

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
Государственное образовательное учреждение
высшего профессионального образования
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ

А. Г. Варжапетян

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ НА GPSS/Н

Учебное пособие

Санкт-Петербург
2007

УДК 519.682
ББК 22.18
В18

Рецензенты:

кафедра морских информационных технологий Российского
государственного гидрометеорологического университета;
доктор технических наук, профессор кафедры вычислительных систем
и информатики Государственного университета водных коммуникаций
В. В. Фомин

Утверждено редакционно-издательским советом университета
в качестве учебного пособия

Варжапетян А. Г.

В18 Имитационное моделирование на GPSS/H: учебное пособие /
А. Г. Варжапетян; ГУАП. — СПб., 2007. — 384 с.: ил.
ISBN 5-8088-0228-8

В учебном пособии рассматриваются вопросы построения, логики действия и начал программирования на современной версии языка имитационного моделирования GPSS/H, который повышает эффективность первой версии Д. Гордона.

В пособии дается сравнение GPSS/H с другими языками имитационного моделирования, исследуются вопросы статистического оценивания результатов моделирования и сокращения числа реализаций. В отдельных главах подробно изучаются особенности работы мощного отладчика языка имитационного моделирования; рассматриваются вопросы повышения эффективности GPSS/H за счет анимации результатов и интеграции с языком более высокого уровня SLX. Описываемые принципы иллюстрируются большим числом примеров и упражнений, что позволяет освоить методы моделирования.

Предназначено для преподавателей основ имитационного моделирования; исследователей, желающих использовать методы языка имитационного моделирования в практической деятельности; студентов и аспирантов в качестве учебно-методического материала.

УДК 519.682
ББК 22.18

ISBN 5-8088-0228-8

© ГУАП, 2007
© А. Г. Варжапетян, 2007

СОДЕРЖАНИЕ

Условные сокращения	7
Предисловие	9
Часть 1	
Общие вопросы компьютерного моделирования	13
Глава 1	
Представление о компьютерном моделировании	13
§ 1.1. Классификация моделей	13
§ 1.2. Компьютерное моделирование	16
§ 1.3. Место имитационных моделей в общей структуре программного обеспечения	19
§ 1.4. Достоинства и недостатки имитационного моделирования	20
§ 1.5. Основные этапы и задачи, реализуемые при имитационном моделировании	22
§ 1.6. Оценка адекватности имитационных моделей	25
Глава 2	
Концептуальная модель для GPSS/H	29
§ 2.1. Классификация математических моделей	29
§ 2.2. Основные обозначения теории массового обслуживания	30
§ 2.3. Некоторые аналитические модели системы массового обслуживания	33
2.3.1. Распределение вероятностей длительности интервалов между заявками	34
2.3.2. Распределение вероятностей длительностей обслуживания	35
2.3.3. Одноканальное обслуживание с пуассоновским входным потоком и экспоненциальным распределением длительностей обслуживания	36
2.3.4. Многоканальное обслуживание с пуассоновским входным потоком и экспоненциальным распределением длительностей обслуживания	39
Глава 3	
Принципы имитационного моделирования	40
§ 3.1. Общие представления	40
§ 3.2. Модельное время	43
§ 3.3. Способы создания квазипараллелизма при имитационном моделировании	52
§ 3.4. Методы имитации случайных чисел	57
3.4.1. Исторический экскурс	57
3.4.2. Принципы моделирования БСВ	60
3.4.3. Методы построения программных генераторов	64
§ 3.5. Оценка качества базовых случайных величин, получаемых от программных генераторов	68

3.5.1. Общие представления	68
3.5.2. Тесты оценки качества БСВ	69
3.5.3. Теоретическая оценка качества генераторов	73
Часть 2	
Язык имитационного моделирования GPSS/H	77
Глава 4	
Особенности языка имитационного моделирования GPSS/H	79
§ 4.1. История развития GPSS/H и общая идеология	79
§ 4.2*. Сравнение GPSS/H с другими версиями	89
§ 4.3. Основные правила работы с пакетом GPSS/H (студенческая версия)	92
§ 4.4. Структура объектов модели	94
§ 4.5. Формат записи модельного файла	99
Глава 5	
Принципы функционирования языка имитационного моделирования GPSS/H	105
§ 5.1. Характеристики основных операторов	105
5.1.1. Операторы блоков	106
5.1.2. Операторы управления	126
5.1.3. Операторы описания	139
§ 5.2. Основные атрибуты GPSS/H	142
5.2.1. Стандартные числовые атрибуты	143
5.2.2. Стандартные логические атрибуты	145
5.2.3. Стандартные символьные атрибуты	145
§ 5.3. Правила окончания процесса имитационного моделирования	145
5.3.1. Правило окончания по числу стартов	146
5.3.2. Правило окончания по времени испытаний	149
§ 5.4. Принципы движения транзактов	151
5.4.1. Общие основы движения транзактов	151
5.4.2*. Принципы управления транзактами	155
5.4.3*. Общие основы изменения флага состояния модели и индикатора состояния Хакт	163
§ 5.5. Разбор ошибок и упражнения	165
5.5.1. Разбор ошибок	166
5.5.2. Упражнения	168
Глава 6	
Модели одноканального и многоканального обслуживания	171
§ 6.1. Общие представления	171
§ 6.2. Одноканальное обслуживание	173
6.2.1. Схема одноканального обслуживания без ухода Хакт из модели	176
6.2.2. Сбор дополнительной информации (использование очередей)	180

§ 6.3. Многоканальное обслуживание	184
6.3.1. Многоканальное обслуживание групп идентичных серверов	186
6.3.2. Перекрытие памятей и введение Хакт только при необходимости	190
6.3.3. Получение ряда реплик в одном пакетном режиме	192
6.3.4. Изменение приоритета транзакта в процессе ИМ	196
6.3.5. Использование уровней приоритета для управления получением данных	200
§ 6.4. Автоматизация процесса имитационного моделирования и сокращение итогового отчета	202
6.4.1. Автоматизация процесса ИМ	202
6.4.2. Сокращение итогового отчета	204
§ 6.5. Разбор ошибок и упражнения	205
6.5.1. Разбор ошибок	205
6.5.2. Упражнения	207

