования при моделировании систем сетевого подхода, реализуемого в виде N -схем, и основные понятия сетей Петри и их модификаций были даны в § 2.6. Рассмотрим возможность применения N -схем для формального описания процесса функционирования некоторой моделируемой системы S . Характерной особенностью N -схем является то, что с их помощью можно моделировать процессы в системах S , в которых происходит последовательная смена дискретных состояний, в том числе если эта смена происходит при выполнении разнообразных условий. Таким образом, с использованием N -схем могут быть описаны системы S , относящиеся к разным классам: аппаратные, физические, программные, экономические и т. д.

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



Рис. 8.16. Моделируемый процесс обслуживания реальной системы

цедуре проектирования [30, 33, 54]. Рассмотрим особенности такого подхода, используя для общности и простоты понятия, введенные в § 8.2 для Q -схем, на следующих примерах.

Пример 8.6. Пусть процесс функционирования некоторой реальной системы S (процессор ЭВМ, мультиплексный канал, станок в технологической цепочке и т. п.), являющийся по своей природе процессом обслуживания, представлен в виде двухфазной одноканальной Q -схемы (рис. 8.16). Тогда этот процесс можно представить N -схемой, структура которой показана на рис. 8.17.

Чтобы маркировать эту структуру, нужно задать состояние системы. Пусть в накопителе H_1 находятся две заявки, в H_2 — заявок нет, канал обслуживания K_1 свободен. Такому состоянию соответствует маркировка, показанная на рис. 8.18. Процесс выполнения этой N -схемы моделирует процесс функционирования системы S , представленной в виде Q -схемы.

Через переход d_1 эта N -схема может быть сведена с другой N -схемой, моделирующей процесс порождения заявок на обслуживание, аналогично — по переходу d_4 — с системой потребления заявок.

Пример 8.7. Пусть имеется некоторая система S , например производственно-технологическая, процесс функционирования которой представлен в виде Q -схемы (рис. 8.19). По технологическому циклу для выполнения заказа необходимо выполнить две фазы обслуживания: сначала обслуживание в канале K_1 , затем либо в K_2 , либо в K_3 . Операторы F_1 и F_2 обслуживают (поддерживают в работоспособном состоянии) каналы, причем F_1 обслуживает K_1 и K_2 , а F_2 — K_1 и K_3 .

Тогда в этой системе могут быть следующие состояния:

- а — заказ пришел и ждет в накопителе H_1 ;
- б — заказ обработан K_1 и ждет в накопителе H_2 ;
- в — заказ выполнен и находится в накопителе H_3 ;
- г — канал K_1 не занят;
- д — канал K_2 не занят;
- е — канал K_3 не занят;
- ж — оператор F_1 не занят;
- з — оператор F_2 не занят;
- и — канал K_1 выполняет заказ под управлением F_1 ;
- к — канал K_1 выполняет заказ под управлением F_2 ;

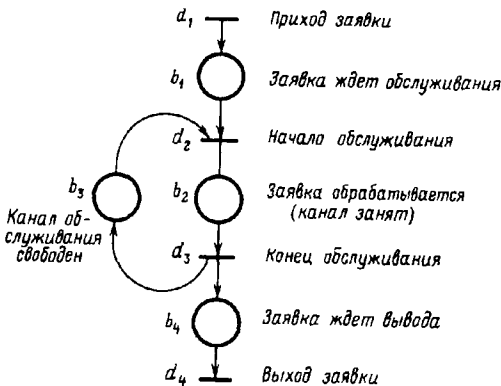


Рис 8 17. Структура N -схемы

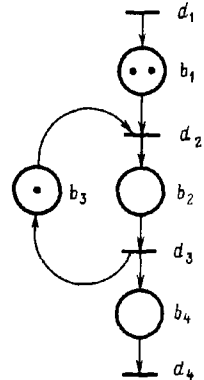


Рис 8 18 Маркировка структуры

л — канал K_2 выполняет заказ под управлением F_1 ;
 м — канал K_3 выполняет заказ под управлением F_2 и могут происходить следующие события-переходы:

- 1 — поступление заказа;
- 2 — F_1 начинает выполнение заказа на K_1 ;
- 3 — F_1 закончил выполнение заказа на K_1 ;
- 4 — F_2 начинает выполнение заказа на K_1 ;
- 5 — F_2 закончил выполнение заказа на K_1 ;
- 6 — F_1 начинает выполнение заказа на K_2 ;
- 7 — F_1 закончил выполнение заказа на K_2 ;
- 8 — F_2 начинает выполнение заказа на K_2 ;
- 9 — F_2 закончил выполнение заказа на K_2 ;
- 10 — заказ отправляется на доставку.

После этого построение N -схемы происходит формально: состояниям системы соответствуют позиции N -схемы, событиям — переходы. Нанесем маркировку, соответствующую такому состоянию системы, при котором каналы свободны, операторы не заняты, в системе нет заказов (рис. 8.20).

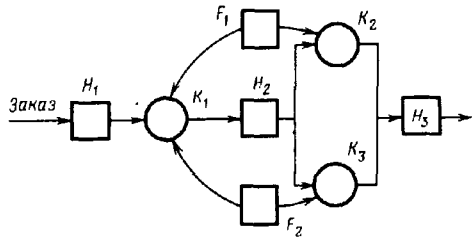


Рис. 8.19. Моделируемый процесс обслуживания

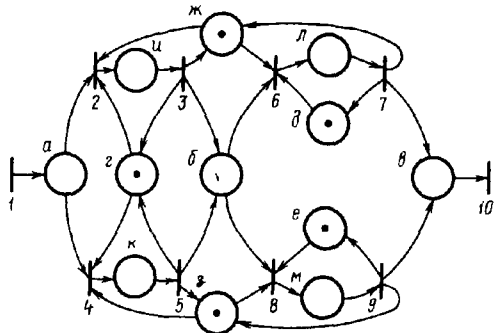


Рис. 8.20. Структура N -схемы

рис. 8.20).

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

Для примера 8.7 построена таблица предусловий и постусловий (табл. 8.2). Эта таблица является описанием структуры N -схемы, удобным для ввода в ЭВМ. Кроме таблицы для выполнения процесса моделирования должна быть задана начальная маркировка в виде n -мерного вектора. Для примера 8.7:

Таблица 8.2

События	Предусловия	Постусловия	События	Предусловия	Постусловия
1	Нет	а	6	б, ж, д	л
2	а, ж, г	и	7	л	в, д, ж
3	и	б, ж, г	8	б, е, з	м
4	а, г, з	к	9	м	е, в, з
5	к	б, г, з	10	в	Нет

$$M = (0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0).$$

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

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

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

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

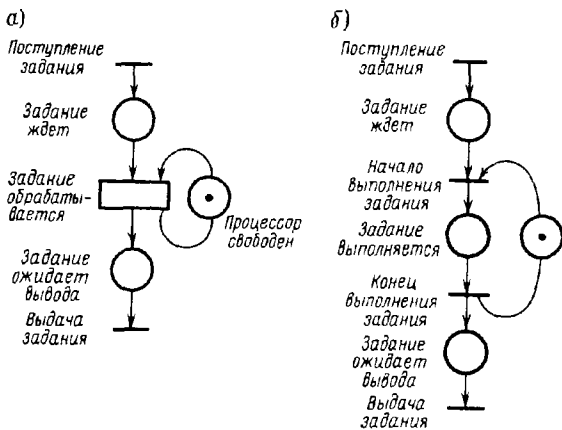


Рис. 8.21. Структура *N-схемы* с непримитивными (а) и примитивными (б) событиями

Пример 8.8. Рассмотрим особенности использования понятий примитивных и непримитивных событий на примере обработки заданий процессором ЭВМ. На рис. 8.21 представлены *N-схемы* (эквивалентные сети Петри) для моделирования обработки задания в процессоре с применением перехода, соответствующего непримитивному событию (рис. 8.21, а), и с применением только переходов — примитивных событий (рис. 8.21, б).

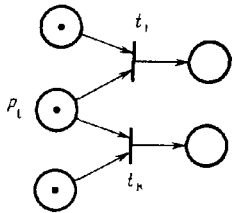


Рис. 8.22. Конфликт двух переходов в *N-схеме*

Ранее упоминалось, что в *N-схемах* все разрешенные переходы срабатывают одновременно и независимо. Однако с помощью *N-схем* можно моделировать и такие системы S , в которых порядок запуска в разрешенных переходах имеет существенное значение. Ситуация, в которой невозможно одновременное выполнение двух разрешенных переходов, изображена на рис. 8.22, где два разрешенных перехода d_j и d_k находятся в конфликте. Может быть запущен только один из них, так как при запуске он удаляет метку из общего входа и запрещает другой переход.

Моделирование параллельных процессов. Возможность моделирования параллелизма и довольно простые процедуры объединения подсистем, представленных *N-схемами*, делают их весьма полезным инструментом моделирования сложных аппаратно-программных информационно-вычислительных комплексов и сетей, состоящих из большого количества одинаковых компонент. Поясним это следующим примером.

Пример 8.9. Рассмотрим процесс функционирования ЭВМ с конвейерной обработкой. При построении высокопроизводительных асинхронных ЭВМ широко применяется метод конвейерной обработки чисел. Этот метод обработки подобен функционированию сборочного конвейера и особенно удобен для работы с векторами и массивами. Конвейер состоит из набора операций, которые могут выполняться одновременно в разных блоках ЭВМ. Когда операция в k -м блоке завершается, ее результат передается в $(k+1)$ -й блок, а k -й блок принимает результат операции из $(k-1)$ -го блока. Если каждая операция запускается по завершении предыдущей, то имеем дело с асинхронным способом управления конвейером. Для управления k -м блоком такого конвейера необходима информация о выполнении следующих условий:

- входной регистр заполнен;
- входной регистр пуст;
- выходной регистр заполнен;
- выходной регистр пуст;
- блок k занят;
- блок k свободен;
- пересылка осуществляется.

На рис. 8.23, а показано, как строится *N-схема* для моделирования асинхронного конвейера такого типа, причем эта модель позволяет анализировать взаимодействия между блоками, игнорируя конкретные детали процессов, которые происходят внутри блоков. Эти процессы в свою очередь могут быть промоделированы *N-схемами* и соединены между собой в соответствии со схемой, показанной на рис. 8.23, б. Такая возможность построения иерархических моделей может быть весьма полезной при моделировании сложных систем S .

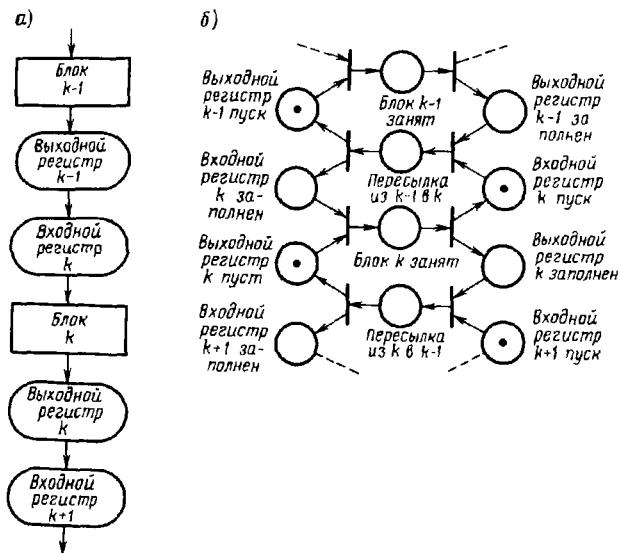


Рис. 8.23. Блок-схема (а) и модель (б) устройства управления асинхронной ЭВМ с конвейерной обработкой

Особенности программирования N -схем. Как уже отмечалось, N -схема представляет собой формализованное описание процесса функционирования системы S , причем структура N -схем отражает причинно-следственные связи в системе S , а совместно с начальной маркировкой — процессы, которые в этой системе происходят. Таким образом, переход от N -схем к моделирующей программе может производиться формальным путем, т. е. автоматически, с использованием специального языка и транслятора. Такие языки и трансляторы созданы за рубежом и в нашей стране [4, 24, 28, 29, 30, 50, 54].

Рассмотрим особенности программирования N -схем, моделирующих процессы в системе S , на конкретных примерах.

Пример 8.10. Рассмотрим принципы программирования N -схем в языке *MODAL*, который предназначен для моделирования и реализации вычислительных алгоритмов на базе N -схем (сетей Петри). Он содержит семь операторов:

- START — описание начальной маркировки;
- STOP — описание конечной маркировки;
- T(MOD) — описание структуры сети Петри,
- TYPE — оператор промежуточного вывода на печать;
- SOMMENT
- COMPLEXITY } вспомогательные операторы;
- END } конец.

Операторы START и STOP представляют собой списки позиций, в которых устанавливаются метки в начале и должны оказаться в конце моделирования

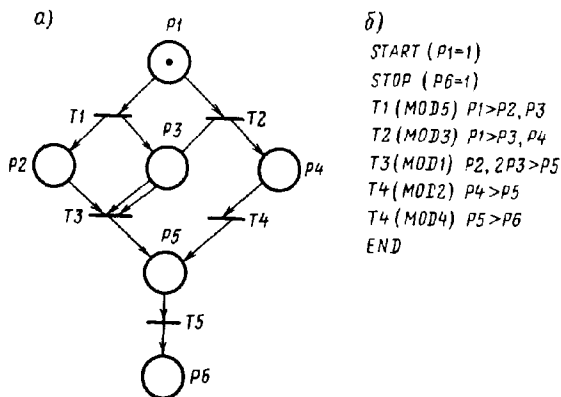


Рис. 8.24. Структура N -схемы с начальной маркировкой (а) и ее описание на языке *MODAL*

Оператор $T(\text{MOD})$ представляет собой таблицу пред- и постусловий для каждого перехода N -схем.

Пусть задана N -схема с начальной маркировкой (рис. 8.24, а). Соответствующее ей описание на языке *MODAL* представлено на рис. 8.24, б.

Кроме описания структуры должны быть составлены программные модули — описания переходов (событий) $\text{MOD1} — \text{MOD5}$, причем все модули должны содержать единую для данного набора *COMMON*-область, которая служит для передачи данных между модулями.

Естественно предположить, что для моделирования конкретных систем S в узкоспециализированной области количество возможных событий-переходов ограничено. Можно составить список стандартных событий и соответствующий ему набор программных модулей. Вместе со средствами описания структуры N -схемы, поясняющей взаимодействия, такой набор составит пакет моделирования систем в специализированной области. Известны системы, построенные по такому принципу для моделирования аппаратных и программных средств вычислительных систем, для моделирования протоколов связи. Известны также примеры успешного применения N -схем (сетей Петри) для исследования социальных, экономических систем, сложных физических и химических процессов.

Расширение N -схем. Возможные расширения N -схем, заданных в виде сетей Петри, о которых говорилось в § 2.6, рассмотрим с точки зрения их прикладного использования, т. е. с ориентацией на увеличение моделирующих возможностей данного аппарата. Для моделирования процессов в информационных системах наибольший интерес представляют N -схемы в виде временных сетей и E -сетей, являющихся наиболее мощным расширением сетей Петри [28].

Задание временной сети, т. е. N_S -схемы, включает семь множеств

$$N_S = \langle B, D, I, O, M, v, v \rangle,$$

где B, D, I, O, M имеют тот же смысл, что и ранее (см. § 2.6); $v = (t_1, \dots, t_i, \dots)$ — возрастающая последовательность действительных чисел, называемая временной базой; $v: B \times v \Rightarrow v$ — функция временных задержек.

Фактор времени учитывается в N_S -схемах путем введения пассивного состояния метки в позиции. При поступлении метки в позицию b_i она остается в пассивном состоянии (не может участвовать в возбуждении переходов) на время $v(b_i, t_S) - t_S$ и только после этого переходит в активное состояние.

Подклассом временной сети является сеть Мерлина, где время пассивного состояния метки определяется как случайная величина, находящаяся в диапазоне между t^* и t^{**} ($t^* \leq t_S \leq t^{**}$), которые задаются в описании N_S -схемы.

Пример 8.11. Построим в виде N_S -схемы модель микропроцессорного абонентского пункта (АП) информационно-вычислительной сети [30, 54]. Структура этого АП показана на рис. 8.25, где обозначено: МП — микропроцессор; ОЗУ — оперативное запоминающее устройство; ПА — периферийный адаптер; ЭПМ — электрифицированная пишущая машинка; КС₁ и КС₂ — входной и выходной каналы связи.

Задание N_S -схемы в виде временной сети включает в себя задание структуры $\langle B, D, I, O \rangle$, вектора разметок M_0 и вектора задержек Z .

Структуру N_S -схемы зададим графически, так как именно графическое представление обладает наибольшей наглядностью и простотой соотношения со структурой объекта, т. е. в данном примере со структурой АП. На рис. 8.26 для наглядности пунктиром выделены элементы N_S -схемы, относящиеся к конкретным элементам структуры АП (рис. 8.25).

Вектор разметок имеет вид

$$M_0 = (010100001110000001 N_{B0}),$$

а вектор задержек

$$Z = (z_{MP}, 0, z_{КС1}, 0, 0, z_{МП}, 0, z_{ЛМ1}, 0, 0, 0, z_{MP}, z_{ПМ2}, 0, z_{КС2}, 0, 0).$$

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

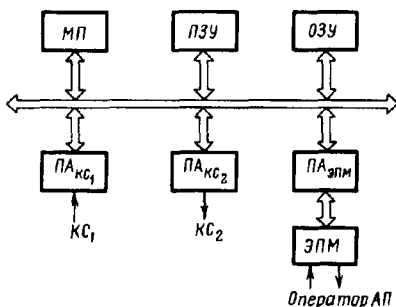


Рис. 8.25. Структура АП

Еще большие возможности для моделирования сложных систем дают такие расширения N -схем, как E -сети, которые обозначим как N_E -схемы. В отличие от временных сетей в E -сетях определено дополнительно четыре типа переходов: разветвление, объединение, управляемое разветвление и приоритетное объединение. Важной особенностью N_E -схем является также детализация представления метки. С каждой меткой в N_E -схеме связа-

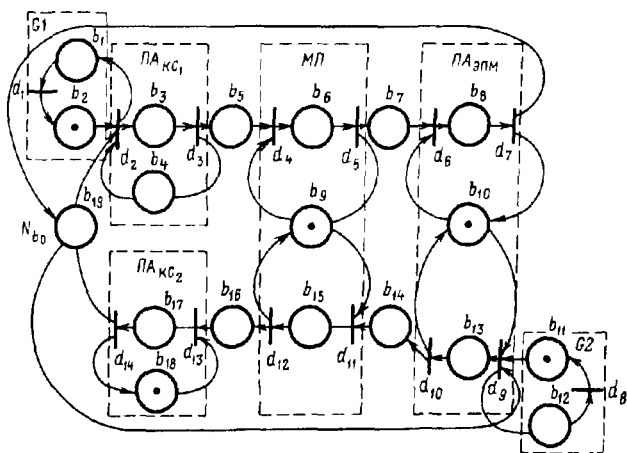


Рис. 8.26. Структура N_S -схемы

ны n описателей. Каждый из описателей метки несет в себе определенную количественную информацию о моделируемом объекте, т. е. системе S .

Структурное задание модели элементов системы S в форме N_E -схемы позволяет использовать модульный принцип разработки имитационной модели с применением библиотеки E -сетевых модулей и их параметрической настройки. В этом случае E -сетевая модель является основой для сборки имитационной программы из модулей, реализованных на некотором языке программирования [30, 54].

Базой для построения программных модулей служит обобщенный алгоритм функционирования E - сетевого модуля, который рассмотрим на следующем примере.

Пример 8.12. Поступление метки во входную позицию перехода d_m (рис. 8.27) инициирует процедуру проверки готовности перехода к срабатыванию (рис. 8.28). Проверяемые при этом условия включают в себя анализ разметки решающей позиции $M\{b_k\}$, где $b_k \in B_R$, а также проверку разметки выходных позиций $M\{0(d_m)\}$ и сопряженных позиций $M\{I(d_m)\}$. В том случае, когда $b_k \in B_p$, на данной фазе производится также вычисление решающей процедуры перехода $z(d_m)$.

При достижении требуемых условий переход вступает в активную фазу, содержание которой определяется элементами описания перехода $i(d_m)$ и $p(d_m)$, причем элемент $i(d_m)$ может быть задан либо непосредственно, либо в виде функции. В последнем случае необходимо предварительно вычислить эту функцию, а затем моделировать требуемое время $t(d_m)$.

Процедура перехода $p(d_m)$ выполняется в два этапа: сначала вычисляются предикаты Π_j , а затем реализуется требуемое подмножество операций над описателями меток $\{l_j\}$. Далее переход d_m вступает в фазу завершения, содержанием которой

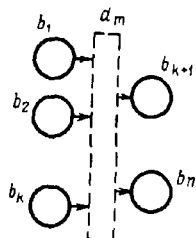


Рис. 8.27. Обобщенная структура E -сетевого перехода

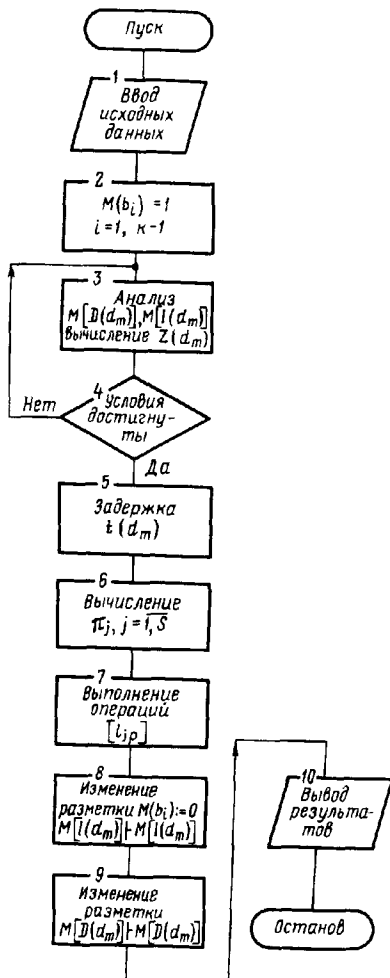


Рис. 8.28. Алгоритм работы E-сетевого перехода

особенности использования ЯИМ для имитации на базе N_E -схем на примере применения системы моделирования GPSS, которая детально была рассмотрена ранее.

Пример 8.13. Проведем анализ соответствия между элементами N_E -схем и объектами языка моделирования GPSS (см. § 5.3). В результате сравнительного анализа можно сформулировать следующее 37.

1. Метка N_E -схемы может быть представлена динамическим объектом GPSS, т. е. транзактом.
2. Описатели меток аналогичны параметрам транзактов GPSS.

является смена разметки входных и выходных позиций перехода $M\{I(d_m)\}$ и $M\{O(d_m)\}$. Разметка позиции b_i меняется на нулевую: $M(b_i) := 0$. При $b_i \in B_j$ разметка решающей позиции становится неопределенной: $M(b_k) := e^0$. Затем выполняется перемещение меток в ряд выходных позиций $O(d_m)$ в соответствии с типом данного перехода. Эта операция завершает работу перехода по обработке поступившей метки.

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

Использование ЯИМ. Программная реализация моделей систем S на базе расширенных N -схем (N_S -, N_E -схем) более сложна по сравнению с программированием моделей на основе обычных сетей Петри. Для упрощения перехода к моделирующей программе рационально использовать языки имитационного моделирования (ЯИМ). Рассмотрим

3. Позиция N_E -схемы идентична объекту, принадлежащему к аппаратной категории GPSS типа накопителя единичной емкости.

4. Решающие позиции $b_k \in B_R$ N_E -схемы в зависимости от принадлежности к множеству B_p реализуются двумя способами: а) если $b_k \in B_p$, то b_k эквивалентна набору объектов типа булевых переменных вычислительной категории GPSS; б) если $b_k \in B/B_p$, то b_k может быть представлена накопителем единичной емкости аппаратной категории GPSS.

5. Временные параметры переходов $t(d_m)$ реализуются объектом ADVANCE операционной категории GPSS.

6. Операции вычисления предикатов Π_j , $j = \overline{1, S}$, для $\rho(d_m)$ соответствует применение блока TESTE, изменяющего маршруты транзактов, в сочетании с булевыми переменными вычислительной категории GPSS.

7. Операции $\{I_p\}$ для $\rho(d_m)$ выполняются с помощью блоков ASSIGN динамической категории GPSS в сочетании с арифметическими переменными.

8. Хранение значений описателей меток можно имитировать путем записи значений параметров транзактов в ячейки хранимых значений (X, XH) с помощью блока SAVEVALVE запоминающей категории GPSS.

9. Процессы синхронизации движения меток через переход d_m и удаления меток из решающей позиции $b_k \in B_R$ могут быть обеспечены с помощью логических переключателей LOGIC S и LOGIC R аппаратной категории GPSS.

10. Макропозиция генератора аналогична блоку GENERATE динамической категории GPSS.

11. Макропозиция поглощения функционально идентична блоку TERMINATE динамической категории GPSS.

12. Макропозиция очереди может интерпретироваться в GPSS записью транзакта в цепь пользователя.

Таким образом, рассмотренное представление моделей элементов информационных систем в виде N_E -схем позволяет упростить этап определения их базовой структуры. Задание модели в виде N_E -схем допускает достаточно простую программную реализацию имитационной модели на ЯОН или ЯИМ.

8.4. МОДЕЛИРОВАНИЕ ПРОЦЕССОВ ФУНКЦИОНИРОВАНИЯ СИСТЕМ НА БАЗЕ А-СХЕМ

Особенности использования при моделировании систем обобщенного агрегативного подхода, реализуемого с помощью А-схем, и основные понятия агрегативных систем были даны в § 2.7. Остановимся на возможностях использования А-схем для формализации процессов функционирования различных систем [4, 36, 37].

Формализация на базе А-схем. Рассмотрим частный случай А-схем в виде кусочно-линейных агрегатов (КЛИА), позволяющих описать достаточно широкий класс процессов и дающих возможность построения на их основе не только имитационных, но и аналитических моделей. В отличие от общей постановки (см. § 2.7) полагаем, что на вход агрегата А не поступают управляющие сигналы $\bar{u}(t)$, т. е. агрегат рассматривается как объект, который в каждый момент времени характеризуется внутренними состояниями $z(t) \in Z$; в изолированные моменты времени на вход агрегата А могут поступать входные сигналы $x(t) \in X$, а с его выхода могут сниматься выходные

сигналы $y(t) \in Y$. Класс КЛА выделяется с помощью конкретизации структуры множеств Z, X, Y , т. е. пространств состояний, входных и выходных сигналов соответственно, а также операторов переходов V, U, W и выходов G .

Пусть имеется некоторое конечное или счетное множество $I = \{0, 1, 2, \dots\}$, которое назовем множеством основных состояний, а элементы этого множества $v \in I$ — основными состояниями. Каждому основному состоянию $v \in I$ поставим в соответствие некоторое целое неотрицательное число $\|v\|$, называемое рангом основного состояния. Кроме того, каждому состоянию $v \in I$ поставим в соответствие выпуклый многогранник $Z^{(v)}$ в евклидовом пространстве размерности $\|v\|$. Будем считать, что $Z = \bigcup_{v \in I} Z^{(v)}$, т. е. пространство состояний Z можно представить состоящим из всевозможных пар вида $(v, \vec{z}^{(v)})$, где $v \in I$, а $\vec{z}^{(v)}$ является вектором размерности $\|v\|$, принимающим значения из многогранника $Z^{(v)}$. Вектор $\vec{z}^{(v)}$ будем называть вектором дополнительных координат. Если $\|v\| = 0$, то в данном основном состоянии v дополнительные координаты не определяются.

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

Определим действие оператора U , описывающего поведение КЛА при отсутствии входных сигналов $x(t)$. Пусть в начальный момент времени агрегат A находится в состоянии $z(t_0) = (v, z^{(v)}(0))$, где $z^{(v)}(0)$ — внутренняя точка многогранника $Z^{(v)}$. Тогда при $t > t_0$ точка $z^{(v)}(t)$ перемещается внутри многогранника $Z^{(v)}$ до тех пор, пока не достигнет его границ. Момент времени t , когда это произойдет, называется «опорным». Тогда при $t_0 < t \leq t_1$ основное состояние агрегата

$$v(t) = v = \text{const} \quad (8.4)$$

и данному состоянию v соответствует вектор дополнительных координат $\vec{a}^{(v)}$ размерности $\|v\|$, причем

$$\vec{z}(t) = \vec{z}(0) + \Delta t \vec{a}^{(v)}, \quad \Delta t = t - t_0. \quad (8.5)$$

Пусть $Z_j^{(v)}$ — j -я грань многогранника $Z^{(v)}$, содержащего $m(v)$ граней, которые могут быть заданы линейными уравнениями вида

$$\sum_{i=1}^{\|v\|} \gamma_{ji}^{(v)} z_i^{(v)} + \gamma_j^{(v)} = 0, \quad j = \overline{1, m(v)},$$

где $z_i^{(v)}$ — компоненты вектора $\vec{z}^{(v)}$, $i = \overline{1, \|v\|}$.

Тогда можно показать, что значение опорного момента t_1 определяется траекторией $z(t)$ и может быть найдено из соотношения

$$t_1 = \min_j \left\{ t : t > t_0, \sum_{i=1}^{\|v\|} \gamma_{ji}^{(v)} [z_i^{(v)}(0) + \Delta t x_j^{(v)}] + \gamma_j^{(v)} = 0 \right\}.$$

Обозначим

$$\tau_j = - \left[\gamma_{j0}^{(v)} + \sum_{i=1}^{\|v\|} \gamma_{ji}^{(v)} z_i^{(v)}(0) \right] / \left[\sum_{i=1}^{\|v\|} \gamma_{ji}^{(v)} x_i^{(v)} \right], \quad j = \overline{1, m(v)}. \quad (8.6)$$

Пусть

$$\tau = \min_j \{ \tau_j > 0 \}, j = \overline{1, m(v)}. \quad (8.7)$$

Тогда

$$t_1 = t_0 + \tau. \quad (8.8)$$

В момент времени t_1 состояние КЛА изменяется скачкообразно и значение $\vec{z}(t_1 + 0)$ является случайным и задается распределением P_1 , которое зависит лишь от состояния $\vec{z}(t_1)$. Широкий класс систем описывается КЛА, у которых P_1 зависит не от всего вектора $\vec{z}(t_1)$, а лишь от значения основного состояния v и номера j -й грани $Z^{(v)}$, на которую вышел вектор дополнительных координат $\vec{z}^{(v)}$.

В момент времени t_1 может выдаваться выходной сигнал y , что описывается оператором G (см. § 2.7). При этом для КЛА

$$z_y = \bigcup_{v=0}^{\infty} \bigcup_{j=1}^{m(v)} Z^{(v)}$$

и множество Y имеет структуру, аналогичную Z , т. е. выходные сигналы $y = (\lambda, \vec{y}^{(v)})$, где λ — элемент некоторого конечного или счетного множества; $\vec{y}^{(v)}$ — вектор, принимающий значения из евклидова пространства размерности, зависящей от λ .

При $t > t_1$ функционирование КЛА вновь описывается формулами (8.4) и (8.5) до очередного особого момента времени t_2 , где $\Delta t = t_2 - t_1$ и т. д.

Для КЛА множество значений входных сигналов X структурно аналогично множествам Y и Z , т. е. входные сигналы $x = (\mu, \vec{x}^{(w)})$, где μ — элементы некоторого конечного или счетного множества; $\vec{x}^{(w)}$ — действительный вектор, размерность которого зависит от μ .

Если в рассматриваемый момент времени t состояние КЛА $z(t) = (v, \vec{z}^{(v)})$ и поступает входной сигнал $x = (\mu, \vec{x}^{(w)})$, то при этом состояние агрегата меняется скачкообразно $z(t+0)$ в соответствии с действием оператора V (см. § 2.7). Состояние $z(t+0)$ является случайным и задается распределением P_2 , которое зависит от $z(t)$ и x . В рассматриваемый момент времени выдается выходной сигнал, необходимость выдачи и содержание которого зависят от состояния $z(t)$ и содержания поступившего входного сигнала x . Далее КЛА снова функционирует в соответствии с (8.4) и (8.5) до следующего момента времени выхода вектора состояний на границу допустимых значений или до момента времени наступления входного сигнала.

Простой вид формул для вычисления «опорных» моментов (8.6) ... (8.8) является следствием кусочно-линейного закона изменения состояний $z(t)$ и обеспечивает простоту машинной реализации модели в виде отдельного КЛА или A -схемы, составленной из нескольких КЛА.

Рассмотрим особенности формализации процессов функционирования системы S , представленных в виде частных типовых математических схем (D - и P -схем), в виде КЛА. При этом надо иметь в виду, что представление процессов функционирования реальных систем в виде КЛА является неоднозначным, так как неоднозначно могут быть выбраны состояния агрегатов [4].

Пример 8.14. Рассмотрим особенности КЛА D -схемы, представляющей собой обыкновенное дифференциальное уравнение первого порядка вида

$$d\vec{z}/dt = f[t, \vec{z}(t), x(t)],$$

где $x(t)$ известная функция времени

Представим это уравнение в конечно-разностном виде, выбрав какой-нибудь метод численного интегрирования уравнения (например, метод ломаных Эйлера), а шаг интегрирования h возьмем из условий близости решения исходного уравнения и его кусочно-линейной аппроксимации. В результате кривая $\hat{z}(t)$, изображающая решение этого уравнения, заменяется ломаной $\hat{z}(t)$, звенья которой в точках $t_k = t_0 + kh, k=0, s$, имеют тангенс угла наклона, равный $f[t_k, \hat{z}(t_k), x(t_k)]$.

Представим систему S , описываемую этим дифференциальным уравнением, в виде КЛА. В качестве состояния агрегата выберем пару $(v, \vec{z}^{(v)})$, где v — основное состояние, которое соответствует номеру интервала времени длины h вида $[t_0 + vh, t_0 + (v+1)h]$; $\vec{z}^{(v)}$ — вектор дополнительных координат, который сформируем следующим образом. В качестве координаты $z_1(t)$ возьмем ломаную конечно-разностного уравнения, а в качестве координаты $z_2(t)$ — время, оставшееся до окончания текущего интервала. Тогда состояние такого КЛА определяется как $z(t) = [v, z_1(t), z_2(t)]$, где координаты $z_1(t)$ и $z_2(t)$ вектора $\vec{z}^{(v)}(t)$ изменятся линейно в пределах интервалов, причем координата $z_2(t)$ убывает с единичной скоростью и обращается в нуль в момент времени $t_v = t_0 + vh, v = \overline{1, s}$. В эти моменты времени состояние совершает детерминированный скачок $z(t_v + 0) = [v+1, z_1(t_v), h]$.

После скачка при $t_v < t \leq t_{v+1}$ координаты описываются соотношениями

$$z_1(t) = z_1(t_v) + (t - t_v)f[t_v, z_1(t_v), x(t_v)]; z_2 = h - (t - t_v),$$

где $x(t_v)$ — входной сигнал, поступающий в моменты времени $t_v = t_0 + vh$.

Таким образом, в этом случае при построении КЛА считается, что $I = \{0, 1, \dots, s\}$, $\|v\| = 2, Z^{(v)} = \{z^{(v)} : z_2^{(v)} \geq 2\}, m(v) = 1, Z_1^{(v)} = \{z^{(v)} : z_2^{(v)} = 0\}$.

Выходными сигналами $y(t)$ могут быть любые функции от состояния.

Пример 8.15. Рассмотрим особенность представления в виде КЛА P -схемы, представляющей собой конечный асинхронный вероятностный автомат Мура, который не имеет жесткой тактности, а изменяет свое состояние только при поступлении входного сигнала. Пусть X_n и Y_n — конечные входной и выходной алфавиты автомата, а Z_n — конечное множество его внутренних состояний. Полагая для определенности, что $X_n = \{1, 2, \dots, K\}, Y_n = \{1, 2, \dots, M\}, Z_n = \{1, 2, \dots, N\}$ функционирование такой P -схемы описывается следующим образом: если в момент времени t автомат находился в состоянии $z_n(t) = i$ и поступил входной сигнал $x_n(t) = k$, то состояние автомата $z_n(t+0) = j$ выбирается случайно с вероятностью $p_j^{(k)} \geq 0, \sum_{j=1}^N p_j^{(k)} = 1, k = \overline{1, K}$.

Выдаваемый при этом выходной сигнал $y_n \in Y_n$ является однозначной функцией нового состояния, в которое перешел автомат, т. е. $y_n = m = \Phi(j)$, где Φ — некоторая детерминированная функция с множеством значений Y_n и областью определения Z_n .

Для представления такой P -схемы в виде КЛА в качестве множества входных сигналов агрегата X выберем множество X_n , а в качестве множества выходных сигналов Y — множество Y_n . В качестве основных состояний КЛА I выберем множество Z_n и будем полагать, что $\|v\| = 0$ для всех $v \in I$, т. е. вектор дополнительных координат $\vec{z}^{(v)}$ не определяется. При таком задании КЛА многогранники $Z^{(v)}$ не определяются, т. е. отпадают вопросы, связанные с движением внутри многогранников, выходом на границу и распределением P_1 .