Глава 7

Работа с отладчиком программ моделирования (тестовый режим)	212
§ 7.1. Особенности работы с отладчиком	212
7.1.1. Запуск отладчика	212
7.1.2. Содержание окон	214
7.1.3. Выход из сеанса отладчика	216
§ 7.2. Функциональные клавиши и команды отладчика	216
7.2.1. Функциональные клавиши	216
7.2.2. Команды и коды объектов	217
§ 7.3. Основы использования отладчика	220
§ 7.4. Применение основных команд в тестовом режиме	222
7.4.1. Команда <u>D</u> ISPLAY	223
7.4.2. Команды <u>T</u> RAP и <u>U</u> NTRAP	224
7.4.3. Команды <u>B</u> REAK и <u>U</u> NBREAK	225
7.4.4. Команда <u>A</u> T	226
7.4.5. Команды <u>R</u> N, <u>C</u> ONTINUE, <u>S</u> TEP	227
7.4.6. Команды <u>S</u> TOP, <u>S</u> ET	228
7.4.7. Пример использования введенных команд	228
§ 7.5. Практические советы по работе с отладчиком	233
§ 7.6. Упражнения	234

Глава 8

Статистические возможности языка имитационного моделирования GPSS/H	238
§ 8.1. Генерация случайных переменных с заданной функцией распределения	238
8.1.1. Обоснование выбора функции распределения случайных величин	238
8.1.2. Генерация БСВ в GPSS/H	240
8.1.3. Получение случайных чисел с заданными функциями распределения	242

8.1.4. Получение непрерывных и дискретных функций	250
§ 8.2. Планирование процесса имитационного моделирования . .	261
8.2.1. Продолжительность процесса ИМ	262
8.2.2. Статистическое планирование и регрессионный анализ	268
8.2.3. Имитационное моделирование и метод «бутстреп» . . .	276
§ 8.3. Особенности процессов имитационного моделирования .	278
8.3.1. Типы процессов ИМ	279
8.3.2. Использование потоков, дополняющих БСВ (антитез) . .	284
§ 8.4. Выбор наилучшей альтернативы в Парето-оптимальном	
множестве	293
8.4.1. Выбор из двух альтернатив в процессе ИМ	294
8.4.2. Выбор лучшей из k сравниваемых альтернатив	302
§ 8.5. Упражнения	311
Часть 3	
Расширение возможностей имитационного моделирования на языке	
имитационного моделирования GPSS/H	313
Глава 9	
Анимационный пакет Proof Animation	313
§ 9.1. Философия анимации	313
§ 9.2. Краткое описание программного продукта Proof Anima-	
tion	315
9.2.1. Добавление анимации к моделированию с использова-	
нием Proof	315
9.2.2. Структура ПП Proof	318
9.2.3. Выполнение основных действий	321
Глава 10	
Интегрированный пакет SLX	324
§ 10.1. Представление о языке SLX	324
10.1.1. Введение в SLX (Simulation Language with Extensibi-	
lity)	324
10.1.2. Особенности языка SLX	326
§ 10.2. Пример использования SLX	339
Заключение	348
Приложения	350
Приложение 1. Операторы блоков GPSS/H	350
Приложение 2. Операторы управления и описания	359
Приложение 3. Перечень функций, встроенных в GPSS/H	365
Приложение 4. Описание эха МФ и граф итогового отчета	366
Приложение 5. Ответы к упражнениям	371
Библиографический список	384

УСЛОВНЫЕ СОКРАЩЕНИЯ

- АМП — амперпеременная
БСВ — базовая случайная величина
ВД — временная дискрета
ВДв — время движения
ВЭ — вычислительный эксперимент
ГПС — гибкая производственная система
ГСЧ — генератор случайных чисел
ДО — дисциплина обслуживания
ДСВ — дискретная случайная величина
ЕВА — единица времени анимации
ЖЦ — жизненный цикл
ИН — идентификационный номер
ИПС — индикатор перехода к сканированию
ИС — индикатор состояния
ИТ — информационные технологии
КМ — компьютерное моделирование
ЛМВ — локальное модельное время
ММ — математическое моделирование
МФ — модельный файл
НОД — наибольший общий делитель
ОБ — оператор блока
ОВМ — обобщенный «будстреп»-метод
ОВО — оператор выходного отчета
ОО — оператор описания
ООП — объектно-ориентированное программирование
ОУ — оператор управления
ПО — программное обеспечение
ПП — программный пакет (продукт)
ППП — пакет прикладных программ
РР — равномерно распределенная случайная величина
СБС — список будущих событий
СЛА — стандартный логический атрибут
СМО — система массового обслуживания
СОБ — следующий оператор блока
СП — список пользователя
СС — счетчик свершений
ССА — стандартный символьный атрибут
СТС — список текущих событий
СУ — система управления
СУБД — система управления базами данных
СЧА — стандартный числовой атрибут
СЭ — случайный элемент
ТОБ — текущий оператор блока
УПМ — управляющая программа моделирования
ФИС — флаг изменения состояния
ЦКП — центральное композитное планирование
ЯИМ — язык имитационного моделирования

ПРЕДИСЛОВИЕ

Нарастающее усложнение систем управления всех видов (технических, организационно-технических и организационных) приводит к пониманию невозможности адекватного описания процессов функционирования таких систем только аналитическими методами. Недаром мировая практика научных исследований свидетельствует о том, что методы компьютерного моделирования (КМ) занимают около 70 % в общем объеме исследовательского инструментария.

Стремительное внедрение новых информационных технологий (ИТ) в корне меняет методы познания окружающего нас мира. На знамени XXI века начертаны слова: информационные технологии, методы управления и качество.

Информационные технологии включают в себя компьютерную поддержку жизненного цикла изделий (CALS-технологии), интеграцию всех этапов разработки и изготовления продукции (ICAM-технологии), автоматизацию разработки программного обеспечения (ПО) (CASE-технологии), создание вычислительных сетей различного уровня и многое-многое другое.

Современные методы управления должны учитывать идеи реинжиниринга корпораций, ситуационного менеджмента, целевого проектирования, ориентацию на суженное производство (lean production) и, соответственно, опираться на возможности современных ИТ.

Методы менеджмента качества сложных систем не мыслятся сегодня без использования инструментов инжиниринга качества, таких как структурирование или развертывание функции качества, робастное проектирование, система Махаланобиса—Тагути, метод шести сигм и т. д. Естественно, что ни один из этих методов не может не опираться на современное ПО.