Таким образом, функционирование такого КЛА сводится к скачкам состояния при поступлении входных сигналов, причем из-за отсутствия вектора дополнительных координат такие скачки сводятся лишь к скачкам основного состояния v , что требует только задания распределения P_2 , которое совпадает с распределением $P_j^{(k)}$. Содержание выходного сигнала, выдаваемого в момент поступления входного сигнала КЛА, определяется только функцией Φ .

Если предположить, что $\|v\| = 0, \|\lambda\| = 0, \|\mu\| = 0$ для всех v, λ, μ , то КЛА превращается в P -схему общего вида.

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

Пример 8.16. Рассмотрим технологию перехода от содержательного описания к *A-схеме* на примере *Q-схемы*, структура которой приведена на рис. 8.6. Такой переход возможен, так как *A-схема* отражает наиболее общий подход к формализации процесса функционирования системы *S*. Для представления этой системы в виде *A-схемы* будем использовать пять типов агрегатов, а именно: A^E — внешняя среда; A^H — накопитель; A^K — канал; A^P — распределитель; A^C — сумматор. Функции агрегатов A^H и A^K соответствуют функциям таких элементов *Q-схемы*, как накопитель (Н) и канал (К). Агрегат A^E позволяет формализовать взаимодействие между агрегатами *A-схемы* и внешней средой *E*. Использование вспомогательных агрегатов A^P и A^C вызвано необходимостью синхронизации работы агрегатов в составе *A-схемы* в соответствии с принятыми дисциплинами постановки в очередь и обслуживания заявок. Кроме того, через агрегат A^C возможна передача сигналов от различных выходных контактов одних агрегатов на один и тот же входной контакт другого агрегата, что запрещено делать непосредственно (см. § 2.7).

При таких предположениях структура *A-схемы* будет иметь вид, приведенный на рис. 8.29. Опишем работу каждого типа агрегатов, показанных на рис. 8.30, в отдельности.

Агрегат «Внешняя среда» A^E (рис. 8.30, а) имеет два входных контакта и один выходной: на вход $X_1^{(E)}$ поступают обслуженные заявки (сигнал $x_1^{(E)}=1$); на вход $X_2^{(E)}$ — заявки, получившие отказ в обслуживании (сигнал $x_2^{(E)}=1$); с выхода $Y_1^{(E)}$ снимают заявки через промежутки времени, распределенные по заданному закону распределения входящего потока заявок. Вектор состояний агрегата $A^E: \vec{z}^E(t) = 1$.

Агрегат «Канал» A^K (рис. 8.30, б) имеет три входных контакта и один выходной: на вход $X_1^{(K)}$ подают сигнал поступления заявки на обслуживание; на вход $X_2^{(K)}$ — сигнал разрешения выдачи обслуженной заявки; на вход $X_3^{(K)}$ — сигнал блокировки

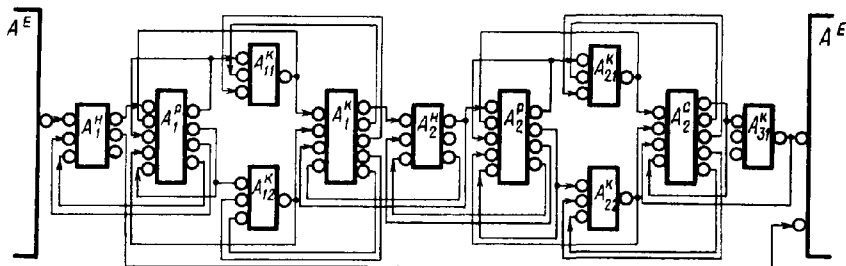


Рис. 8.29. Пример *A-схемы* общего вида

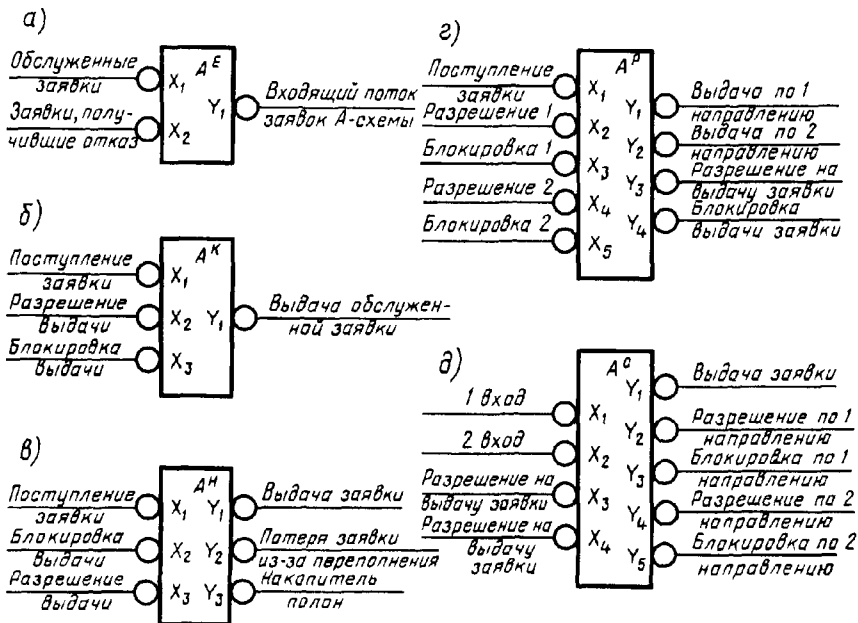


Рис. 8.30. Агрегаты

выдачи обслуженной каналом заявки; с выхода $Y_1^{(K)}$ снимают сигнал выдачи обслуженной каналом заявки. Вектор состояний агрегата

$$\vec{z}^K(t) = \{z_1^K(t), z_2^K(t), z_3^K(t)\},$$

где $z_1^K(t)$ — время, оставшееся до окончания обслуживания заявки, которая находится в канале;

$$z_2^K(t) = \begin{cases} 1, & \text{если блокировка отсутствует,} \\ 0 & \text{в противном случае;} \end{cases}$$

$$z_3^K(t) = \begin{cases} 1, & \text{если заявка находится в канале,} \\ 0 & \text{в противном случае.} \end{cases}$$

В том случае, если время обслуживания заявки в канале истекло, т. е. $z_1^K(t) \leq 0$, но ее выдача из канала запрещена, т. е. $z_2^K(t) = 0$, заявка остается в канале до тех пор, пока не придет сигнал $z_2^K(t) = 1$.

Агрегат «Накопитель» A^H (рис. 8.30, в) имеет три входа и три выхода, входные контакты $X_1^{(H)}$, $X_2^{(H)}$, $X_3^{(H)}$ соответствуют по своим функциям контактам $X_1^{(K)}$, $X_2^{(K)}$, $X_3^{(K)}$ агрегата A^K , с выхода $Y_1^{(H)}$ выдается заявка, стоящая в очереди в накопителе первой; с выхода $Y_2^{(H)}$ выдаются заявки, потерянные из-за переполнения накопителя; с выхода $Y_3^{(H)}$ поступает сигнал о том, что накопитель полностью заполнен. Внутреннее состояние агрегата A^H описывается вектором

$$\vec{z}^H(t) = \{z_1^H(t), z_2^H(t)\},$$

где $z_1^H(t)$ — число заявок в накопителе;

$$z_2^H(t) = \begin{cases} 1, & \text{если блокировка отсутствует,} \\ 0 & \text{в противном случае.} \end{cases}$$

Агрегат «Распределитель» A^P (рис. 8.30, з) служит для разделения поступающего на вход X_1^P потока заявок по двум направлениям — выходам Y_1^P и Y_2^P , что соответствует принятой дисциплине обслуживания, т. е. алгоритму взаимодействия накопителя и канала. В рассматриваемом примере поступившая в A^P -заявка передается через выход Y_1^P , если соответствующий ему агрегат A^K свободен для принятия на обслуживание этой заявки; в противном случае заявка выдается через выход Y_2^P . Информация о занятости соответствующих агрегатов A^K , на которые поступают заявки с выходов Y_1^P , Y_2^P , передается на входные контакты $X_1^K - X_2^K$. Если оба агрегата A^K не могут принять заявки от агрегата A^P , то на выходной контакт Y_1^P выдается сигнал, запрещающий передачу заявки A^P . Как только один из агрегатов A^K освободится (о чем выдается соответствующий сигнал A^P на входы $X_1^K - X_2^K$), сигнал с контакта Y_3^P разрешает посылающему агрегату A^H пересылку заявки через A^P в A^K , т. е. заявка в A^P не хранится. Внутреннее состояние агрегата определяется вектором

$$\vec{z}^P(t) = \{z_1^P(t), z_2^P(t)\},$$

где

$$z_1^P(t) = \begin{cases} 1, & \text{если разрешена передача заявки по выходу,} \\ 0 & \text{в противном случае;} \end{cases}$$

$$z_2^P(t) = \begin{cases} 1, & \text{если разрешена передача заявки по выходу,} \\ 0 & \text{в противном случае.} \end{cases}$$

Агрегат «Сумматор» A^C (рис. 8.34, д) выполняет функции, обратные агрегату A^P , т. е. избирательно суммирует поступающие заявки от двух посылающих агрегатов A^K и передает их на вход принимающего агрегата (A^H и A^K). При поступлении на вход X_1^C или X_2^C сигнала разрешения передачи заявки от принимающего агрегата сумматор A^C должен последовательно опросить принимающие агрегаты. Для этого сначала передается сигнал разрешения на первый из посылающих агрегатов (контакт Y_1^C). От этого посылающего агрегата либо поступает сигнал на входной контакт X_1^C , либо он отсутствует. В первом случае поступившая заявка передается дальше (контакт Y_1^C), в противном случае посылается сигнал разрешения (контакт Y_2^C) на второй из посылающих агрегатов. Внутреннее состояние агрегата A^C определяется вектором

$$\vec{z}^C(t) = \{z_1^C(t)\},$$

где

$$z_1^C(t) = \begin{cases} 1, & \text{если выдача заявки разрешена,} \\ 0 & \text{в противном случае.} \end{cases}$$

Таким образом, используя набор агрегатов A^E , A^K , A^H , A^P и A^C , можно описать процесс функционирования рассматриваемой системы (см. рис. 8.6). Для возможности формализации более сложных систем требуется в пределах данного класса объектов (Q -схем) увеличение числа состояний перечисленных агрегатов, а для других классов систем — расширение набора агрегатов.

Моделирующий алгоритм A -схемы. Укрупненная схема моделирующего алгоритма такой системы, представленной в виде A -схе-

мы, приведена на рис. 8.31. В основу моделирования положен принцип просмотра состояний модели в моменты скачков, т. е. «принцип δz » («принцип особых состояний»). Обработка каждого особого состояния выполняется блоками 6 и 12. Работа такого блока иллюстрируется схемой, представленной на рис. 8.32, и сводится к выбору типа агрегата (A^E , A^K , A^H , A^P и A^C), для которого реализуется дальнейшее «продвижение» при моделировании.

Схема моделирующего алгоритма, имитирующего воздействие на систему S внешней среды E , показана на рис. 8.33, а. При этом определяется, какое событие имело место, поступление или выдача сигнала из внешней среды, т. е. заявки входного потока в A -схему (блок 6.2.1). При наступлении времени выдачи заявки она выдается в A -схему (блок 6.2.2) и генерируется интервал времени между моментом поступления новой заявки (блок 6.2.3).

Схемы моделирующих алгоритмов, имитирующих работу агрегатов A^K и A^H , приведены на рис. 8.33, б, в соответственно. Работа

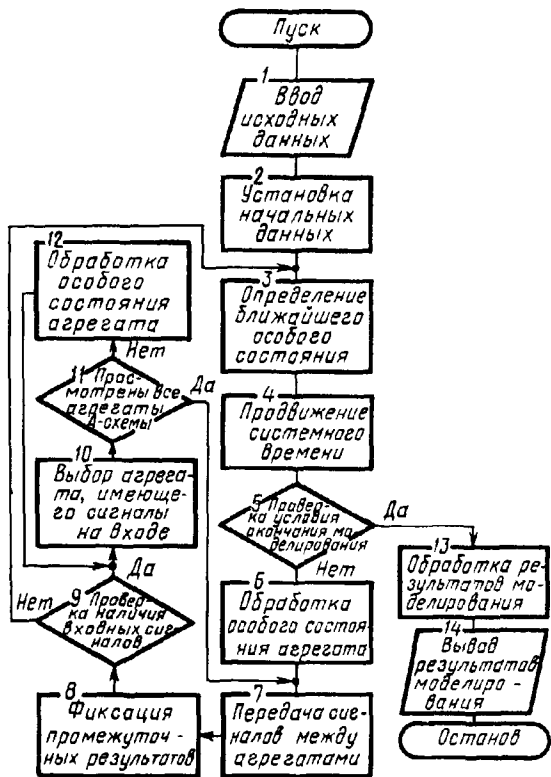


Рис. 8.31. Укрупненная схема моделирующего алгоритма A -схемы

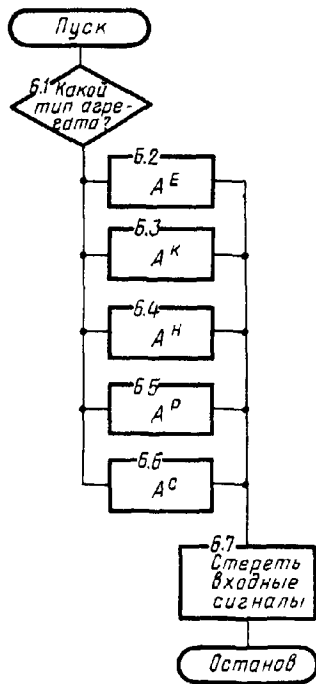


Рис. 8.32. Блок-схема алгоритма блока 6 (рис. 8.31)

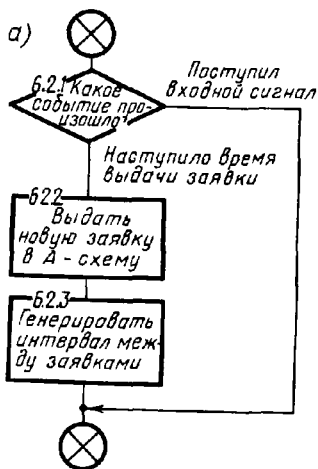
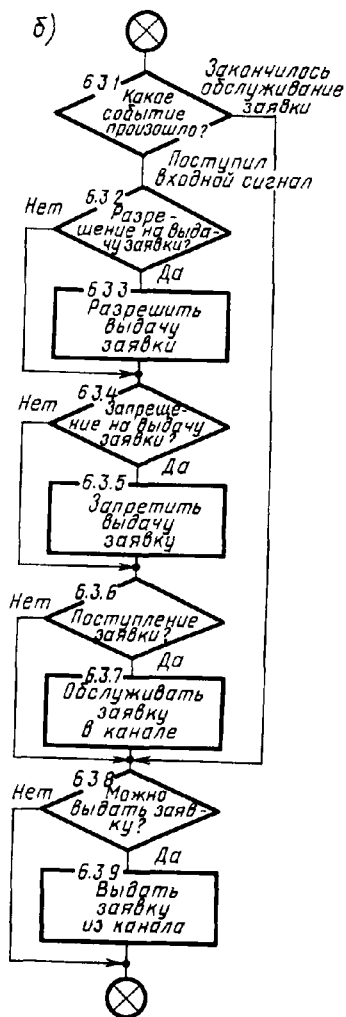


Рис. 8.33. Схемы алгоритмов блока 6.1 (а); блока 6.3 (б); блока 6.4 (в); блока 6.5 (г); блока 6.6 (д) (рис. 8.36)

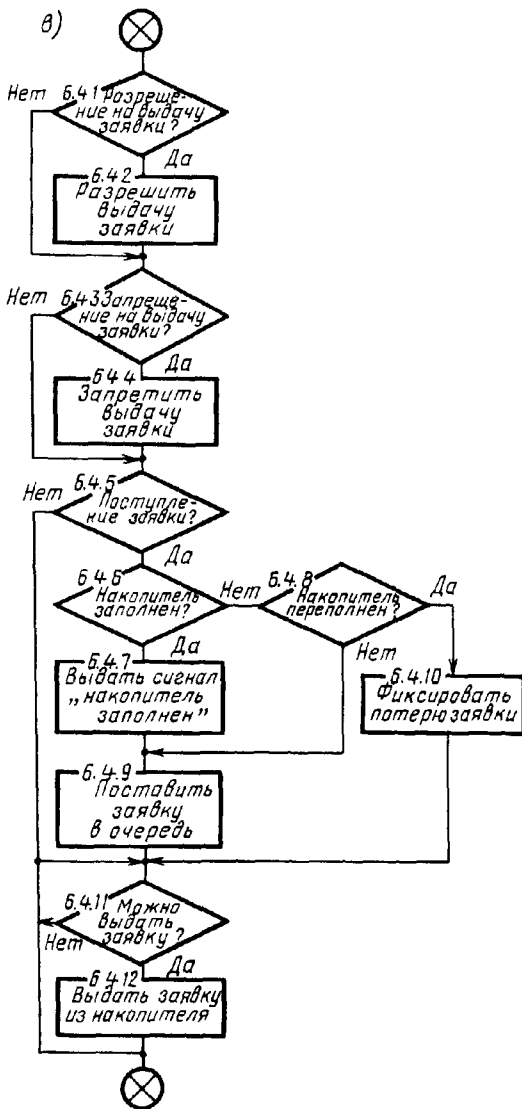
ЭТИХ СХЕМ полностью соответствует описанию процесса функционирования агрегатов A^K и A^H , показанных на рис. 8.30, б, в.

Схемы работы агрегатов A^P и A^C , выполняющих вспомогательные функции сопряжения агрегатов, показаны на рис. 8.33, г, д соответственно. Они реализуют взаимодействие основных агрегатов A^E , A^K и A^H , разрешая или запрещая передачу сигналов между ними в зависимости от ситуации с учетом правил обмена сигналами в A -схеме, описанных в § 2.7. При этом в схемах предусмотрено тестирование ошибок (блоки 6.5.8 и 6.6.6), связанных с нарушением при задании исходных данных этих правил обмена сигналами в A -схеме.

Из рассмотренного примера моделирования конкретной системы S (в данном случае заданной в виде Q -схемы) видно, что агрегативный подход является тем фундаментом, на котором базируется построение автоматизированной имитационной системы,



Продолжение рис. 8.33



Продолжение рис. 8.33

разработка ее внутреннего и внешнего математического и программного обеспечения. При этом стандартная форма математической модели исследуемого объекта позволяет не только унифицировать моделирующие алгоритмы, но и применять также стандартные методы обработки и анализа результатов моделирования, реализованные в виде специальных библиотек программ [7, 10, 12, 21].