Даже такое простое перечисление ключевых направлений современного научно-технического прогресса позволяет заключить, что исследование и разработка новых проектов невозможна без использования методов компьютерного моделирования. Мир КМ многогранен и охватывает спектр от моделирования простых систем и производственных ситуаций до исследования взаимодействия социальных, экономических и природных систем в их сложном переплетении и взаимодействии. Рассмотрению проблем КМ посвящен ряд публикаций [5, 6, 8, 9].

На протяжении ряда лет автор, работая в промышленности, занимался вопросами моделирования сложных систем управления [2, 3]. В данном учебном пособии автор ставит более скромную задачу — рассмотреть особенности языка имитационного моделирования (ЯИМ) GPSS/H. Зачастую упоминание об этом ЯИМ вызывает легкое отторжение, так как у всех возникает воспоминание о первой версии GPSS 1963 г. Однако рассматриваемая здесь версия GPSS/H 1999 г. не только коренным образом отличается от ранних версий, но и имеет ряд уникальных черт, делающих ЯИМ приближенным к универсальным языкам программирования и открывающим ряд новых возможностей, весьма важных при исследовании систем

массового обслуживания (СМО). Под разряд СМО подпадают многие задачи из перечисленных направлений. Опыт преподавания ЯИМ в Санкт-Петербургском государственном университете аэрокосмического приборостроения в рамках курсов «Исследование систем управления», «Автоматизированные системы научных исследований» и «Менеджмент качества» показывает, что студенты и аспиранты не только осваивают идеи ЯИМ GPSS/H, но и успешно применяют их в своей практической инженерной и управленческой деятельности.

Прекрасно понимая, что проблема КМ весьма сложна и ее разные аспекты требуют отдельных монографий, автор ограничился рассмотрением специфики и возможностей одного из ЯИМ, а именно GPSS/H. За рубежом его описанию посвящено большое число публикаций, например [12–15], а на русском языке написана только глава в монографии автора [4]. Тем не менее, ЯИМ заслуживает большего внимания и несомненно достоин распространения не только для целей обучения студентов, но и для практического использования инженерами, менеджерами и научными работниками.

Литературные источники легко могут быть пополнены с помощью поисковых систем Интернета по ключевому слову Simulation — GPSS/H.

Автор выражает искреннюю признательность профессору Мичиганского университета США Томасу Шрайберу, творческие контакты с которым способствовали написанию этого учебного пособия, а также президенту Wolverine Software Corp. Джеймсу Хенриксену, предоставившему в распоряжение автора ряд уникальных материалов.

Автор с сожалением констатирует, что в известных ему учебных заведениях до сих пор пользуются первыми версиями ЯИМ GPSS, и надеется, что пособие будет способствовать продвижению новых идей.

Учебное пособие рассчитано на широкий круг читателей: так, исследователи смогут применять возможности ЯИМ для решения своих задач, преподаватели — использовать материалы пособия для лекций, а студенты и аспиранты — для изучения общих идей имитационного моделирования.

Глоссарий

Необходимость введения словаря, предваряющего основной текст, объясняется тем, что:

— в существующей научно-технической литературе используются различные термины для определения одних и тех же или сходных понятий, связанных с КМ;

— область моделирования, представляемая в пособии, достаточно узка и имеет специфические определения, не используемые порой при моделировании глобальных систем.

Определения, вошедшие в словарь, заимствованы из источников [4, 13] и ГОСТов.

Амперпеременная — переменная, введенная в GPSS/H и резко увеличивающая его вычислительную мощность. Название объясняется тем, что переменная обозначается знаком амперсанд — &.

Антитеза — значение случайного числа дополнительного потока, являющееся разностью между единицей и значением случайного числа основного потока.

Атрибут — стандартные числовые, символьные и логические характеристики, а также характеристики встроенных функций.

Буфер (накопитель или память) — одна из единиц аппаратной категории, способная накапливать приходящие заявки.

Вычислительный эксперимент (прогон) — процесс получения характеристик моделируемой системы на компьютере в течение одной *реплики*. Число реплик в прогоне зависит от заданной статистической точности или от времени, отведенного на прогон.

Имитационное моделирование — часть компьютерного моделирования, позволяющая: а) получать на ЭВМ статистические характеристики случайного процесса функционирования системы, б) определять лучший способ управления в детерминированных структурах путем проведения вычислительного эксперимента.

Заявка (в GPSS — транзакт) — динамический элемент, движущийся по модели от оператора к оператору блоков строго один в единицу времени.

Канал — путь прохождения заявок через систему массового обслуживания (различают одноканальные и многоканальные СМО).

Компьютерное моделирование — расширение возможностей математического моделирования за счет использования последних достижений ИТ, внедряемых во все сферы человеческой деятельности.

Модель — отображение особенностей реальной системы путем создания концептуального или машинного описания.

Модуль — часть программы ИМ, включающая неисполняемый модуль описания, модуль исполнения и модуль управления.

Оператор — элемент ЯИМ, предназначенный для построения машинной модели (различают операторы описания (compiler directives), операторы блоков (blocks) и операторы управления (control statements)).

Операнд — числовое или символьное значение одной из характеристик оператора любого вида.

Поток — последовательность событий при моделировании (различают потоки случайных чисел, генерируемых встроенными генераторами, входные потоки заявок на обслуживание, выходные потоки обслуженных транзактов, потоки необслуженных заявок).

Прибор (устройство) — одна из единиц аппаратной категории, осуществляющая обслуживание транзактов.

Приоритет — показатель значимости приходящего транзакта, являющийся одним из его операндов.

Прогон — проведение вычислительного эксперимента с моделью, состоящего из n реплик.

Процесс — совокупность событий, действий и связанных с ними временных отрезков, характеризующих функционирование системы или ее модели.