Применение агрегативного подхода при моделировании систем дает ряд преимуществ по сравнению с другими, менее универсальными подходами. Так, агрегативный подход в силу модульной структуры модели и дискретного характера обмена сигналами дает возможность использовать внешнюю память ЭВМ для хранения сведений о моделируемых объектах, что в значительной степени снижает ограничения по сложности, возникающие при попытке представить процесс функционирования моделируемой системы S в целом как последовательность взаимосвязанных системных событий для записи его в виде

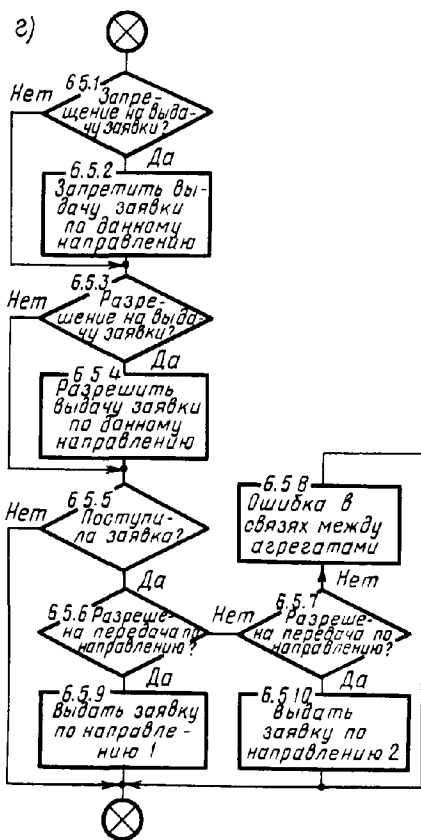
моделирующего алгоритма или на языке имитационного моделирования. При этом объем программ имитации мало зависит от сложности моделируемого объекта, которая определяет лишь число операций, требуемых для реализации машинной модели M_c , и объем памяти, необходимой для хранения сведений об агрегатах и их

связях. Важно, что такие имитационные программы позволяют проводить их предварительную автономную отладку и являются программами многоразового использования, что повышает оперативность решения задач моделирования систем. При наличии таких отлаженных программных модулей время подготовки к моделированию практически совпадает со временем формализации моделируемой системы S в виде A -схемы и задания исходных данных.

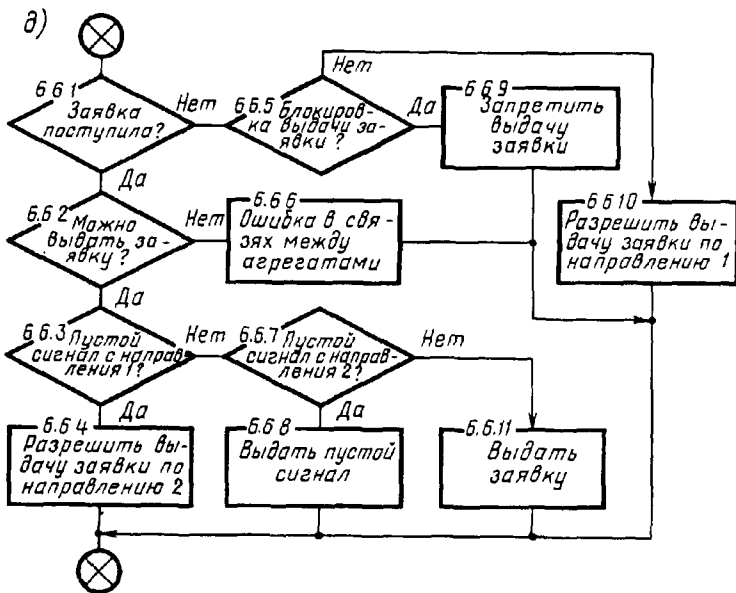
При агрегативном подходе возникают и некоторые трудности, например, связанные с организацией диалога пользователя с имитационной системой, так как представление моделируемой системы в виде A -схемы предполагает и структуризацию в соответствующем виде входных данных. Следовательно, пользователь, как и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

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

Таким образом, использование типовых математических схем, рассмотренных в данной главе на примере Q - и A -схем, позволяет формализовать процесс функционирования конкретной системы S ,



Продолжение рис. 8.33



Продолжение рис. 8.33

т. е. переход от концептуальной модели системы M_m к ее машинной модели M_m . Типовые математические схемы при моделировании конкретных систем будут рассмотрены в гл. 10.

Контрольные вопросы

- 8.1. Какие основные блоки выделяются при построении иерархической модели системы?
- 8.2. Какие существуют способы построения моделирующих алгоритмов *Q-схем*?
- 8.3. Чем отличаются синхронный и асинхронный моделирующие алгоритмы *Q-схем*?
- 8.4. В чем суть структурного подхода при моделировании систем на базе *N-схем*?
- 8.5. Каковы особенности использования языков имитационного моделирования на базе *N-схем*?
- 8.6. В чем заключаются особенности формализации процессов функционирования систем на базе *A-схем*?
- 8.7. Каково преимущество использования типовых математических схем при имитационном моделировании?

ГЛАВА 9

МОДЕЛИРОВАНИЕ ДЛЯ ПРИНЯТИЯ РЕШЕНИЙ ПРИ УПРАВЛЕНИИ

Машинное моделирование является эффективным инструментом исследования характеристик процесса функционирования сложных систем на этапе их проектирования. Но этим возможности этого метода не ограничиваются: в современных системах управления машинное моделирование используется непосредственно в контуре управления, на его основе решаются задачи прогнозирования для принятия решений по управлению объектом, т. е. реализуются адаптивные системы управления. Построение таких адаптивных систем стало возможным, с одной стороны, после решения ряда вопросов информационного подхода к проблеме управления, а с другой стороны, после проработки задач моделирования в реальном масштабе времени на современных ЭВМ с учетом ограниченности ресурсов в системе управления объектом.

9.1. ГНОСЕОЛОГИЧЕСКИЕ И ИНФОРМАЦИОННЫЕ МОДЕЛИ ПРИ УПРАВЛЕНИИ

Создание системы управления (СУ) различными объектами требует наличия большого объема информации как о самом объекте, так и о его входных и выходных переменных. Эта информация необходима для построения адекватной модели СУ, на основе которой может быть эффективно осуществлен процесс управления. При этом следует различать два вида информации, необходимой для построения и совершенствования модели и СУ: априорную и текущую. Априорная информация об объекте управления (ОУ), его входных и выходных переменных, внутренних состояниях необходима для построения модели, по которой будет создаваться СУ этим объектом: выбираться структура, алгоритмы и параметры СУ, критерий функционирования. Обычно для сложных вновь проектируемых ОУ отсутствует необходимая для создания СУ модель, и задача управления должна решаться в условиях недостаточной или вовсе отсутствующей априорной информации об объекте. Речь идет об отсутствии информационной («управленческой») модели ОУ, устанавливающей взаимосвязь между выходными и входными переменными [41, 43, 54].

Особенности системы управления. Проблема создания СУ неизбежно возникает при разработке ОУ и при их модернизации. На первый взгляд может показаться, что в тех случаях, когда новая СУ разрабатывается для уже давно функционирующей системы S , дли-

тельное время находящейся в эксплуатации, положение с априорной информацией лучше и построение модели проще. Опыт показывает, что это не так, и получение информационной модели и в этом случае весьма трудоемко. Таким образом, как для случая вновь проектируемой системы S , так и для уже функционирующей возникает проблема получения дополнительной информации для создания СУ. Единственным эффективным путем получения такой информации в настоящее время является машинное моделирование.

В том случае, когда СУ создана и функционирует вместе с системой S , управляя ею, существует необходимость в получении текущей информации, вызванная в основном двумя причинами. Во-первых, это потребность в совершенствовании СУ, а во-вторых, необходимость уточнения поведения системы и возникающих в ней ситуаций с целью компенсации изменений характеристик системы S как ОУ. Процессы, с которыми связана текущая информация первого вида, являются достаточно медленными и для управления ими необходима подсистема эволюционного управления, а процессы второго типа являются более быстрыми и для управления ими необходима подсистема оперативного управления в реальном масштабе времени (РМВ).

Следует подчеркнуть, что по темпу принятия решений и месту решения задач подсистемы эволюционного и оперативного управления существенно отличаются друг от друга. Так, например, процессы оперативного управления могут быть на несколько порядков более быстрыми по сравнению с процессами эволюционного управления.

Важнейшей задачей современной теории и практики управления является построение модели ОУ, т. е. формализация закономерностей функционирования объекта. На основе этой модели определяются структура, алгоритмы и параметры СУ, выбираются аппаратно-программные средства реализации системы. Одним из эффективных методов построения модели сложного объекта является идентификация.

Широкое развитие в настоящее время работ по формализации процессов и построению их моделей во многих областях исследований (технике, экономике, социологии и т. д.) преследуют две основные цели. Первая из них связана со значительным увеличением возможностей изучения на базе ЭВМ сложных процессов функционирования различных объектов при помощи метода моделирования, для чего необходимо математическое описание исследуемого процесса. Не меньшее значение в технических системах имеют модели, используемые для достижения второй цели, т. е. применяемые непосредственно в контуре управления объектами.

Эволюционные и десиженские модели. Невозможность ограничиться только одной универсальной моделью связана с тем, что, с одной стороны, перед этими моделями ставятся различные цели, а с другой стороны, они описывают процессы, протекающие в раз-

личных масштабах времени, причем степень полноты модели, ее соответствие реальному объекту зависят от целей, для которых эта модель используется. Модели первого типа имеют в основном гносеологический характер, от них требуется тесная связь с методами той конкретной области знаний, для которой они строятся. Модели такого типа являются достаточно «инерционными» в своем развитии, так как отражают эволюцию в конкретной области знаний. Такие модели будем называть *эволюционными*. Модели второго типа имеют информационный характер и должны соответствовать конкретным целям по принятию решений по управлению объектом, который они описывают. Такие модели будем называть *десиженсными*. Деление на гносеологические (эволюционные) и информационные (десиженсные) модели достаточно условно, но оно удобно для отражения целей моделирования.

В информационных моделях, используемых непосредственно для принятия решений в СУ, требование оперативности является одним из основных. Оно вызвано тем, что при каждом воздействии на ОУ необходимо в модели учесть действительные изменения, происшедшие в объекте, и внешние возмущения, на основе которых рассчитывается управление. Это требование оперативности, т. е. необходимость работы такой модели в РМВ, часто ведет к отказу от сложных и точных моделей, к разработке специальных, так называемых робастных, алгоритмов построения моделей, использование которых в СУ обычно ведет к поставленной цели [18, 21, 43, 54].

Появление идентификации в начале 60-х годов было связано с острой необходимостью разработки методов построения именно информационных моделей ОУ. Отсутствие таких моделей сдерживало процесс автоматизации этих объектов, использования ЭВМ в контуре управления. Объекты оказались неподготовленными к внедрению вычислительной техники из-за отсутствия их математического описания, их информационных моделей. Построение информационной модели методами идентификации должно быть направлено на ликвидацию этого разрыва и разработку методов оперативного получения модели ОУ. При этом методы идентификации должны предусматривать использование ЭВМ для решения задач построения информационной модели.

Элементы теории моделирования. Отсутствие формальных методов перехода от гносеологических моделей к информационным в современной теории управления не дает возможности получить по имеющейся информации адекватное описание, необходимое для создания СУ. Но учет сведений, содержащихся в гносеологических моделях, может значительно увеличить объем априорной информации о рассматриваемом ОУ. Поставив цель построения гносеологической модели процесса функционирования системы S для получения необходимой априорной информации для построения эффективной СУ и сузив класс объектов моделирования до конкретного, т. е. до поведения конкретной системы S , решим задачу

построения прикладной теории эволюционного и десиженсного моделирования, позволяющей эффективно (в реализационном аспекте) перейти от гносеологических («исследовательских») моделей к информационным («управленческим») моделям. Наиболее просто такой переход можно совершить, если оба этих класса моделей будут базироваться на единую концептуальную модель, использовать единую систему информации (базу знаний) и иметь единую критериальную систему. Рассмотрим сначала особенности гносеологических и информационных моделей.

Вопрос применимости некоторой математической модели к изучению рассматриваемого объекта не является чисто математическим вопросом и не может быть решен математическими методами. Только критерий практики позволяет сравнивать различные гипотетические модели и выбрать из них такую, которая является наиболее простой и в то же время правильно передает свойства изучаемого объекта, т. е. системы S .

Ориентируясь на общие вопросы методологии моделирования сложных технических систем, сформулируем требования к прикладной теории моделирования, а точнее — к элементам этой теории в ее приложении для решения конкретно поставленной задачи. Как уже отмечалось выше, эта задача ставится следующим образом. Необходимо сначала построить и реализовать на ЭВМ эволюционную модель процесса функционирования системы S , полученную в ходе стратегической идентификации ОУ, а затем на ее базе построить десиженсную модель, используемую для решения практических задач оперативного управления в адаптивной СУ сетью. Или, используя терминологию теории идентификации, необходимо построить конкретную дискретную адаптивную систему управления с идентификатором и предсказателем (комбинированную) в цепи обратной связи (ДАСК), т. е. реализовать сначала стратегический идентификатор, а затем на его базе тактический оперативный идентификатор и предсказатель, рассматривая в качестве ОУ не реальную систему S (ввиду ее отсутствия), а машинную модель процесса ее функционирования.

Таким образом, можно поставленную задачу трактовать и как задачу автоматизации исследования объекта (машинной модели M_m) для целей синтеза тактической и оперативной модели, используемой непосредственно в контуре управления системой S , а затем для проверки эффективности управления в целом.

Прежде чем переходить к изложению элементов теории моделирования процессов в системе S , дадим ряд определений. Напомним, что под моделированием будем понимать исследование объекта посредством изучения его модели, т. е. другого объекта, более удобного для этой цели. Под сложностью моделируемого объекта будем понимать фактически сложность сведений о нем (его описания), зависящую от целей моделирования и уровня, на котором выполняется описание. Таким образом, сложность возрастает не

только при введении в рассмотрение новых качеств, но и при переходе к более детальному описанию процесса функционирования объекта моделирования, т. е. системы S .

Задачу прикладной теории моделирования сформулируем, исходя из тех требований, которые будет предъявлять к ней пользователь (исследователь, разработчик системы S), проводящий эксперименты с процессами функционирования S и ее элементов для решения конкретной прикладной задачи. В таком контексте основной задачей при решении проблем управления является выбор моделей на уровне оперативного управления, сохраняющих при этом существенные для СУ черты S с учетом ограничений реализации в РМВ (особенно при оперативном управлении). В дальнейшем модель, практически реализуемую с учетом ограниченности ресурсов, будем называть *трактабельной*. Таким образом, помимо теоретических вопросов построения модели вообще будем рассматривать вопросы трактабельности модели, связанные с формальным представлением ее описания, его упрощением, проверкой адекватности упрощенной модели и т. д.

Тот факт, что моделируемая система S существует лишь как замысел разработчика, вносит в проблему разработки такой теории значительные трудности. В частности, не удастся непосредственно проверить адекватность модели процесса функционирования системы S с помощью реального объекта. Частично эта трудность устраняется путем проведения натуральных экспериментов с элементами S . Ряд существенных трудностей возникает из-за неполноты исходной информации об объекте моделирования.

Большой объем знаний о системах и их элементах, накопленный к настоящему времени, подлежащий объединению в рамках теории моделирования и несоизмеримый с познавательными возможностями одного исследователя, выдвигает необходимость организации и детализации таких знаний (теории) в систему, затрагивающую лишь существенно ограниченное число объектов при сохранении общности подхода. При этом развитие отдельных методов статистического моделирования, языков моделирования, теории планирования машинных экспериментов и т. д. оказывается недостаточным.

Создание прикладной теории, обеспечивающей конкретные потребности разработчика модели и охватывающей весь процесс моделирования в широком смысле этого слова, требует системного подхода и прежде всего установления основ теории: понятий об объекте, предмете, содержании, структуре и логике теории.

Объект прикладной теории моделирования. Объектом разрабатываемой прикладной теории является непосредственно процесс моделирования поведения системы S , т. е. процесс перехода от моделируемого объекта (системы S) сначала к статической модели S^s , используемой при стратегической идентификации, а затем и к динамической модели D^s , непосредственно используемой при оператив-

ном управлении с использованием методов и алгоритмов СУ. При этом ориентируются на критериальную систему K . Такой переход осуществляется через описание (концептуальную модель), фиксирующее сведения об объекте S в понятиях языка L (терминах типовых математических схем) [41, 54]. При выборе математической схемы моделирования M вводится также понятие среды S , позволяющее использовать информацию прикладного характера J о целях моделирования, законах функционирования системы S , имеющемся математическом аппарате и т. д. для исследования методов и алгоритмов управления системой A .

Таким образом, так как объектом данной прикладной теории моделирования является процесс моделирования, то возникает необходимость в построении и изучении «модели моделей», или репромодели RM (от англ. reproduce — воспроизводить, делать копию, порождать). Репромодель представляет собой упрощенный и наглядный прототип создаваемых моделей, используемых в СУ, и дает возможность эффективного приближения к таким моделям с максимальным использованием априорной и оперативной информации о поведении системы S , поступающей в процессе ее функци-



Рис. 9.1. Схема разработки модели системы S

онирования. Для решения поставленной задачи разработки модели для СУ схема репромоделей приведена на рис. 9.1. После того как сформулирована концептуальная модель S и введены понятия компонент сред S , основное содержание элементов прикладной теории моделирования для управления системой составят компоненты M , A , S^S и D^S (критерий K считается заданным), причем переход от M к S^S , следуя терминологии [29, 53], составит статику моделирования, а переход от M к множеству D^S с привлечением информации из компонент S^S и A составит динамику моделирования. Такое разделение на статику и динамику условно показано на рис. 9.1 пунктирной и сплошной линиями соответственно.

Движение в пространстве статических моделей процесса функционирования системы S^S назовем *эволюцией* (или эволюционным моделированием), а движение в пространстве динамических (активных) моделей D^S , используемых в контуре управления, — *самоорганизацией* (или моделированием с самоорганизацией). Важно отметить, что компоненты объекта теории L , C , E , M имеют исходное происхождение, базирующееся на эвристических представлениях, и могут при необходимости изменяться (развиваться) в интересах самой прикладной теории. Это существенно отличает прикладную теорию моделирования от естественно-научных теорий.