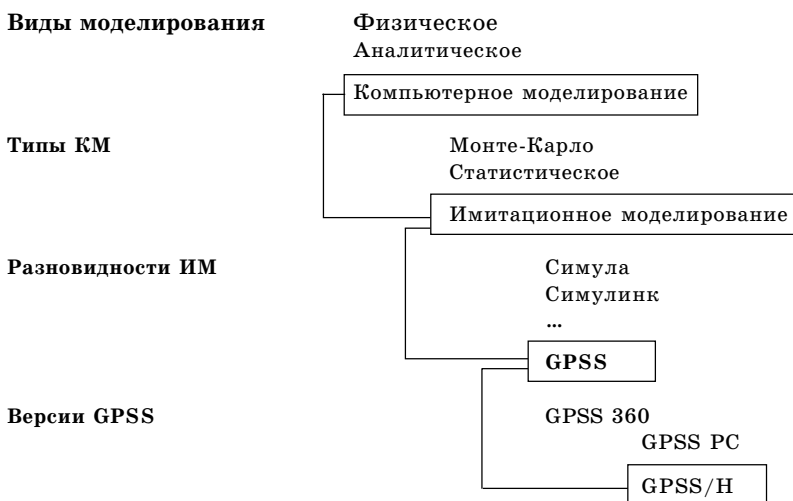
Список (последовательность) (в первоисточнике Chain — цепь) — перечень транзактов, находящихся в одном из шести возможных списков: будущих событий, текущих событий, прерываний, синхронизации, пользователя и, очень редко, применяемого системного.

Реплика — одна реализация вычислительного эксперимента или прогона.

Примечание. Приведенный список терминов дает общее представление о направленности пособия, внесенные в глоссарий термины будут ниже определены более подробно. Здесь будем придерживаться англоязычной нотации и употреблять сочетания «плавающая точка», хотя нам привычней говорить «запятая».

Структура учебного пособия

Структура предлагаемого учебного пособия может быть представлена следующей блок-схемой, на которой показана логика выбора целевой направленности пособия, которая ограничена исследованием особенностей и возможностей GPSS/H.



Следуя указанной логике, пособие состоит из трех частей (десяти глав) и приложений.

Первая часть «Общие вопросы компьютерного моделирования» содержит три главы. Первая посвящена общим представлениям о классификации моделей, уточнению места компьютерного моделирования, содержанию его отдельных разделов, уточнению схемы и порядка КМ. Во второй главе рассматриваются элементы теории массового обслуживания, являю-

щиеся основой для транзактного способа создания квазипараллелизма при ИМ. В третьей главе рассматриваются вопросы определения модельного времени, создания квазипараллелизма и особенности генерирования базовых случайных величин (БСВ) с помощью генераторов случайных чисел.

Вторая часть «Язык имитационного моделирования GPSS/H» содержит пять глав. В четвертой и пятой главах уточняется место GPSS/H среди других ЯИМ, проводится сравнительный анализ всех версий ЯИМ GPSS, рассматриваются принципы построения и функционирования ЯИМ GPSS/H. В первую очередь обращается внимание на внутреннюю организацию ЯИМ, что позволит более осознанно работать с пакетом. Опыт преподавания выявил ряд моментов, наиболее трудных для восприятия студентами. К их числу относятся правила прекращения испытаний, организация непоследовательного прохождения транзактов, правила терминирования и некоторые другие. Шестая глава посвящена рассмотрению моделей одноканального и многоканального обслуживания в пакетном режиме. Глава содержит ряд параграфов, посвященных различным аспектам ИМ (сбор дополнительной информации, автоматизации процесса ИМ, использование приоритета и т. п.). Седьмая глава посвящена работе с отладчиком (дебаггером) ЯИМ — рассмотрены особенности статистические задачи, посвященные использованию антитез, позволяющих сократить число реплик при одновременном уменьшении дисперсии, а также задача выбора наилучшего среди различных вариантов.

Третья часть содержит краткое описание новых программных пакетов (ПП), расширяющих возможности ЯИМ GPSS/H.

В приложениях приведены многие дополнительные характеристики ЯИМ, которые важны для специалистов, непосредственно использующих пакет ЯИМ, но могут затруднять восприятие материала при изучении основных положений.

Адрес приобретения ПП

Разработчиком и распространителем ПП GPSS/H:

— профессиональной 32-разрядной версии, работающей под Windows 98, 2000, NT, XP;

— студенческой версии, работающей в эмуляции DOS или под любой оболочкой типа NC, VC, Far;

— пакета анимации движения транзактов Proof Animation;

— интегрированного пакета SLX (C + GPSS/H + специализированный язык) — является

Wolverine Software Corporation

7617 Little River Turnpike, Suite 900

Annandale, VA 22003-2603

mail@wolverinesoftware.com

Все реквизиты и прайс-листы ПП можно найти на сайте корпорации.

Часть 1
ОБЩИЕ ВОПРОСЫ
КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ

Глава 1
ПРЕДСТАВЛЕНИЕ
О КОМПЬЮТЕРНОМ МОДЕЛИРОВАНИИ

§ 1.1. КЛАССИФИКАЦИЯ МОДЕЛЕЙ

Возможность представления моделью (лат. *modulus* — мера, образец) природных явлений, процессов или объектов окружающего нас мира присуща человеку-исследователю еще с ранних этапов развития общества. «Модель (по определению БСЭ) — образ (в том числе условный или мысленный) какого-либо объекта или системы объектов, используемый при определенных условиях в качестве их “заместителя” или “представителя”.» Достаточно вспомнить хрустальный небесный свод геоцентрической модели, модель атома и т. д. При представлении модели средствами математики и логики возникает абстрактный образ реального объекта, при исследовании образа реального объекта в качестве модели имеет место конкретное исследование. Таким образом, моделирование, в том числе и имитационное, находится в промежутке между этими двумя крайними точками (рис. 1.1).

Модель должна ответить на множество вопросов исследователя: что будет, если ...?; каковы размеры ...?; насколько корректны упрощения ...? и множество других.

В результате модель из вспомогательного средства, заменяющего исследуемый объект (модель автомашины, портновский манекен), стала превращаться в способ получения информации о вновь создаваемой исследуемой или управляемой системе.

Подчеркнем, что под *информацией* будем понимать не столько продукт человеческого разума, получаемый в процессе познания, сколько объективную философскую категорию, связывающую темп

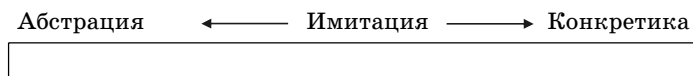


Рис. 1.1. Место имитационного моделирования в модельном пространстве

процессов, происходящих в системе, с уровнем организации самой системы.