Предмет прикладной теории моделирования. Высказывания, составляющие любую теорию, формируются относительно предмета теории, а именно системы понятий, отображающих с той или иной степенью обобщения объект теории (репромодель). Таким образом, задание предмета прикладной теории моделирования процессов в системе S равносильно заданию репромоделей. Очевидно, что вообще различным репромоделям должны соответствовать различные аспекты теории. Применительно к СУ сужаем круг этих аспектов за счет конкретизации целей моделирования путем введения в репромодель компонент A , ограниченных методами и алгоритмами оперативного управления. Построение репромоделей по схеме, приведенной на рис. 9.1, позволяет использовать как информацию общего характера о процессах моделирования и управления J , так и конкретную информацию о методах и алгоритмах управления системой S^* с учетом выбранных критериев оценки эффективности K .

Содержание, структура и логика прикладной теории. Содержание прикладной теории моделирования охватывает две части: базис теории, включающей систему эвристических принципов, полученных при обобщении имеющегося опыта моделирования сложных объектов вообще, и тело теории, содержащее эвристические правила машинной реализации конкретных моделей процесса функционирования $S(S^S$ и $D^S)$, которые будут рассмотрены в § 10.1.

Предложения теории, относящиеся к компонентам M , A , S^S и D^S или возможным переходам между ними, содержат множество

условий, позволяющих точно их сформулировать лишь для простейших случаев. В пределе предложения сводятся к описанию фактов, относящихся к отдельным реализациям процесса моделирования, которые назовем *прецедентами* Pr . Отметим, что Pr составляют эмпирическую основу прикладной теории моделирования, а множество $\{Pr\}$, классифицированное по условиям, может рассматриваться как обобщенное предложение теории, содержащее весь зафиксированный в $\{Pr\}$ опыт моделирования сложных систем вообще.

Более определенные предложения теории могут быть получены на основе системного подхода с детализацией репромодели по этапам построения и реализации S^S и D^S , когда ставятся различные цели при моделировании процессов в системе S . В общем случае репромодель, т. е. ее базис, задается множеством принципов $\{Pr\}$, определяющих желаемые свойства моделей (S^S и D^S) и другие ограничения. Использование $\{Pr\}$ регламентируется предложениями теории, относящимися к ограниченному множеству обобщенных ситуаций. Поиск этих ситуаций в множестве известных $\{Pr\}$ позволяет накопить необходимые факты в количестве, достаточном для формулировки обобщенных предложений.

Говоря о прикладной теории моделирования с системных позиций, невозможно обойти ее реализационный аспект. В теории это отражено введением понятия трактруемости модели, т. е. ее реализуемости в рамках принятых ресурсных ограничений (например, на оперативную память и быстродействие ЭВМ). Особенно важна трактруемость десиженсных моделей, непосредственно используемых в СУ, так как часто от нее зависит эффективность конкретного метода и алгоритма управления (а иногда и возможность его использования вообще). Вопросы трактруемости модели ставятся во главу угла при проведении стратегического и тактического планирования машинных экспериментов (см. гл. 6). Поэтому не будем останавливаться на этих вопросах детально, отметим только, что трактруемость модели достигается выполнением набора практических правил реализации модели $\{Pr\}$, которые и составляют тело прикладной теории моделирования.

Таким образом, в конечном итоге множество прецедентов $\{Pr\}$ выражается через меньшее число эвристических принципов $\{Pr\}$ и практических правил реализации $\{Pr\}$ (базис и тело теории). Это позволяет считать репромодель и систему $[\{Pr\}, \{Pr\}]$ основой «системного» аспекта прикладной теории моделирования. При практическом применении неизбежно объединение «прецедентного» и «системного» аспектов теории моделирования на основе логического понятия дополнительности. В данном случае это способствует сужению общей проблемы моделирования за счет введения в прикладную теорию компоненты A . Для обеспечения возможности развития репромодель должна строиться как открытая система,

т. е. с соблюдением принципов архитектуры открытых систем, что нашло свое отражение при машинной реализации моделей [41, 54].

Относительно логики прикладной теории моделирования отметим, что она опирается на индуктивный подход, т. е. обобщение и классификацию множества прецедентов $\{Pr\}$, оставляя место для дедуктивного подхода в рамках конкретных математических схем M .

Вопросы практического воплощения прикладной теории моделирования непосредственно связаны с реализацией соответствующих инструментальных средств моделирования (см. гл. 5) и возможностью ее использования для решения задач моделирования конкретных систем (см. гл. 10).

9.2. МОДЕЛИ В АДАПТИВНЫХ СИСТЕМАХ УПРАВЛЕНИЯ

Одной из центральных проблем современной теории управления является проблема управления динамическими объектами в условиях неопределенности, т. е. проблема построения адаптивных СУ. Принцип работы этих систем основан на изменении параметров и структуры в результате наблюдения и обработки текущей информации так, чтобы адаптивная или обучающая система с течением времени улучшила свое функционирование, достигая в конечном итоге оптимального состояния. В адаптивных СУ недостаток априорной информации компенсируется благодаря целенаправленной обработке текущей информации. Рассмотрим возможность и особенности использования машинных моделей M_m для решения основных задач построения адаптивных СУ.

Адаптация в системах управления. Под *адаптацией* понимается процесс изменения структуры, алгоритмов и параметров системы S на основе информации, получаемой в процессе управления с целью достижения оптимального (в смысле принятого критерия) состояния или поведения системы при начальной неопределенности и изменяющихся условиях работы системы во взаимодействии с внешней средой E .

Адаптация использует обучение и самообучение для получения в условиях неопределенности информации о состояниях и характеристиках объекта, необходимой для оптимального управления. Обучение понимается как процесс выработки в некотором объекте тех или иных свойств его реакции на внешние воздействия путем многократных испытаний и корректировок. Самообучение отличается от обучения отсутствием внешней корректировки.

Характерная черта адаптации — текущее накопление информации о процессе функционирования системы S и внешней среды E и ее использование для улучшения избранного показателя качества. Процесс накопления информации связан с затратами времени,

что в итоге приводит к запаздыванию в получении системой управления информации, необходимой для принятия решений. Это существенно снижает эффективность работы систем управления в реальном масштабе времени. Поэтому актуальной является задача прогнозирования состояний (ситуаций) системы S и внешней среды E и характеристик (поведения) системы S для адаптивного управления. Такой прогноз может быть выполнен при использовании методов моделирования в системе управления в реальном масштабе времени (см. § 9.3).

Виды используемых моделей. Выделяются два направления в теории и практике построения адаптивных СУ — создание систем с эталонной моделью (АСЭМ) и с идентификацией объекта управления (АСИ). В АСИ сначала осуществляется идентификация объекта, а затем по оценкам его параметров определяются параметры управляющего устройства, а в АСЭМ осуществляется подстройка параметров управляющего устройства так, чтобы замкнутая система была близка к эталонной модели. Авторы считают, что дальнейшее развитие АСЭМ и АСИ пойдет по пути взаимного проникновения методов и результатов исследования, что позволит синтезировать алгоритмы, обладающие всеми достоинствами как того, так и другого направления. Широкое применение в СУ средств вычислительной техники вызвало особый интерес к дискретным адаптивным системам управления (ДАС), которым в последнее время посвящается большая часть публикаций по адаптивным системам [41, 43, 54].

Следует отметить, что выбор за классификационный признак наличия или отсутствия эталонной модели для современных ДАС не является, по сути дела, оправданным, так как эталонная модель в той или иной форме присутствует в любой ДАС. Сравнительно недавно предложена и развита более обоснованная классификация ДАС на прямые и непрямые и дана трактовка их общности, свойств и особенностей [37, 50]. Согласно этой классификации, все ДАС можно подразделить на два типа: *непрямые* ДАС, в которых параметры управляющего устройства определяются по оценкам параметров объекта с помощью некоторого вычислительного устройства, и *прямые* ДАС, в которых параметры управляющего устройства определяются непосредственно, без вычислительного устройства.

К *непрямым* ДАС относятся системы с идентификатором в контуре адаптации (ДАСИ), а к *прямым* — системы с предсказателем (ДАСП) в контуре. В соответствии с этой классификацией ДАС, используемые для управления процессами в таком сложном объекте, как информационная система S , можно отнести к *непрямым* комбинированным (ДАСК), так как в адаптивной системе управления S имеют место идентификатор и предсказатель, реализуемые с помощью вычислительных устройств, причем ком-

бинирование понимается как в смысле использования ДАСИ и ДАСП, так и в смысле использования принципов АСИ и АСЭМ.

Создание и развитие теории ДАС обусловлено прежде всего неполнотой априорной информации о процессе функционирования исследуемого объекта (в нашем случае ИС и ее элементов). Именно от объема априорной информации зависит и математическая постановка задачи, а часто этим определяется не только подход, но и метод ее решения. Исходя из того, что элементы ИС часто являются мало изученными объектами, т. е. практически отсутствуют априорные сведения о них, напрашивается вывод о необходимости построения непараметрических ДАС. Но для такой сложной системы, как ИС, следует отметить возникающие существенные трудности при использовании непараметрической адаптации для всей системы, т. е. при практическом рассмотрении ОУ как «черного ящика»: сложность методов и громоздкость алгоритмов адаптивного управления и, как следствие, их практическая нереализуемость с учетом ограничений вычислительных ресурсов ИС, а часто и необходимости управления в РМВ.

В ряде случаев более перспективен параметрический подход к решению проблемы адаптивного управления при максимальном использовании априорной информации об ОУ и процессе его функционирования. Поэтому применительно к проблеме построения ДАС сложными объектами в ИС можно сделать следующий вывод. Нельзя использовать один и тот же подход к решению задач адаптивного управления на различных уровнях. На каждом из уровней необходимо использовать те методы адаптации, которые позволяют достичь наиболее эффективного управления в каждом конкретном случае.

Идентификация процессов. Как уже отмечалось, одно из важнейших направлений в области идентификации и управления связано с дискретными АС, содержащими в контуре управления идентификатор, т. е. ДАСИ. Процесс идентификации, осуществляемый в ДАСИ, условно разделяется на два этапа, на каждом из которых информация для решения задачи идентификации поступает непосредственно с ОУ в виде реализаций входных и выходных переменных.

Первый этап связан с решением задачи идентификации в широком смысле, или задачи стратегической идентификации. Сюда относятся построение концептуальной модели, выбор информативных переменных, оценка степени стационарности объекта, выбор структуры и параметров модели, оценка точности и достоверности модели реальному объекту.

Второй этап предусматривает текущую идентификацию — уточнение модели в связи с текущими изменениями объекта и внешних воздействий; здесь обычно решаются задачи идентификации

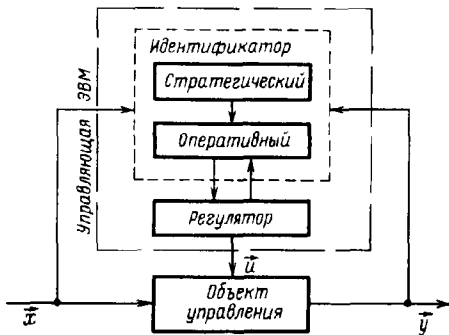


Рис. 9.2. Структурная блок-схема дискретной адаптивной системы с идентификатором

в узком смысле, т. е. задачи оценки поведения объекта или его состояний.

Структурная блок-схема классической ДАСИ приведена на рис. 9.2, где \vec{x} , \vec{y} , \vec{u} — векторы входов, выходов и управления. Стратегический идентификатор осуществляет решение задач идентификации в широком смысле вне контура управления, а оперативный идентификатор — в узком смысле и является составной частью замкнутого контура управления.

В целом ДАСИ обладают рядом важных практических достоинств: автоматизация идентификации, объединение процессов идентификации и управления, универсальность, высокая надежность. Для сложных объектов трудоемкость процесса идентификации соизмерима с трудоемкостью процесса проектирования системы. Объединение процессов идентификации и управления сокращает сроки создания и освоения системы в результате параллельного проведения работ и, кроме того, является, пожалуй, единственной возможностью оперативно компенсировать текущие изменения характеристик объекта и воздействий внешней среды в процессе функционирования. Из сказанного ясно, что одним из основных в ДАСИ является процесс идентификации. В ДАСИ реализуется принцип современной теории управления: «хорошая модель — залог успешного управления» [43, 54].

9.3. МОДЕЛИРОВАНИЕ В СИСТЕМАХ УПРАВЛЕНИЯ В РЕАЛЬНОМ МАСШТАБЕ ВРЕМЕНИ

С ускорением темпов развития экономики и интенсификации производственных процессов все шире внедряется автоматизация на предприятиях: от организационного управления цехами и участками до управления технологическими процессами выпуска различных изделий. Наиболее перспективным направлением является создание гибких автоматизированных производств и производственных систем, позволяющих на базе использования современных робототехнических комплексов, станков с числовым программным управлением, средств вычислительной техники оперативно переходить на выпуск новейших изделий, отслеживая динамику потребностей и конъюнктуру мирового рынка. Управление в таких гибких системах наиболее эффективно может быть реализовано на ба-

зе локальных сетей ЭВМ, обеспечивающих взаимодействие и координацию всех информационно-вычислительных ресурсов для управления отдельными агрегатами в системе и дающих возможность проводить обработку информации в реальном масштабе времени.

Особенности управления в реальном масштабе времени. Прогресс развития национальной экономики в настоящее время все теснее связывается с тем, насколько эффективно происходит накопление, обмен и выдача информации различным пользователям (администраторам, исследователям, работникам сферы обслуживания и т. д.). По сути дела, на базе современных средств вычислительной техники и техники связи создается настоящая «индустрия» производства и потребления информации, требующая больших информационно-вычислительных ресурсов и оперативного доступа к ним. Наиболее перспективно объединение всех информационно-вычислительных ресурсов с помощью цифровых сетей интегрального обслуживания, позволяющих в единой цифровой форме передавать различные виды информации (оперативные, диалоговые данные и файлы ЭВМ, речь, телевизионные сигналы и т. п.). Для эффективного удовлетворения требований различных пользователей к качеству и своевременности доставки информации управление сетями интегрального обслуживания должно быть реализовано в реальном масштабе времени.

Можно привести и другие примеры систем, управление которыми должно осуществляться в реальном масштабе времени. Все перечисленные системы объединяет то, что они относятся к классу больших систем (см. гл. 1), что усложняет решение задач управления ими. При разработке систем управления такими объектами обычно отсутствует априорная информация об условиях их работы. Это делает перспективным построение адаптивных систем управления (см. § 9.2).

В рассмотренных в предыдущих параграфах примерах метод моделирования применялся для целей исследования характеристик систем S во взаимодействии с внешней средой E , проектирования (синтеза) структуры, алгоритмов и параметров системы и т. п. Во всех этих случаях, как правило, отсутствовали жесткие ограничения на время между началом моделирования и получением результата, в качестве технических средств предполагалось использование высокопроизводительных ЭВМ и ГВК.

Прогнозирование и принятие решений. Основной целью моделирования является прогнозирование в широком смысле этого слова. Моделирование позволяет сделать вывод о принципиальной работоспособности объекта (системы S), оценить его потенциально возможные характеристики, установить зависимость характеристик от различных параметров и переменных, определить оптимальные зна-

чения параметров и т. п. Машинные модели M_m , используемые в качестве имитаторов и тренажеров, дают возможность предсказать поведение системы S в условиях взаимодействия с внешней средой E .

Использование метода моделирования для получения прогноза при принятии решений в системе управления в реальном масштабе времени выдвигает на первое место задачу выполнения ограничения на ресурс времени моделирования процесса функционирования системы S . Поэтому рассмотрим более подробно особенности прогнозирования на основе машинной модели M_x в реальном масштабе времени.

Для управления объектом может использоваться в системе либо информация о состояниях (ситуациях) системы S и внешней среды E , либо информация о выходных характеристиках (поведении) системы S во взаимодействии с внешней средой E . Это обстоятельство определяет и цели моделирования. В одном случае требуется оценить изменения состояний $z_k \in Z$, $k = \overline{1, n_z}$, за время прогнозирования τ_n (назовем такое моделирование *ситуационным*). В другом случае требуется оценить выходные характеристики $y_j \in Y$, $j = \overline{1, n_y}$, на интервале времени $(0, T)$ (назовем такое моделирование *бихевиоральным*).

Таким образом, цель ситуационного моделирования — получение прогноза вектора состояний $\vec{z}(t)$ (см. 2.3)), а цель бихевиорального моделирования — оценка вектора выходных характеристик $\vec{y}(t)$ [см. (2.2) и (2.5)]. Например, если в качестве концептуальной модели M_x процесса функционирования системы S используется Q -схема, то при ситуационном моделировании требуется прогнозировать такие состояния, как число заявок в накопителях, количество занятых каналов и т. д., а при бихевиоральном моделировании в этом случае необходимо оценивать такие характеристики, как вероятность потери заявки, среднее время задержки заявки в системе и т. д. Соответственно целям ситуационного и бихевиорального моделирования должен отличаться и подход к разработке и реализации моделирующих алгоритмов, хотя принципы их построения («принцип Δt » и «принцип δz ») сохраняются.

Другой особенностью моделирования для принятия решений по управлению объектом в реальном масштабе времени является существенная ограниченность вычислительных ресурсов, так как такие системы управления, а следовательно, и машинные модели M_m , реализуются, как правило, на базе мини- и микроЭВМ или специализированных микропроцессорных наборов, когда имеется ограничение по быстродействию и объему памяти. Это требует тщательного подхода к минимизации затрат ресурсов по моделированию в реальном масштабе времени [12, 29, 52].

Кроме того, следует учитывать, что достоверность и точность решения задачи моделирования (прогнозирование ситуаций или поведения) системы существенно зависят от количества реализаций N , которые затрачены на получение статистического прогноза (см. гл. 7). Таким образом, возникает проблема поиска компромисса между необходимостью увеличения затрат времени на моделирование, т. е. числа реализаций N [на интервале $(0, T)$] для повышения точности и достоверности результатов моделирования (прогнозирование), и необходимостью уменьшения затрат машинного времени из условий управления в реальном масштабе времени.

При использовании машинной модели M_m в контуре управления системой S в реальном масштабе времени возникает также проблема оперативного обновления информации как в базе данных об объекте, так и в базе данных об эксперименте, т. е. в данном случае о конкретном прогнозе.

Рассмотрим более подробно возможности построения моделирующих алгоритмов для ситуационных и бихевиоральных моделей. При ситуационном моделировании важно не потерять информацию о смене состояний системы S , так как от этого зависит эффективность управления. Поэтому построение детерминированных моделирующих алгоритмов, когда используется «принцип Δt », приводит либо к увеличению времени моделирования при уменьшении Δt , либо к снижению достоверности прогноза состояний при увеличении Δt . Это говорит в пользу использования стохастических алгоритмов, а именно тех вариантов, которые наиболее просто реализуются, т. е. асинхронных спорадических алгоритмов.