Информация через философскую категорию *отражение* связана с категорией *материя* [4]. Под отражением понимается свойство материи воспринимать и сохранять в своей структуре следы воздействия другой системы. По А. Урсулу, информация — отраженное разнообразие реальности мира. Отсюда, чем полнее и разнообразнее модель динамической, нелинейной и неравновесной системы, тем более адекватно она отображает реальную систему. Информация объективно присутствует всегда, даже в случае классической механики; так, при соударении двух тел второе получало информацию от первого, как ему двигаться. Правда, при этом вся информация сводилась к одной динамической силе и не давала ничего нового, поэтому и не бралась в расчет. Учет информации даже в простых случаях классической динамики мог бы дать единую концепцию, объединяющую все разделы физики. Однако этот вопрос при всей его важности не имеет отношения к настоящему учебному пособию.

Вернемся к нашему представлению о моделях, полагая, что любой алгоритм — это модель деятельности, а в силу системности Вселенной любая целесообразная деятельность невозможна без моделирования.

Классификация системного мира моделей весьма широка, поэтому рассмотрим суженную классификацию, отвечающую задачам пособия, и дадим к ней краткое пояснение (рис. 1.2).

Ф — физическое (прямое) моделирование, Ф1 предусматривает использование в качестве модели саму систему (опытный образец), а Ф2 — другую систему со схожей физической природой (макет авто-

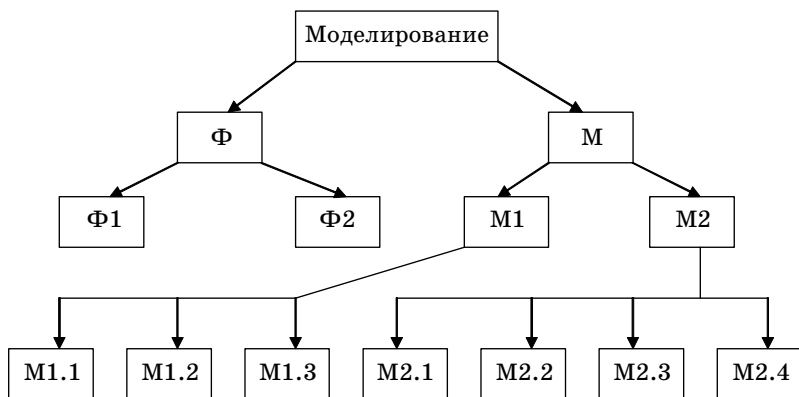


Рис. 1.2. Классификация методов моделирования

мобиля, сооружения, плотины). Такой вид моделирования способствовал созданию теории подобия.

М — математическое моделирование (ММ), распадающееся на две большие группы **М1** и **М2**.

М1 — аналитическое моделирование, которое можно разделить на **М1.1** — явное аналитическое описание искомым характеристик системы на одном из языков математики (см. гл. 2); **М1.2** — приближенные численные методы, когда все объекты аппроксимируются числами или их комплектами в принятой числовой сетке, а результаты получаются в виде таблиц или графиков; **М1.3** — качественные методы, когда изучаются свойства решений задач данного класса без нахождения самих решений. Зачастую эти методы реализуются с помощью экспертного оценивания. Такого вида методы широко используются в теории качества, квалиметрии, экономике, социологии и т. д.

М2 — компьютерное моделирование (см. § 1.2), когда математическая модель интерпретируется в программу для ЭВМ. Характерно, что с появления статьи Дж. Неймана и С. Улама в 1948 г. — первой работы по применению метода Монте-Карло, многие специалисты продолжают называть КМ методами Монте-Карло или статистических испытаний. Это в принципе не верно, так как КМ разделилось на четыре направления [5, 9] (см. рис. 1.2).

М2.1 — методы Монте-Карло или методы вычислительной математики, использующие методы **М1.2** с учетом возможностей современных компьютеров. Этими методами можно вычислять любые, не берущиеся аналитическим путем, многократные интегралы, решать системы уравнений. Интересующимся методами вычислительной математики следует обратиться к многочисленной литературе.

М2.2 — методы ИМ (simulation), для которых характерно воспроизведение на ЭВМ процесса функционирования системы с сохранением его логической структуры и последовательности его протекания во времени, что позволяет путем многократного повторения набрать необходимые статистические данные и судить о состоянии объекта в различные моменты времени, оценивать выходные характеристики, выбирать оптимальное поведение или проводить сравнение альтернативных вариантов. Основной акцент пособия делается на рассмотрение именно ИМ.

М2.3 — методы статистической обработки данных моделирования на основе методов планирования эксперимента. Имеется целый ряд монографий, посвященных этим вопросам, в том числе [1, 7].

В настоящее время существует большое число пакетов прикладных программ (ППП), которые условно можно разбить на три группы:

— пакеты углубленного статистического анализа, написанные специалистами по статистике для таких же специалистов, с собственным языком, позволяющим программировать новые статистические процедуры (SAS, Statgraphics);

— пакеты базовой статистики, ориентированные на пользователей, не являющихся специалистами по статистическому анализу, и содержащие классические методы анализа с дружественным пользовательским интерфейсом в виде многочисленных пояснений, примеров и подсказок, среди них необходимо отметить прекрасный раздел по статистике в ППП MATLAB и ППП «Статистика»;

— проблемно-ориентированные пакеты, использующие терминологию и критерии в данной предметной области (экология, медицина и т. п.).

М2.4 — комплексы ИМ, объединяющие все названные виды КМ, пользовательский интерфейс, автоматизированные системы поддержки принимаемых решений и т. д. Это перспективное развивающееся направление предназначено для исследования сложных систем (подробнее см. работу [9]).

§ 1.2. КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ

Из классификации (см. рис. 1.2) в силу направленности пособия будем рассматривать только один из видов КМ, а именно имитационное моделирование СМО, и частично вопросы статистической обработки результатов ИМ.