При бихевиоральном моделировании важно получить усредненную статистическую оценку характеристик системы S на интервале $(0, T)$. Поэтому при построении моделирующих алгоритмов важно при заданной точности и достоверности результатов моделирования выбрать наиболее просто реализуемый алгоритм, требующий минимальных затрат времени и оперативной памяти на его прогон. В этом случае эффективными могут оказаться как стохастические, так и детерминированные моделирующие алгоритмы. Выбор принципа построения моделирующего алгоритма для принятия решений в системе управления может быть проведен только с учетом особенностей конкретной S .

С точки зрения программирования моделей M_m при моделировании в реальном масштабе времени также имеется ряд особенностей. Это в первую очередь связано с отсутствием или невозможностью использования ЯОН и ЯИМ для программной реализации моделей исходя из возможностей программного обеспечения мини- и микро-ЭВМ и жестких ограничений на время счета по моделирующему алгоритму. В этом случае основное применение находят языки низкого уровня, что усложняет процесс разработки программного обеспечения моделирования в реальном масштабе времени, но

обычно позволяет получить достаточно эффективные рабочие программы моделирования. Для ускорения процесса разработки программного обеспечения моделирования в реальном масштабе времени и повышения его качества рационально разрабатывать соответствующие пакеты прикладных программ, которые с использованием ресурсов высокопроизводительных ЭВМ генерируют рабочие программы моделирования.

Таким образом, моделирование процесса функционирования систем для целей управления в реальном масштабе времени имеет ряд специфических особенностей, но методика моделирования и принципы реализации моделирующих алгоритмов сохраняются.

Контрольные вопросы

- 9.1. Что называется информационной моделью системы?
- 9.2. Каковы характерные черты эволюционных моделей систем?
- 9.3. Что называется трактабельностью модели системы?
- 9.4. В чем суть адаптации применительно к системам управления различными объектами?
- 9.5. Какова роль эталонной модели в контуре управления?
- 9.6. Какие модели используются для принятия решений?
- 9.7. Какие требования предъявляются к модели, реализуемой в реальном масштабе времени?

ГЛАВА 10

ИСПОЛЬЗОВАНИЕ МЕТОДА МОДЕЛИРОВАНИЯ ПРИ РАЗРАБОТКЕ АВТОМАТИЗИРОВАННЫХ СИСТЕМ

После изучения основ методологии моделирования, освоения технологии машинной имитации, рассмотрения вопросов реализации моделирующих алгоритмов и программ на ЭВМ необходимо, с одной стороны, подвести итоги, т. е. сформулировать, исходя из ранее рассмотренного, общие правила построения и способы реализации моделей систем, а с другой стороны, показать, как в целом работает инструмент моделирования в доступных приложениях. Поэтому в данной, заключительной, главе формулируются эвристические принципы и практические методы реализации машинных моделей, которые иллюстрируются приложениями к разработке организационно-производственных систем и информационно-вычислительных сетей, т. е. тех классов ИС, которые лежат в сфере будущей деятельности дипломированных специалистов.

10.1. ОБЩИЕ ПРАВИЛА ПОСТРОЕНИЯ И СПОСОБЫ РЕАЛИЗАЦИИ МОДЕЛЕЙ СИСТЕМ

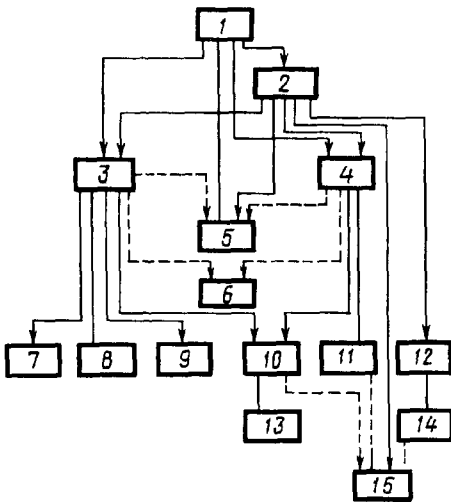
В настоящее время метод машинного моделирования широко применяется при разработке обеспечивающих и функциональных подсистем различных АСОИУ (интегрированных АСУ, автоматизированных систем научных исследований и комплексных испытаний, систем автоматизации проектирования и т. д.). При этом, как уже отмечалось, независимо от объекта можно выделить следующие основные этапы моделирования: 1) построение концептуальной модели системы S и ее формализация; 2) алгоритмизация модели системы S и ее машинная реализация; 3) получение результатов машинного моделирования и их интерпретация.

Методология машинного моделирования. На первом этапе моделирования формулируется модель, строится ее формальная схема и решается вопрос об эффективности и целесообразности моделирования системы S (об аналитическом расчете или имитационном моделировании) на вычислительной машине (на ЭВМ, АВМ или ГВК). На втором этапе математическая модель, сформулированная на первом этапе, воплощается в машинную, т. е. решается проблема алгоритмизации модели, ее рационального разбиения на блоки и организации интерфейса между ними, а также задача получения необходимой точности и достоверности результатов при проведении машинных экспериментов. На третьем этапе ЭВМ используется для имитации процесса функционирования системы S , для сбора

необходимой информации, ее статистической обработки и интерпретации результатов моделирования.

При этом следует учитывать, что на всех этапах моделирования переход от описания к машинной модели M_m , разбиение модели на части, выбор основных и второстепенных параметров, переменных и характеристик системы являются неформальными операциями, построенными на эвристических принципах, охватывающих как механизм принятия решений, так и проверку соответствия принятого решения действительности. Обобщая полученные результаты в области методологии машинного моделирования, можно условно разделить эвристические принципы моделирования на совокупность основных правил построения моделей систем и способов их машинной реализации, причем правила определяют общие свойства, которыми должна обладать построенная машинная модель, а способы реализации дают конкретные приемы получения нужных свойств модели системы. Следует отметить, что правила построения и способы их реализации образуют единую систему, так что обособленное их рассмотрение не дает полного представления о методологии машинного моделирования [29, 36, 37, 53].

Иерархическая структура взаимосвязи эвристических правил построения и практических способов реализации машинных моделей M_m может быть условно представлена в виде схемы (рис. 10.1), которая задает цепь неформальных действий, выполняемых при моделировании систем в широком смысле этого слова. На рисунке



приняты следующие обозначения: правила: 1 — составление точности и сложности модели; 2 — соразмерность погрешностей моделирования и описания; 3 — реализация блочного представления модели; 4 — специализация моделей для конкретных условий; 5 — достаточность набора элементов модели; 6 — наглядность модели для исследователя и пользователя; способы: 7 — минимальный обмен информацией между блоками; 8 — упрощение модели по критерию интерпретации; 9 — удаление блоков с модификацией критерия; 10 — замена зависимых воздействий независимыми; 11 — проверка точности на условных моделях; 12 —

Рис. 10.1 Схема взаимосвязи правил построения и способов реализации машинных моделей

проверка точности по сходимости результатов; 13 — выбор эквивалента входных блоков; 14 — сравнение моделей различной сложности; 15 — параллельное моделирование вариантов системы.

На схеме сплошными линиями показаны связи общих правил и способов с частными, пунктирными — возможность использования соответствующего правила или способа. Коротко рассмотрим основной смысл перечисленных правил и способов моделирования и их взаимосвязь.

Правила построения машинных моделей. Правило сопоставления точности и сложности модели (правило 1) характеризует компромисс между ожидаемой точностью и достоверностью результатов моделирования и сложностью модели системы S с точки зрения ее машинной реализации. Правило соразмерности погрешностей моделирования системы и ее описания (правило 2) представляет, по сути, «баланс точностей», определяемый соответствием систематической погрешности моделирования из-за неадекватности модели M_m описанию системы S с погрешностью в задании описания вследствие неопределенности исходных данных; взаимным соответствием точностей блоков модели; соответствием систематической погрешности моделирования на ЭВМ и случайной погрешности представления результатов моделирования.

Следует помнить, что сложность модели системы S характеризуется затратами времени на построение модели M_m , затратами машинного времени на ее реализацию и объемом памяти конкретной ЭВМ, используемой для моделирования, причем выигрыш в затратах машинного времени получают при сравнительной оценке вариантов разбиения модели M_m на блоки. Отсюда вытекает следующий способ реализации этих правил, а именно способ параллельного моделирования вариантов системы (способ 15), т. е. возможность параллельного моделирования конкурирующих вариантов исследуемой системы S с оценкой разностей соответствующих показателей качества функционирования.

Практическая реализация правил 1 и 2 возможна лишь при наличии гибкой системы, позволяющей создать достаточное разнообразие вариантов модели, т. е. необходимо выполнение правила достаточности набора элементов модели M_m (правило 5) — типовых процедур моделирования и оптимизации в математическом и программном обеспечении моделирования.

Построение моделей во многом — творческая задача, решаемая человеком, т. е. при ее решении должно быть соблюдено правило наглядности модели для исследователя (правило 6), выполнение которого дает возможность исследователю и пользователю (заказчику) оперировать с привычными представлениями об объекте моделирования, что позволяет избежать многих ошибок и упрощает трактовку полученных результатов. В частности, необходимость блочной конструкции модели M_m вызывается не только

особенностями ее машинной реализации, но и удобствами сохранения понятий, которыми привык оперировать пользователь.

Переходить от описания системы S к ее машинной модели M_m наиболее рационально путем построения блочной модели, т. е. необходимо выполнение правила реализации блочного представления модели (правило 3), в соответствии с которым надо находить блоки, удобные для автономного моделирования (на ЭВМ, АВМ и ГВК), и блоки, допускающие исследования натурными методами; принимать решение о существенности или несущественности каждого блока для задачи исследования характеристик данной системы S с целью сохранения структуры описания в пределах этого блока, замены ее упрощенным описанием или удаления блока из модели.

Способы реализации машинных моделей. Разбиение на блоки с точки зрения дальнейшей реализации модели целесообразно проводить, по возможности минимизируя число связей между блоками модели, т. е. отсюда вытекает способ минимального обмена информацией между блоками (способ 7).

Кроме того, при решении вопроса о допустимости удаления блоков из модели целесообразно пользоваться способом упрощения модели M_m по критериям интерпретации (способ 8), т. е. несущественными считаются те блоки, которые мало влияют на критерий интерпретации результатов моделирования и в силу этого могут быть удалены из модели, в том числе и в процессе моделирования системы. Способы удаления блоков различаются в зависимости от характера взаимодействия этих блоков с оставшейся частью системы. Удаляя оконечные блоки, составляющие описание взаимодействия системы S с внешней средой E , необходимо учесть это при формировании критерия интерпретации результатов моделирования, т. е. это соответствует способу удаления блоков с модификацией критерия (способ 9).

Рассмотрим теперь способ замены блока, осуществляющего воздействие на исследуемую часть системы S . Такой блок не является автономным и его нельзя заменить одним эквивалентным, не зависимым от исследуемой части системы. Но в ряде случаев удастся указать диапазон изменения переменных, т. е. функционирование исследуемой части системы можно изучать путем многократного моделирования (по числу воздействий) при различных значениях переменных внутри заданного интервала. Эти предположения реализуются способом замены зависимых воздействий независимыми (способ 10).

При реализации модели M_m системы S необходимо решить путем сопоставления вопрос о способе выбора эквивалента входных воздействий (способ 13): упрощение замкнутого контура, образуемого входным блоком и исследуемой частью системы без разрыва обратной связи; построение вероятностного эквивалента на основе предварительного его исследования (частичного моделирования);

замена входного блока наилучшим воздействием по отношению к исследуемой части системы.

До сих пор рассматривались только блоки, реализующие структурное разделение машинной модели на непересекающиеся части, но можно использовать и временное разделение на блоки (условные подмодели), которые отражают различные этапы или режимы функционирования системы S , т. е. в этом случае в них могут входить пересекающиеся части системы. В ряде случаев выделение условных подмоделей позволяет добиться упрощений при реализации машинной модели M_m , сузить разброс результатов моделирования и тем самым сократить требуемое количество прогонов. Обобщая схему условных подмоделей, можно сформулировать правило специализации для конкретных условий (правило 4), определяющее целесообразность использования набора частных условных подмоделей, предназначенных для анализа характеристик процесса функционирования системы S в конкретных условиях и дающих возможность судить о системе в целом по совокупности частных показателей, полученных на условных подмоделях, построенных с учетом особенностей планирования машинных экспериментов.

При этом специализация полной модели системы позволяет в отдельных случаях проверить точность ее упрощенного блочного представления, т. е. отсюда вытекает способ проверки точности на условных моделях (способ 11). Условные подмодели строятся независимо друг от друга, что позволяет ускорить исследование, выполняя параллельные машинные эксперименты со всеми подмоделями, например на нескольких ЭВМ.

Динамика моделирования системы S может быть определена как движение в некотором подпространстве моделей $\{M\}$. Причем при исследовании систем движение идет в сторону усложнения модели. Отсюда вытекает способ проверки точности по сходимости результатов (способ 12), т. е. проверки точности результатов моделирования, получаемых на моделях возрастающей сложности. Такой способ позволяет двигаться «снизу — вверх» в подпространстве моделей $\{M\}$ от упрощенной модели, заведомо реализуемой на ЭВМ, в сторону ее развития и усложнения в пределах ограничений вычислительных ресурсов. В таком движении в подпространстве моделей $\{M\}$ следует остановиться, когда различие моделей становится незначительным. Эти особенности и реализуются способом сравнения моделей с различной сложностью (способ 14).

Рассмотренные эвристические правила и способы моделирования задают общую схему построения и реализации модели системы S , но не конкретные решения для каждого этапа машинного моделирования. Даже при работе с конкретным программно-техническим обеспечением для исследования определенного класса систем, например в виде пакета прикладных программ моделирования, необходимо предварительно решить ряд задач формализации

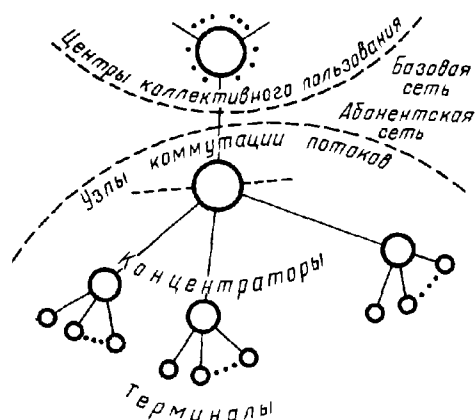
объекта моделирования, планирования машинных экспериментов и других, которые были рассмотрены в предшествующих главах.

10.2. МОДЕЛИРОВАНИЕ ПРИ РАЗРАБОТКЕ РАСПРЕДЕЛЕННЫХ АВТОМАТИЗИРОВАННЫХ СИСТЕМ И ИНФОРМАЦИОННЫХ СЕТЕЙ

Рассматривая АСОИУ с точки зрения технологии обработки информации и принятия решений, можно выделить функциональную схему управления, состоящую из обеспечивающих подсистем, находящихся во взаимосвязи как между собой, так и с внешней средой. При проектировании АСОИУ различных уровней, исходя из общности решаемых задач, принято выделять информационное, математическое, программное, техническое и организационное обеспечение [2, 25, 34, 35, 52].

Объект моделирования. Техническое обеспечение — одна из основных составных частей АСОИУ, той материально-технической базы, с помощью которой реализуются экономико-математические методы управления. Комплекс технических средств включает в себя разнообразные средства вычислительной техники, сбора и передачи информации, обеспечивающие своевременную и качественную переработку управляющей информации, причем территориальная удаленность объектов управления в АСОИУ требует применения средств передачи информации, основная задача которых — обмен информацией между местом ее возникновения и информационно-вычислительным центром с необходимой скоростью и достоверностью.

Наиболее перспективным направлением в области создания



технического обеспечения АСОИУ является построение информационно-вычислительных сетей, цифровых сетей интегрального обслуживания, позволяющих наиболее эффективно использовать ресурсы обработки и хранения информации [35, 38, 51, 54]. Структурная схема такой сети показана на рис. 10.2, где выделены уровни базовой (магистральной) сети, реализующей обмен информацией между центрами коллективного пользования, и терминальной (абонентской) сетью, обеспечи-

Рис. 10.2. Структурная схема информационно-вычислительной сети

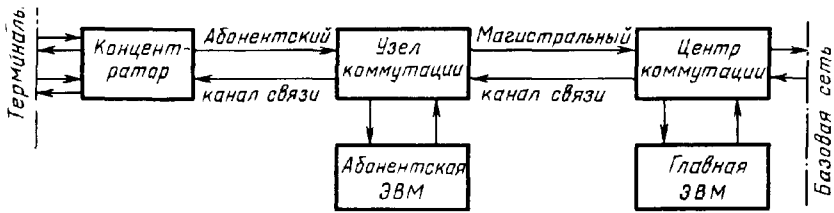


Рис. 10.3. Структурная схема взаимодействия терминальной и базовой сети

вающей обмен информацией между пользователями и ЭВМ. Основными структурными элементами сети являются: узлы (центры) коммутации потоков, осуществляющие все основные операции по управлению сетью, включая коммутацию и маршрутизацию потоков сообщений (пакетов); концентраторы, обеспечивающие сопряжение входных низкоскоростных каналов связи с выходным высокоскоростным каналом; терминалы, выполняющие функции организации доступа пользователя к ресурсам сети и функции по локальной обработке информации; каналы связи, реализующие обмен информацией между узлами сети (узлами коммутации, концентраторами, терминалами) с требуемым качеством.

Рассмотрим более подробно работу фрагмента такой информационно-вычислительной сети на уровне взаимодействия терминальной и базовой ее частей (рис. 10.3). Информация, требующая обработки, поступает с терминалов пользователей в виде сообщений длиной q бит с интенсивностью λ сообщений/с. Абонентская ЭВМ, подключенная к узлу коммутации, производительностью h бит/с обрабатывает поступающую от концентраторов информацию. Мультиплексные каналы ЭВМ обслуживают по k терминалов каждый, передавая данные к ЭВМ со скоростью B бит/с. При недостатке вычислительной мощности для обработки информации пользователей абонентская ЭВМ через узлы коммутации и магистральный канал связи с пропускной способностью C бит/с подключается посредством центра коммутации к ЭВМ верхнего уровня сети (главным ЭВМ), которые имеют суммарную производительность H бит/с при наличии n мультиплексных каналов. При этом предполагается, что процессы коммутации выполняются мгновенно. При проектировании АСОИУ необходимо оценить среднее время обработки информации T_0 и вероятность отказа в выполнении работ $P_{от}$ в случае работы только с абонентской ЭВМ, т. е. в автономном режиме, и в случае подключения к одной из ЭВМ сети коллективного пользования.