Тем не менее, приведем краткое описание математического моделирования, которое позволяет благодаря абстрактным математическим формулам точно, однозначно и количественно оценить исследуемый объект. Усложнение исследуемых систем привело к резкому усложнению их математического описания, что, в свою очередь, приводило к необходимости делать всевозможные упрощающие допущения. При этом возникла опасность ухода от реального представления о системе. Выходом из этого положения являлся либо прогресс самих математических методов, либо изыскание иных методов описания. Появление мощных современных компьютеров и возникновение информационных технологий привело ко второму рождению ММ. Оно стало вторгаться практически во все сферы человеческой деятельности. В ряде областей ММ стало вытеснять физическое моделирование, так произошло, в частности, в авиационной промышленности, где начался демонтаж аэродинамических труб. Дальнейший прогресс ММ идет за счет:

— разработки новых численных методов решения задач моделирования, возможных только при условии использования ИТ;

— стремительного увеличения объемов памяти и производительности ЭВМ, что позволяет на порядки увеличить размерности решаемых задач и перейти к качественно новым задачам моделирования. Так, прогресс ММ позволил: исследовать эффекты синергизма (комбинированное действие, когда выходной эффект системы превышает действие, оказываемое компонентами по отдельности); оценивать бифуркационные состояния (вероятностное разветвление процесса функционирования системы); прогнозировать развитие диссипативных структур (переход в качественно новое состояние, характеризующееся более высоким уровнем самоорганизации); создавать более совершенные модели Вселенной (Большой взрыв, горячая Вселенная, теория струн), теории искусственного интеллекта, исследований в разных областях социологии, экономики и т. д.

В то же время возникла мощная оппозиция применению ММ в плохо структурируемых, не формализуемых областях, т. е. в таких областях, где человек (оператор, ЛОР, ЛПР) является основным элементом. По определению академика РАН РФ А. А. Самарского, процесс ММ базируется на триаде «модель—алгоритм—программа». До появления ЭВМ основную роль играла модель в виде математических уравнений, алгоритм представлял собой схему ручных расчетов для приближенного решения уравнений, а программа отсутствовала вообще. В начале использования ЭВМ первого поколения программе отводилась второстепенная роль — представление алгоритма в машинных кодах. Развитие ИТ привело к тому, что ЭВМ стали использовать для моделирования процессов функционирования системы, причем в этом случае имелись алгоритм и программа, а математическая модель в ее классическом виде практически отсутствовала или молчаливо предполагалось, что математической моделью является одно из аналитических представлений. Это направление получило название имитационного моделирования и представлено в работах Н. П. Бусленко, Н. Н. Моисеева, Р. Шеннона и многих других. Таким образом, в ММ началось опережающее развитие третьей компоненты триады — программы или программного обеспечения процесса моделирования.

Необходимо четко понимать разницу между программированием и моделированием. Программирование в настоящее время располагает большим арсеналом языков программирования, средств автоматизации управления вычислительными ресурсами, создания программ, автоматизации работ с большими массивами данных, так называемых систем управления базами данных (СУБД).

Однако весь этот арсенал направлен на создание программ, но модель не должна превращаться только в программу, описывающую абстрактные алгоритмы или логические отношения. Компьютерная модель должна оставаться прежде всего моделью реального объекта не зависимо от того, чем описывается его поведение: набором формул или правил, графиком или прогнозными оценками экспертов. Поэтому модель должна допускать исследование всех интересных возможностей: анализ чувствительности, изменение выходных характеристик, определение областей устойчивости и степень робастности, оптимизацию параметров, оценку вариантов построения и т. д. В связи со сказанным все чаще в литературе [4–6, 11] появляется термин *компьютерное моделирование*. КМ объединяет достижения математического моделирования, системного программирования и информационных технологий.

Автор не претендует на формулирование понятия КМ, помня все перипетии известной басни С. Михалкова «Слон живописец», однако приводит некоторые характерные черты КМ, инвариантные к области применения и направлениям исследования. Итак, КМ обладает:

- способностью понимать, интерпретировать и использовать формализованную и не формализованную информацию (математические формулы, логические правила, вербальные описания и т. п.);
- различными формами представления данных и знаний, заполняя пространство между ММ с его аналитическими формами описания и искусственным интеллектом с его формами и правилами представления знаний;
- способностью участвовать в процессе не только автоматизации научных исследований за счет использования самой ЭВМ для модификации различных режимов применения КМ, но и интеграции всех этапов жизненного цикла системы путем использования быстро развивающихся методов ИТ (широко распространенные во всем мире CALS-технологии, CASE-технологии, технологии ICAM и IDEF);
- возможностью расширения круга пользователей, от узкого круга специалистов-математиков и профессиональных программистов до большого класса исследователей, не обладающих профессиональными знаниями в областях математики и программирования, но хорошо знающих предметную область и умеющих обращаться с ППП.

Помня, что КМ разделяется на четыре подгруппы, следует иметь в виду, что обычно при использовании ЯИМ приходится обращаться и к методам Монте-Карло, и к возможностям ППП, созданных для обработки статистических данных. Поскольку задачей пособия является рассмотрение одного из ЯИМ—GPSS/H, специальные вопросы, относящиеся к М2.1, М2.3, М2.4, здесь не рассматриваются. Неко-

торые вопросы освещаются в работах [4, 5, 11], имеются многочисленные ссылки на литературные источники; кроме того, нельзя забывать о неисчерпаемых ресурсах Интернета.

§ 1.3. МЕСТО ИМИТАЦИОННЫХ МОДЕЛЕЙ В ОБЩЕЙ СТРУКТУРЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

По Р. Шеннону, имитация — это «процесс конструирования реальной системы и постановки эксперимента на ней». При этом любые характеристики определяются за счет проведения прогона или нескольких прогонов модели, каждый из которых включает заданное число реплик (реализаций вычислительного эксперимента). ИМ можно использовать в двух направлениях:

1) рассматривать случайные процессы функционирования системы и определять статистические характеристики, что интересно в первую очередь разработчикам и исследователям системы;

2) при известном или детерминированном процессе функционирования системы определять разные варианты построения, элементов конструкции или стратегии управления, что интересно в первую очередь конструкторам, архитекторам или менеджерам.

Оба направления имеют право претендовать на соответствие классическому определению Шеннона. Чтобы уяснить место имитационных моделей в общей структуре ПО, рассмотрим уровни построения ПО.

Уровень 1. Машинные коды, автокоды, машинно-ориентированные языки, операционные системы.

Уровень 2. Алгоритмические языки высокого уровня (C++, Pascal и др.), системы программирования СУБД.

Уровень 3. Специализированные алгоритмические языки моделирования, в том числе и имитационного (SIMULA, SIMSCRIPT, GPSS и др.).

Уровень 4. Интегрированные системы ИМ (например, SLX, СИМ), автоматизированные системы искусственного интеллекта (экспертные, поддержки принятия решений).

Объекты 1-го уровня не требуют никаких комментариев.

Языки 2-го уровня при их универсальности дороги и сложны.

Языки 3-го уровня, теряя в универсальности, приобретают направленность на конкретную область и становятся простыми. Отметим, что GPSS/H, сохранив все преимущества языков 3-го уровня, вобрал в себя многие положительные черты языков 2-го уровня (подробнее см. гл. 4).

Учитывая, что число ЯИМ на сегодняшний день превышает 600, выбор ЯИМ зависит от многих факторов: предметной области, квалификации пользователя, наличия соответствующей ИТ и т. д.

Языки ИМ обладают рядом весьма привлекательных достоинств:

- удобством описания структуры исследуемой системы;
- учетом динамики функционирования и начальных условий;
- возможностью повторения испытаний;
- возможностью отладки и контроля;
- сбором и анализом необходимой статистики и принятием решений;
- простотой восприятия;
- простотой обучения;
- возможностью использования разработчиками системы, не являющимися специалистами в области математики и программирования.

Четвертый уровень включает в себя проблемно-ориентированные интерактивные системы, которые можно разделить на автоматизированные экспертные, оптимизационные системы, а также имитационно-моделирующие комплексы.

На фоне роста информационных потоков, создания баз знаний, сложных поисковых систем ИМ превращается из средства системного анализа в средство исследования, испытаний и отладки сложных систем. Поэтому перспективы применения ЯИМ не только не утрачивают актуальности, а приобретают все большее значение в процессе создания сложных систем.

§ 1.4. ДОСТОИНСТВА И НЕДОСТАТКИ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

Имитационное моделирование позволяет решать ряд сложных задач и имеет *преимущества*:

— при создании ИМ законы функционирования системы могут быть неизвестны, поэтому постановка задачи исследования является не полной и ИМ служит средством изучения особенностей процесса. При этом можно руководствоваться связями между компонентами и алгоритмами их поведения;

— при проведении ИМ выявляется характер связей между внутренними параметрами системы и выходными характеристиками;

— при проведении ИМ можно менять темп моделирования: ускорять при моделировании явлений макромира (например, процессов на Солнце) или замедлять при моделировании явлений микромира (например, процесс существования элементарных частиц);

- при проведении сравнения и выбора альтернатив;
- при изучении узких мест в системе;
- при подготовке специалистов, осваивающих новую технику.

Из перечисленного следует, что ИМ применяется для решения широкого спектра задач практически любой сложности в условиях неопределенности, когда аналитическое моделирование оказывается практически не применимым.

Достоинства ИМ

1. Возможность объединять традиционные математические и экспериментальные компьютерные методы.

2. Высокая эффективность применения при исследовании АСНИ, САПР, экспертных систем, сложных систем управления. По данным RAND Corp., консалтинговые фирмы из всей гаммы возможных средств анализа: линейного, нелинейного, динамического программирования, методов исследования операций, вычислительных методов — более чем в 60 % случаев прибегают к ИМ, так как ИМ позволяет получать ответы в терминах, понятных и привычных для пользователя.

3. Возможность исследовать объекты, физическое моделирование которых экономически нецелесообразно или невозможно.

4. Испытания объекта связаны с опасностью для здоровья человека.

5. Исследование еще не существующих объектов.

6. Исследование труднодоступных или ненаблюдаемых объектов.

7. Исследование плохо формализуемых экологических, социальных или экономических систем.

8. Исследование объектов практически любой сложности при большой детализации и снятии ограничений на вид функций распределения случайных величин.

Недостатки ИМ

1. Самым существенным недостатком является невозможность получить точечную оценку исследуемых характеристик, так как в результате ИМ можно оценить только математическое ожидание и дисперсию.

2. Потеря общности результатов, так как при ИМ оценивается конкретная система.

3. Трудности оптимизации, так как ИМ отвечает на вопрос, «что будет в случае, если...?», но не определяет, будут ли эти условия наилучшими.

4. Трудности с оценкой адекватности ИМ.

5. Создание ИМ сложной системы длительно по времени и требует значительных денежных средств.

Несмотря на эти недостатки, все большее число исследователей прибегает к использованию ИМ в силу достоинств, указанных выше.

Для составления сложной ИМ необходимы опыт и приобретаемые на практике навыки. Это следует учитывать, чтобы при первых неудачах не наступило разочарование в возможностях ИМ.

§ 1.5. ОСНОВНЫЕ ЭТАПЫ И ЗАДАЧИ, РЕАЛИЗУЕМЫЕ ПРИ ИМИТАЦИОННОМ МОДЕЛИРОВАНИИ

Процесс ИМ системы можно разделить на три последовательно выполняемых этапа (рис. 1.3):

I — построение математической (концептуальной) модели S' (см. гл. 2).

II — разработка моделирующего алгоритма S'' (см. гл. 4);

III — исследование системы S с помощью модели S'' (см. гл. 6, 8).

Процесс ИМ не является строго поступательным, между этапами существуют обратные связи, позволяющие вводить новую информацию, вносить уточнения и корректировки.

Построение математической модели на I этапе включает в себя пять взаимосвязанных подэтапов [4, 11]:

1) уяснение и постановка задачи, определение целей исследования;

2) декомпозиция системы на компоненты, допускающие удобное математическое или алгоритмическое описание;

3) определение параметров, переменных, пространства состояний системы, установление пределов изменения каждой характеристики;

4) выбор выходных показателей, т. е. вектора Q ;

5) аналитическое описание с помощью выбранной концептуальной модели S' системы S и проверка ее адекватности.

Построение математической модели системы S начинается с определения параметров системы и переменных, определяющих процесс функционирования системы.

На II этапе при переходе от концептуальной модели S' к моделирующему алгоритму и имитационной модели S'' можно выделить пять основных подэтапов:

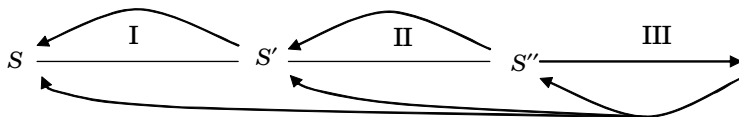


Рис. 1.3. Этапы представления модели

1) выбор способа имитации (см. гл. 3), а также вычислительных и программных средств реализации ИМ;

2) построение логической схемы моделирующего алгоритма;

3) алгоритмизация математических моделей, описывающих поведение элементов системы и связей между ними в рамках выбранного способа имитации;

4) программирование моделирующего алгоритма, т. е. разработка самой имитационной модели;

5) отладка, тестирование и проверка адекватности ИМ.