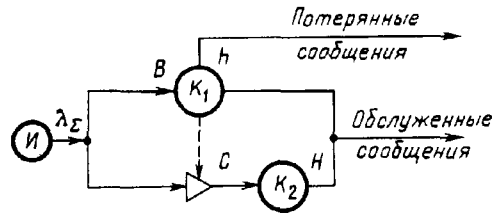


Рис. 10.4. Представление фрагмента сети в виде Q-схемы

Формализация процесса функционирования объекта моделирования. Процесс функционирования данного фрагмента информационно-вычислительной сети может быть представлен в виде Q -схемы, имеющей два параллельных канала обслуживания, а также связи, управляющей блокировкой. Структура такой Q -схемы, формализующей процесс работы фрагмента описанной сети, представлена на рис. 10.4. Здесь И — источник; K_1 и K_2 — каналы обслуживания. В качестве выходного потока источника (И) рассматривается суммарный поток сообщений от терминалов, т. е. поток на выходе концентратора. Выходной поток Q -схемы составят обслуженные сообщения при работе каналов K_1 и K_2 и потерянные сообщения при отключении (блокировке) канала K_2 . В такой постановке решение этой задачи аналитическим методом (в явном виде) с использованием теории массового обслуживания не представляется возможным из-за стохастического характера работы механизма блокировок, поэтому для оценки интересующих характеристик воспользуемся методом имитационного моделирования.

В этом случае можно записать: эндогенные переменные: T_0 — среднее время обслуживания сообщений; $P_{от}$ — вероятность отказа в обслуживании; экзогенные переменные: $\lambda_{\Sigma} = knq\lambda$ — интенсивность входного потока сообщений; h — производительность абонентской ЭВМ; H — суммарная производительность главных ЭВМ сети; B — пропускная способность селекторных каналов ЭВМ; C — пропускная способность магистрального канала связи; уравнения модели: а) при блокированном канале

$T_0 = 2B + q/h$, $P_{от} = (\lambda_{\Sigma} - h)/h$

при $\lambda_{\Sigma} > h$,

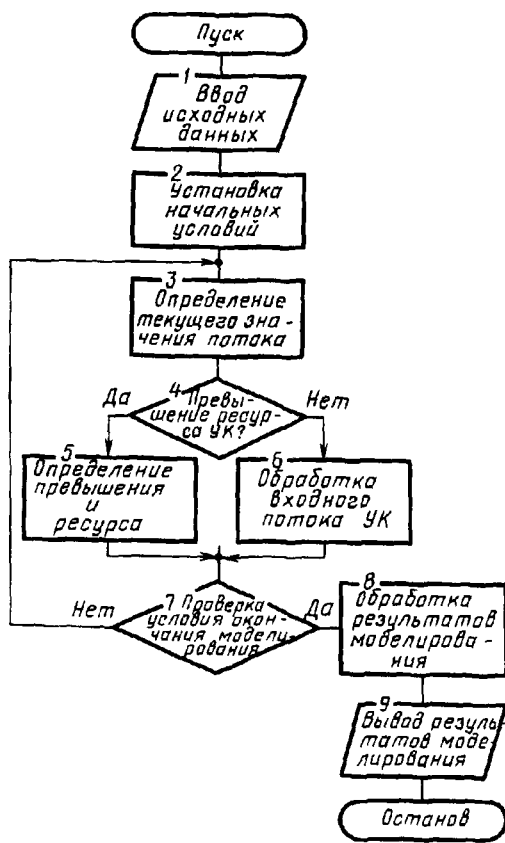


Рис. 10.5. Укрупненная схема моделирующего алгоритма фрагмента сети

б) при работе каналов K_1 и K_2

$$T_0 = 2q/B + q/h + \\ + h(2q/C + q/H)/(\lambda_{\Sigma} - L), P_{от} = 0.$$

Моделирующий алгоритм. Укрупненная схема моделирующего алгоритма процесса функционирования фрагмента информационно-вычислительной сети представлена на рис. 10.5.

Примеры результатов моделирования в виде зависимости $P_{от} = f(kn)$ приведены на рис. 10.6.

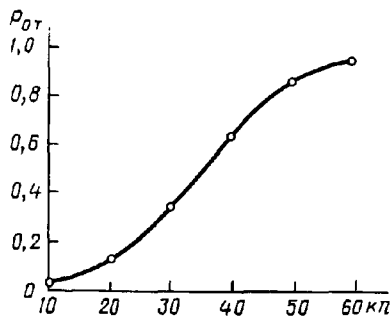


Рис. 10.6. Результат моделирования фрагмента сети

10.3. МОДЕЛИРОВАНИЕ ПРИ РАЗРАБОТКЕ ОРГАНИЗАЦИОННЫХ И ПРОИЗВОДСТВЕННЫХ СИСТЕМ

Эффективность работы АСОИУ существенно зависит от качества моделей, на базе которых реализуется процесс управления. В силу сложности объекта управления проводится его декомпозиция на отдельные части, т. е. выделяются функциональные подсистемы, включая такие, как технико-экономическое планирование, техническая подготовка производства, оперативное управление производством и т. д. [25, 29, 34, 52].

Объект моделирования. Рассмотрим одну из функциональных задач подсистемы оперативного управления основным производством. В качестве примера такой задачи можно привести задачу «Расчет плана сдачи и получения деталей в натуральном выражении».

Решение этой задачи на ЭВМ полностью позволяет автоматизировать получение информации по расчету планов сдачи и деталей в натуральном выражении по цехам на год с разбивкой по кварталам и месяцам на предприятиях с поддетальной системой планирования. Планы сдачи и получения деталей рассчитываются для всех цехов-изготовителей и цехов-потребителей предприятия согласно технологическому маршруту. Основу этих планов составляет развернутый план потребности в деталях на товарный выпуск продукции, скорректированный с учетом величины плана изменения заказов и компенсации неизбежных внутрипроизводственных потерь.

На предприятиях с дискретным характером производства (например, крупносерийного и массового приборостроения и машиностроения) процесс движения изделий (деталей, узлов) состоит из заготовительной, механообрабатывающей и сборочной стадий (рис. 10.7). Обычно стадии заготовки и сборки представляют собой детер-

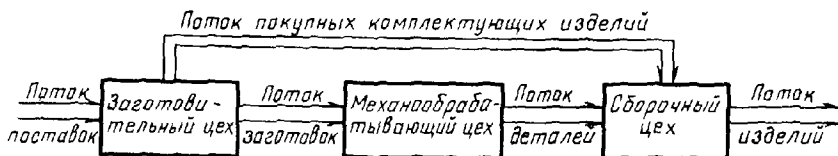


Рис. 10.7. Структурная схема движения изделий в процессе дискретного производства

минированные процессы, а механообрабатывающее производство имеет стохастическую природу. Фактически механообрабатывающие подразделения производства представляют собой некоторую кибернетическую систему типа «черного ящика», на вход которого поступает поток заготовок, а на выходе имеется поток партий готовых деталей. Количество стандартных деталей в партии и время выпуска — случайные величины. Для сглаживания стохастичности и придания потоку партий деталей, поступающих на сборку, ритмичного характера заранее до начала планового периода устанавливаются заделы деталей.

Под *заделами (запасами)* понимаются различного рода заготовки, покупные комплектующие изделия, готовые детали и т. д., находящиеся на разных стадиях производственного процесса в предназначенные для обеспечения бесперебойного хода работы на различных его стадиях. Обычно выделяют следующие заделы: оборотные, представляющие собой запасы деталей, комплексующих изделий и т. п., которые возникают из-за неполной согласованности времени выполнения работы на отдельных линиях или рабочих местах; страховые (резервные), представляющие собой запасы, предназначенные для локализации непредвиденных перебоев и неполадок с тем, чтобы не дать им распространиться далее по направлению технологического маршрута.

Для определения оптимального размера задела деталей в функциональной подсистеме оперативного управления основным производством АСУ предприятием решается соответствующая задача, в основе которой лежит модель, построенная на базе аналитического или имитационного подхода.

Фрагмент производственного процесса, показывающего взаимодействие цеха 1 (механообрабатывающего) и цеха 2 (сборочного) через оборотный и страховой заделы, представлен в виде структурной схемы (рис. 10.8). В процессе производства возможны следующие ситуации: а) нормальная, когда детали из цеха 1 поступают в оборотный задел (связь 1), а из оборотного задела — на сборку (связь 3); в страховом заделе имеется полный запас деталей; б) аварийная, когда детали из цеха 1 не поступают, оборотный задел израсходован, а сборка в цехе 2 обеспечивается только за счет страхового задела (связь 4); в) простойная, когда нет потока деталей

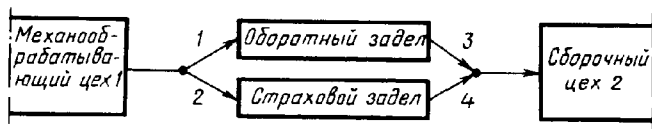


Рис. 10.8. Структурная схема взаимодействия механообрабатывающего и сборочного цехов

из цеха 1, а запасы деталей в оборотном и страховом заделах израсходованы, т. е. цех 2 простаивает; г) переходная, когда в оборотном заделе имеются детали и он пополняется из цеха 1 (связь 1), детали поступают на сборку (связь 3), а кроме того, идет комплектование страхового задела (связь 2).

Формализация процесса функционирования объекта моделирования. Процесс поступления и потребления деталей в цехе 2 можно формализовать в виде Q -схемы (рис. 10.9). Здесь И — источник; К — канал; Н — накопитель.

Таким образом, процесс заготовки изделий для сборки (деталей, покупных комплектующих изделий) можно представить в виде некоего источника (И), выдающего детерминированный поток покупных комплектующих изделий и стохастический поток партий деталей для сборки. В дальнейшем будем предполагать, что детерминированный поток покупных комплектующих изделий отсутствует, так как в общем случае он может быть рассмотрен как частный случай стохастического потока. Тогда на выходе И будет иметь место поток деталей на обработку, который описывается как количество единиц деталей q в партии, каждая из которых поступает через Q часов.

Процесс обработки в механообрабатывающем цехе 1 при наличии оборотного и страхового заделов можно представить в виде канала K_1 с временем обработки $Q_{об} = \text{const}$ и временем межоперационного пролеживания $Q_{мп} = \text{var}$, т. е. время обслуживания каналом K_1 составит $Q_1 = Q_{об} + Q_{мп}$, а также двух накопителей: H_1 и H_2 , причем в результате потерь от брака q_6 количество выпускаемых деталей q_1 является случайным.

Детали после обработки поступают в накопитель H_1 , емкость которого L_1 соответствует номинальному значению деталей в оборотном заделе $L_1 = Z_{об}$. При достижении номинального значения L_1 детали поступают в накопитель H_2 , соответствующий страховому заделу (если он нуждается в пополнении). При заполнении накопителя H_2 , т. е. достижении величины L_2 , клапан на его входе блокируется, причем начальное значение L_1 в накопителе H_1 на планируемый период равно величине одной партии деталей, выпускаемых из канала K_1 , а необходимое значение страхового задела $L_2 = Z_{ст}$ должно быть оценено в результате решения задачи.

Процесс сборки изделий в сборочном цехе 2 можно представить в виде канала K_2 , потребляющего $q_2 = \text{const}$ деталей через

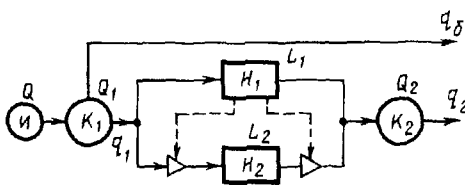


Рис. 10.9. Представление фрагмента производства в виде Q -схемы

интервалы времени $Q_2 = \text{const}$. При нехватке деталей в оборотном заделе (накопитель H_1) потребность для сборки восполняется из страхового задела (накопитель H_2) путем разблокировки соответствующего клапана на выходе H_2 .

При недостатке деталей в оборотном и страховом заделах сборочный участок, т. е. канал K_2 , простаивает до момента поступления из канала K_1 необходимого числа деталей.

Для возможности выбора необходимого значения оборотного задела L_1 определим вероятность простоя сборочного участка (канала K_2) как функцию L_2 , т. е. вероятность $P_{пр}$.

Таким образом, данная модель является двухфазной Q -схемой с двумя параллельными очередями во второй фазе обслуживания и с наличием блокировок. При этом потоки поступления и обслуживания заявок имеют как детерминированный, так и стохастический характер. Учитывая сказанное, получение вероятности $P_{пр}$ аналитическим методом в явном виде невозможно, поэтому воспользуемся методом имитационного моделирования на ЭВМ.

Для рассматриваемой модели производственного процесса в виде Q -схемы имеют место два выходных потока: поток обработанных изделий q_2 и поток бракованных деталей q_6 .

Представим переменные и уравнения модели в следующем виде: эндогенная переменная: $P_{пр}$ — вероятность простоя цеха 2; экзогенные переменные: Q — интервал времени между запуском партий деталей в цех 1; q — размер партий деталей, поступающих на обработку в цех 1; $Q_1 = Q_{об} + Q_{м.п}$ — интервал вре-

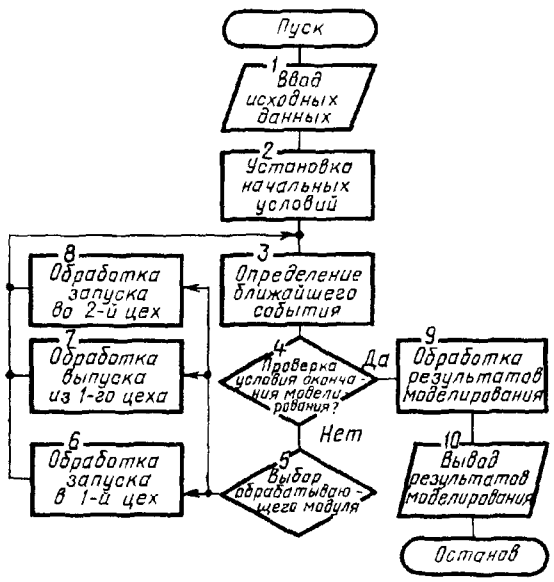


Рис. 10.10. Укрупненная схема моделирующего алгоритма фрагмента производства

мени между моментами выпуска деталей из цеха 1, где $Q_{об}$ — постоянное время обработки; $Q_{м.л}$ — время межоперационного пролеживания; $q_1 = q - q_6$ — число выпускаемых деталей цехом 1, где $q_6 = qF_6$ — число бракованных деталей; F_6 — доля бракованных деталей, которая считается равномерно распределенной в интервале $(0, F_6)$; Q_2 — интервал времени между запуском деталей в цех 2; q_2 — количество деталей в партии для цеха 2; $Z_{ст}$ — страховой задел, значение которого на начало года $Z_{ст0}$; $Z_{об}$ — оборотный задел, значение которого на начало года $Z_{об0}$; уравнение модели

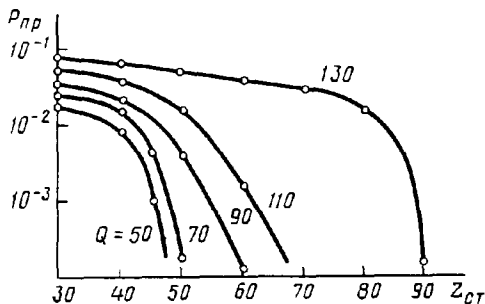


Рис. 10.11. Результат моделирования фрагмента производства

где Φ_T — годовой фонд времени; $T_{пр}$ — время простоя цеха 2 из-за отсутствия деталей. За интервал моделирования $(0, T)$ примем годовой фонд времени $T = \Phi_T = 4080$ ч.

$$P_{пр} = T_{пр} / \Phi_T,$$

Моделирующий алгоритм. Укрупненная схема моделирующего алгоритма фрагмента производства на уровне решения задачи определения страхового задела, т. е. взаимодействия механообрабатывающего (цех 1) и сборочного (цех 2) цехов, представлена на рис. 10.10.

Результаты моделирования в виде зависимости $P_{пр} = f(Z_{ст})$ для различных значений запуска приведены на рис. 10.11.

Контрольные вопросы

Контрольные вопросы

- 10.1. Какие основные этапы моделирования системы можно выделить?
- 10.2. Что представляют собой общие правила построения и способы реализации моделей систем?
- 10.3. Как осуществляется переход от концептуальной к машинной модели системы?
- 10.4. Какие типовые математические схемы использованы для формализации объектов моделирования в данной главе?
- 10.5. Какие инструментальные средства могут быть выбраны для реализации моделей объектов информационных систем, рассмотренных в данной главе?

ЗАКЛЮЧЕНИЕ

ПЕРСПЕКТИВЫ ИСПОЛЬЗОВАНИЯ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ В ИНФОРМАЦИОННОМ ОБЩЕСТВЕ

Процесс проектирования информационных систем, реализующих новую информационную технологию, непрерывно совершенствуется. В центре внимания инженеров-системотехников оказываются все более сложные системы, что затрудняет использование физических моделей и повышает значимость математических моделей и машинного моделирования систем. Машинное моделирование стало эффективным инструментом исследования и проектирования сложных систем. Актуальность математических моделей непрерывно возрастает из-за их гибкости, адекватности реальным процессам, высокой стоимости реализации на базе современных ЭВМ. Все большие возможности предоставляются пользователю, т. е. специалисту по моделированию систем средствами вычислительной техники. Особенно эффективно применение моделирования на ранних этапах проектирования автоматизированных систем, когда цена ошибочных решений наиболее значительна.

Современные вычислительные средства позволили существенно увеличить сложность используемых моделей при изучении систем, появилась возможность построения комбинированных, аналитико-информационных моделей, учитывающих все многообразие факторов, имеющих место в реальных системах, т. е. использования моделей, более адекватных исследуемым явлениям. Поэтому в учебнике особое внимание уделено методологическим аспектам моделирования, типовым математическим схемам, методам и средствам их реализации в процессе моделирования на современных ЭВМ.

В целом в учебнике сделана попытка системного подхода к изложению научных основ моделирования систем. Реализация возможностей машинного моделирования освещается авторами в учебном пособии «Моделирование систем: Практикум», которое издается параллельно с данным учебником издательством «Высшая школа».

Перспективным и значимым для теории и практики системного моделирования является дальнейшее развитие научных основ моделирования с ориентацией на новые информационные технологии в научных исследованиях, проектировании, управлении и обучении.

Одной из основных в настоящее и ближайшее время будет проблема информатизации образования в условиях перехода к информационному обществу. В конце текущего столетия, по определению Н. Винера [35], человечество переходит из века энергетики в век информатики, общество из индустриального превращается в информационное. В этих условиях информатизацию можно рассматри-

вать как процесс перехода от индустриального общества к информационному. Начиная с 80-х годов сумма знаний в обществе возрастает вдвое за каждые два года. Происходит перераспределение трудовых ресурсов из материального производства в сферу информации. В ряде стран суммарные расходы на компьютерную технику, телекоммуникации, электронику превышают расходы на энергетику, т. е. в силу уверенно вступает век информатизации.

Для информационного общества характерно полное удовлетворение информационных потребностей при завершении формирования единой информационной среды, которая определяет и информационную культуру общества и каждого человека. Информационная культура не ограничивается системой знаний в области информационных процессов и должна включать активно-преобразовательный аспект отношения к миру. В создании информационной культуры лидирующая роль принадлежит образованию.

Достижениями информатики, в которой, как в научном направлении, можно выделить 3 уровня: физический — программно-аппаратные средства вычислительной техники и техники связи, т. е. средства телематики (телекоммуникаций и информатики); логический — информационные технологии; прикладной — пользовательские информационные системы. Исходя из этого, можно провести анализ мирового состояния и тенденций развития процесса информатизации и место компьютерного (машинного) моделирования в этом многогранном процессе. Моделирование вообще и компьютерное имитационное моделирование, в частности, получает применение в таких информационных технологиях как:

1. Компьютерные программы, включающие в себя электронные учебники, тренажеры, тьюторы, лабораторные практикумы, тестовые системы, деловые игры, использующие машинные имитаторы.

2. Системы на базе мультимедиа технологий, построенные с использованием персональных ЭВМ, видеотехники, накопителей на оптических дисках, включая системы виртуальной реальности.

3. Интеллектуальные и обучающие экспертные системы, широко использующие имитационный подход в различных предметных областях.

4. Распределение базы данных по отраслям знаний, электронные библиотеки, распределенные и централизованные издательские системы.

5. Средства телекоммуникации, включающие в себя электронную почту, телеконференции, локальные и региональные сети связи, цифровые сети интегрального обслуживания и т. д. с использованием новых интеллектуальных сетевых технологий.

6. Геоинформационные системы, базирующиеся на технологии, объединяющей компьютерную картографию и системы управления базами данных, и реализующие технологию создания многослойной электронной карты, опорный слой которой описывает

географию территории, а каждый из остальных слоев — один из аспектов состояния территории.

7. Технологии защиты информации, включая защиту от несанкционированного доступа в ПЭВМ, защиту от перехвата в сетях и т. п.

Важно отметить, что, говоря о совокупности средств информатизации, следует иметь в виду не только средства вычислительной техники и некоторую «сумму информационных технологий», но также и сумму общественных знаний и умений по использованию указанных средств, которая может быть определена как уровень общественного (или организационного) обучения. Очевидно, что ни одна предметная область, в том числе и машинное моделирование, не может перешагнуть через некоторые объективные стадии такого общественного обучения. Этот факт впервые был облечен в форму модели стадий роста Р. Ноланом в 1979 г. Эта модель показывает, как изменяются расходы на информатизацию, которые определяют уровень организационного обучения, в зависимости от степени проникновения информационной технологии в деятельность организации. Таким образом, наряду с психологической необходимо обеспечить и профессиональную подготовленность пользователей. Особенно это актуально в сфере использования ЭВМ для целей имитации широкого класса систем.

Таким образом, информатизация является объективной реальностью. Возникает вопрос, не может ли информатизация предотвратить глобальный кризис цивилизации и явиться основным механизмом свершения образовательной революции? Считается возможным становление новой системы непрерывного открытого гибкого дистанционного образования на базе перспективных информационных технологий. Это возможно, если вложить в информационные технологии необходимое гуманистическое содержание. В этом смысле цель информатизации общества — создание гибридного интегрального интеллекта всей цивилизации, способного предвидеть и управлять развитием человечества.

В настоящее время в образовании создано и внедрено достаточно большое количество программных и технических разработок, реализующих отдельные информационные технологии. Но при этом используются различные методические подходы, несовместимые технические и программные средства, что затрудняет тиражирование, становится преградой на пути общения с информационными ресурсами и компьютерной техникой, приводит к распылению сил и средств. Наряду с этим различный подход к информатизации на школьном и вузовском уровнях вызывает большие трудности у учащихся при переходе с одного уровня обучения на другой, приводит к необходимости расходования учебного времени на освоение элементарных основ современных компьютерных технологий. Выход может быть найден путем создания системы непрерывного образования «школа — вуз».

Переход к информационному обществу предполагает формирование информационной культуры, как свода правил поведения в информационном обществе, в коммуникационной среде, человеко-машинных системах, вписывающихся в мировую гуманистическую культуру человечества. Вхождение пользователя в мировую сеть (например, Internet) позволяет получать огромные потоки информации, в том числе и для целей моделирования. Например, имеются базы данных по различным версиям языка моделирования GPSS/PC, доступные пользователям Internet. Информационное пространство Российского образования должно отвечать национальным интересам и базироваться на традициях отечественной культуры. Проблема развития интеллекта учащихся может быть решена не только средствами информатики. Проблемы науки об образовании должны решаться с учетом перехода в информационное общество, в котором будет сформирована «инфоносфера личности». Проблема развития интеллектуальных способностей личности смыкается уже сейчас с проблемой информатизацией образования.

Информатизация должна предусматривать эволюционное развитие методологии образования. Не отрицание, а последовательное эволюционное развитие сложившейся методологии образования за счет использования явных преимуществ компьютерных технологий, включая и имитационное моделирование (наглядность, возможность использования различных форм представления информации — звук, изображение, удаленный доступ, обработка и хранение больших объемов информации), должно стать основой опережающего развития образования. Информатизация может внести существенные изменения в структуру и организацию учебного процесса. Необходимо управлять ходом информатизации образования. И опять здесь существенные результаты могут быть получены специалистами по системному моделированию.

Использование персонального компьютера открывает принципиально новые возможности в организации изучения физических явлений с помощью машинной имитации. При традиционном подходе физическое понимание по существу вырабатывалось у учащихся (школьников, студентов и др.) в процессе выполнения ими необходимых математических преобразований и вычислений, что и представляет собой наибольшую трудность для указанной категории учащихся. Благодаря специально разработанным программам имитации физических процессов в реальных системах такую работу берет на себя компьютер, и на долю учащихся остается не само добывание результата, а выявление и усвоение его физических причин. Другими словами, теперь доминирует роль идей и качественных методов исследования, которые связаны с гораздо большей наглядностью и образностью мышления. При этом заметно возрастает эмоциональный фактор, и предмет изучения становится не только доступным, но и интересным.

СПИСОК ЛИТЕРАТУРЫ

1. Абдеев П. Ф. Философия информационной цивилизации. — М.: ВЛАДОС, 1994.
2. Арсеньев Б. П., Яковлев С. А. Интеграция распределенных баз данных. — СПб.: Лань, 2000.
3. Борн Г. Форматы данных. — Киев: торгово-издательское бюро ВНУ, 1995.
4. Бусленко Н. П. Моделирование сложных систем. — М.: Наука, 1988.
5. Веников В. А., Веников Г. В. Теория подобия и моделирования. — М.: Высшая школа, 1984.
6. Гнеденко Б. Д., Коваленко И. Н. Введение в теорию массового обслуживания. — М.: Наука, 1987.
7. Головин Ю. А., Яковлев С. А. Применение языков моделирования в обучении методам программной имитации сложных систем // Тез. докл. 6-й Междунар. конф. «Региональная информатика-98»; Ч. 1. — СПб, 1998.
8. Громов Г. Р. Очерки информационной технологии. — М.: Инфоарт., 1992.
9. Дегтярев Ю. И. Исследование операций. — М.: Высшая школа, 1986.
10. Ермаков С. М., Мелос В. Б. Математический эксперимент с моделями сложных стохастических систем. — СПб.: Изд. ГУ, 1993.
11. Имитационное моделирование производственных систем/ Под ред. А. А. Вавилова. — М.: Машиностроение; Берлин: Техник, 1983.
12. Инструментальные средства персональных ЭВМ. В 10 кн. — М.: Высшая школа, 1993.
13. Калашиников В. В., Рачев С. Т. Математические методы построения стохастических моделей обслуживания. — М.: Наука, 1988.
14. Калининченко Л. А., Рыжкин В. М. Машины баз данных и знаний — М.: Наука, 1990.
15. Калянов Г. Н. CASE структурный системный анализ. — М.: Лори, 1996.
16. Кандрашина Е. Ю., Литвишцева Л. В., Поспелов Д. А. Представление знаний о времени и пространстве в интеллектуальных системах, — М.: Наука, 1989.
17. Квидлер Е. Языки моделирования. — М.: Энергия, 1985.
18. Клейнен Дж. Статистические методы в имитационном моделировании. — М.: Статистика, 1978.
19. Кривулин Н. К. Оптимизация сложных систем при имитационном моделировании // Вестник Ленингр. Ун-та. 1990. № 8.
20. Кулачев А. П. Компьютерный контроль процессов и анализ сигналов. — М.: Информатика и компьютеры, 1999.
21. Линник И. Ю. Улучшение скорости сходимости метода Монте-Карло в некоторых задачах теории массового обслуживания // Кибернетика. № 5. 1978.
22. Математическая теория планирования эксперимента / Под ред. С. М. Ермакова. — М.: Наука, 1983.
23. Марк Д. А., Мак-Гоуэн К. SADT. — Методология структурного анализа и проектирования — М.: Метатехнология, 1993.
24. Моисеев Н. Н. Математические задачи системного анализа. — М.: Наука, 1981.
25. Мухин О. И. Компьютерная инструментальная среда. — Пермь: ПГТУ, 1991.
26. Назаров С. В. Операционные системы специализированных вычислительных комплексов: теория построения и системного проектирования. — М.: Машиностроение, 1989.
27. Николис Г., Пригожин И. Познание сложного. Введение — М.: Мир, 1990.

28. Питерсон Дж. Теория сетей Петри и моделирование систем. — М.: Мир, 1984.
29. Поляк Ю. Г., Филлимонов В. А. Статистическое машинное моделирование средств связи. — М.: Радио и связь, 1988.
30. Применение микропроцессорных средств в системах передачи информации / Под ред. Б. Я. Советова. — М.: Высшая школа, 1987.
31. Прицкер А. Введение в имитационное моделирование и язык СЛАМП. — М.: Мир, 1987.
32. Проектирование и программная реализация экспертных систем на персональных ЭВМ / Под ред. Д. Фохта. — М.: Финансы и статистика, 1990.
33. Романцев В. В., Яковлев С. А. Моделирование систем массового обслуживания. — СПб.: Поликом, 1995.
34. Системное обеспечение пакетов прикладных программ / Под ред. А. А. Самарского. — М.: Наука, 1990.
35. Советов Б. Я. Информационная технология. — М.: Высшая школа, 1994.
36. Советов Б. Я., Яковлев С. А. Моделирование систем. — М.: Высшая школа, 1985.
37. Советов Б. Я., Яковлев С. А. Моделирование систем (2-е изд.). — М.: Высшая школа, 1998.
38. Советов Б. Я., Яковлев С. А. Моделирование систем: Курсовое проектирование. — М.: Высшая школа, 1988.
39. Советов Б. Я., Яковлев С. А. Моделирование систем: Лабораторный практикум. — М.: Высшая школа, 1989.
40. Советов Б. Я., Яковлев С. А. Моделирование систем: Практикум. — М.: Высшая школа, 1999.
41. Советов Б. Я., Яковлев С. А. Построение сетей интегрального обслуживания. — Л.: Машиностроение, 1990.
42. Философские основы моделирования сложных систем управления / Андриющенко М. Н., Советов Б. Я., Яковлев С. А. и др. // Системный подход в технических науках (Методологические основы): Сб. науч. тр. — Л.: Изд АН СССР, 1989.
43. Технология системного моделирования / Под ред. С. В. Емельянова. — М.: Машиностроение; Берлин: Техник, 1989.
44. Форрестер Дж. Основы кибернетики предприятия. — М.: Прогресс, 1981.
45. Хакен Г. Синергетика. Иерархии неустойчивостей в самоорганизующихся системах и устройствах. — М.: Мир, 1985.
46. Шеннон Р. Имитационное моделирование систем. Искусство и наука. — М.: Мир, 1978.
47. Шрайбер Т. Дж. Моделирование на GPSS. — М.: Машиностроение, 1980.
48. Шлеер С., Мейлор С. Объектно-ориентированный анализ: моделирование мира в состояниях. — Киев: Диалектика, 1993.
49. Языковые средства диалога человека и ЭВМ, т. 2. / Под ред. В. Н. Четверикова. — М.: Высшая школа, 1990.
50. Яковлев С. А. Эволюционные имитационные модели процессов и систем как методологическая основа интеллектуальных технологий обучения // Тез. докл. Междунар. конф. «Современные технологии обучения». — СПб., 1996.
51. Яковлев С. А. Комплексный компьютерный учебник «Имитационное моделирование систем» // Тез. докл. Междунар. конф. «Современные технологии обучения». — СПб., 1995.
52. Яковлев С. А., Арсеньев Б. П., Ильин В. П. Интеграция распределенных баз данных на основе сетевых технологий — СПб., изд. ГЭТУ, 1998.
53. Яковлев С. А., Шабуневич Е. В. Моделирующий обучающий комплекс «Имитационные эксперименты с моделями интегральных сетей» // Тез. докл. Межвуз. конф. «Проблемы профессиональной подготовки». Пушкин: 1996.
54. Яковлев С. А., Шабуневич Е. В. Системное моделирование процессов управления информационными ресурсами в цифровых сетях интегрального обслуживания // Сб. науч. тр. «Вычислительная техника и новые информационные технологии». — Уфа: изд. УГАТУ, 1997.

Оглавление

Предисловие	3
Введение. Современное состояние проблемы моделирования систем	5
В.1. Моделирование как метод научного познания	5
В.2. Использование моделирования при исследовании и проектировании сложных систем	7
В.3. Перспективы развития методов и средств моделирования систем в свете новых информационных технологий	9
Контрольные вопросы	19
Глава 1. Основные понятия теории моделирования систем	20
1.1. Принципы системного подхода в моделировании систем	20
1.2. Общая характеристика проблемы моделирования систем	25
1.3. Классификация видов моделирования систем	31
1.4. Возможности и эффективность моделирования систем на вычислительных машинах	38
Контрольные вопросы	44
Глава 2. Математические схемы моделирования систем	45
2.1. Основные подходы к построению математических моделей систем	45
2.2. Непрерывно-детерминированные модели (<i>D-схемы</i>)	50
2.3. Дискретно-детерминированные модели (<i>F-схемы</i>)	54
2.4. Дискретно-стохастические модели (<i>P-схемы</i>)	60
2.5. Непрерывно-стохастические модели (<i>Q-схемы</i>)	64
2.6. Сетевые модели (<i>N-схемы</i>)	71
2.7. Комбинированные модели (<i>A-схемы</i>)	75
Контрольные вопросы	83
Глава 3. Формализация и алгоритмизация процессов функционирования систем	84
3.1. Методика разработки и машинной реализации моделей систем	84
3.2. Построение концептуальных моделей систем и их формализация	88
3.3. Алгоритмизация моделей систем и их машинная реализация	94
3.4. Получение и интерпретация результатов моделирования систем	102
Контрольные вопросы	107
Глава 4. Статистическое моделирование систем на ЭВМ	108
4.1. Общая характеристика метода статистического моделирования	108
4.2. Псевдослучайные последовательности и процедуры их машинной генерации	115
4.3. Проверка и улучшение качества последовательностей псевдослучайных чисел	125
4.4. Моделирование случайных воздействий на системы	130
Контрольные вопросы	143

Глава 5. Инструментальные средства моделирования систем	144
5.1. Основы систематизации языков имитационного моделирования	144
5.2. Сравнительный анализ языков имитационного моделирования	152
5.3. Пакеты прикладных программ моделирования систем	159
5.4. Базы данных моделирования	173
5.5. Гибридные моделирующие комплексы	197
Контрольные вопросы	206
Глава 6. Планирование машинных экспериментов с моделями систем	207
6.1. Методы теории планирования экспериментов	207
6.2. Стратегическое планирование машинных экспериментов с моделями систем	220
6.3. Tактическое планирование машинных экспериментов с моделями систем	226
Контрольные вопросы	239
Глава 7. Обработка и анализ результатов моделирования систем	240
7.1. Особенности фиксации и статистической обработки результатов моделирования систем на ЭВМ	240
7.2. Анализ и интерпретация результатов машинного моделирования	248
7.3. Обработка результатов машинного эксперимента при синтезе систем	253
Контрольные вопросы	259
Глава 8. Моделирование систем с использованием типовых математических схем	260
8.1. Иерархические модели процессов функционирования систем	260
8.2. Моделирование процессов функционирования систем на базе <i>Q-схем</i>	264
8.3. Моделирование процессов функционирования систем на базе <i>N-схем</i>	285
8.4. Моделирование процессов функционирования систем на базе <i>A-схем</i>	295
Контрольные вопросы	306
Глава 9. Моделирование для принятия решений при управлении	307
9.1. Гносеологические и информационные модели при управлении	307
9.2. Модели в адаптивных системах управления	315
9.3. Моделирование в системах управления в реальном масштабе времени	318
Контрольные вопросы	322
Глава 10. Использование метода моделирования при разработке автоматизированных систем	323
10.1. Общие правила построения и способы реализации моделей систем	323
10.2. Моделирование при разработке распределенных автоматизированных систем и информационных сетей	328
10.3. Моделирование при разработке организационных и производственных систем	331
Контрольные вопросы	335
Заключение. Перспективы использования компьютерного моделирования в информационном обществе	336
Список литературы	340

Учебное издание

**Советов Борис Яковлевич
Яковлев Сергей Алексеевич**

МОДЕЛИРОВАНИЕ СИСТЕМ

Ведущий редактор Н. Е. Овчеренко.

Художественный редактор Ю. Э. Иванова.

Художник К. Э. Семенков.

Технические редакторы Л. А. Овчинникова, Н. В. Быкова.

Оператор В. Н. Новоселова.

Корректор О. Н. Шебашова.

Компьютерная верстка Н. С. Михайлова.

ЛР № 010146 от 25.12.96. Изд. № ВТИ-90. Слано в набор и подп. в печать 19.04.2001
Формат 60x88¹/₁₆. Бум. газетн. Гарнитура «Таймс». Печать офсетная
Объем: 21,07 усл. печ. л., 21,07 усл. кр.-отт., 23,30 уч.-изд. л.
Тираж 6000 экз. Зак. № 1047

ГУП «Издательство «Высшая школа», 127994, Москва, ГСП-4, Неглинная ул., 29/14

Факс: 200-03-01, 200-06-87

E-mail: V-Shkola@g23.relcom.ru <http://www.v-shkola.ru>

Набрано на персональных компьютерах издательства

Отпечатано во ФГУП ИПК «Ульяновский Дом печати»
432980, г. Ульяновск, ул. Гончарова, 14