На III этапе можно выделить три основных подэтапа:

1) планирование вычислительных экспериментов;

2) проведение прогона или прогонов;

3) обработка, анализ и интерпретация результатов.

На I этапе проводится выяснение особенностей, которые представляют интерес для исследователя, уточняются виды и формы преобразования материальных, энергетических и информационных потоков, уточняются взаимодействия с другими системами и окружающей средой. Таким образом осуществляется проблемная ориентация. При этом сближаются точки зрения и снимаются имеющиеся разногласия. Само концептуальное описание правильнее начинать со специальной задачи, а затем универсализировать ее на класс задач. Информация должна получаться из всех возможных источников: Интернета, публикаций в специальных журналах, выбора аналитических моделей, суждений экспертов. Серия стандартных вопросов этого этапа такова:

- что представляет собой система (характеристики, состав, взаимодействия, отношение с окружающей средой);

- как характеристики эволюционируют во времени;

- каков характер корреляционных связей;

- обладает ли система свойствами робастности и т. д.

Все это позволяет описать концептуальную модель, для чего определяются входные параметры, возмущающие воздействия, выходные реакции системы. После чего обязательно проверяется ее адекватность, т. е. насколько модель отвечает реальной системе и какова степень доверия результатам ИМ (см. подробнее § 1.6).

На II этапе производится составление ИМ. В случае описания сложных систем ИМ может создаваться усилиями специалистов разных отделов и даже разных организаций, при этом уровень может доходить до подмоделей, их отладки и сборки общей ИМ. На этом этапе также проводится оценка адекватности, но в отличие от I этапа, где определяется, *правильна ли модель*, на II этапе проверяют, *насколько правильно работает эта модель*, и проводят пилотный прогон модели.

Наконец, III этап является основным, ради которого и строилась модель, на этом этапе набирается статистика, позволяющая принимать решения по любой из основных задач ИМ.

Следующие группы задач ИМ можно отнести к основным.

1. Оценка значений показателей выходного вектора \mathbf{Q} , а также значений параметров компонентов.

2. Нахождение функциональной зависимости между вектором \mathbf{Q} и значениями параметров системы.

3. Сравнение систем с разными вариантами структур и разными значениями параметров для одной функциональной структуры.

4. Оптимизация системы S на множестве параметров на основе двойственной задачи оптимизации (достижение максимума выходного параметра при заданном значении ресурсов либо минимизация ресурсов при достижении заданного значения выходного параметра).

Любая из названных задач может быть решена на III этапе с помощью методов планирования имитационных экспериментов. В результате ИМ системы S при заданном времени и векторе параметров системы получается фазовая траектория системы S

$$\{x(t) \in X, t \in T\} \quad (1.1)$$

и значения выходных показателей для каждого интервала модельного времени и для всего интервала моделирования в целом

$$q(t) = q(\theta', T). \quad (1.2)$$

Результаты одного вычислительного эксперимента (ВЭ), имитирующего фазовую траекторию (1.1) и выходные показатели (1.2), называются *репликой* или реализацией. Итогом одной реплики является одно значение искомой характеристики. Для получения представительной статистики необходимо провести ряд реплик в рамках одного прогона или ряд прогонов с меняющимися условиями, после чего проводится подэтап обработки данных ВЭ. Анализ полученной статистики может выявить и контринтуитивное поведение системы, что может быть вызвано плохим проведением I этапа или невыявленной неадекватностью модели исследуемой системе. Целью обработки статистики является не только оценка характеристик, но и уточнение асимптотических свойств, определение колебательности системы и областей устойчивости. На основе полученных результатов принимаются необходимые решения. Названные группы задач практически перекрывают всю область возможных применений ИМ.

§ 1.6. ОЦЕНКА АДЕКВАТНОСТИ ИМИТАЦИОННЫХ МОДЕЛЕЙ

При ИМ неизбежно возникает проблема обоснования возможности перенесения на исследуемую систему управления (СУ) выводов, полученных при функционировании модели. Под адекватностью ИМ понимаем степень отражения параметрами модели характеристик исследуемой системы с точностью, требуемой для конкретного исследования. Оценка адекватности впрямую мало реальна, и чаще всего руководствуются косвенными соображениями типа:

— согласуются ли результаты со здравым смыслом, причем факт отсутствия противоречий или их наличие еще не доказывают неадекватность модели;

— согласуются ли результаты ИМ с предполагаемыми статистическими оценками.

Тем не менее, в последнее время появился ряд работ, делающих обнадеживающие попытки оценки адекватности, распадающейся на два связанных процесса:

верификации — т. е. проверки идентичности концептуальной модели исследуемой системы;

пригодности модели — возможности перенесения результатов моделирования на исследуемую систему.

Верификация — общепринятая процедура и чаще всего неизбежная. Верификация предусматривает предупредительные и отладочные процедуры.

К *предупредительным* относятся:

- проверка пригодности входных данных, контроль набора и т. п.;
- построение программы в виде трех разделов:
 - структура модели,
 - исходные данные,
 - запуск программы со строгой последовательностью операторов;
- проверка датчиков БСВ;
- проверка точности вычислений (формат данных, округление, усечение).

Отладка начинается с анализа ошибок предыдущих этапов, возможности их повторения, изучения логики программы. Иногда полезно после написания машинной программы снова попытаться построить концептуальную модель. В процедуру отладки также входят корректировка синтаксиса и семантики, анализ чувствительности модели. Отладка ведется по разделам программы.

Оценка пригодности в ИМ — процедура достаточно спорная, считается даже, что она может дискредитировать полезную модель. Кроме того, оценка пригодности, являясь многокритериальным процес-

сом, достаточно сложна, и единой системы критериев для такой оценки не существует.

Анализ ряда статей позволил представить классификацию критериев оценки пригодности (табл. 1.1).

Техническая пригодность должна выяснить обоснованность теоретических посылок, положенных в основу модели. Вначале оцениваются все сделанные допущения, затем оценивается пригодность данных. Выявленные расхождения относятся к «узким местам» модели и должны быть уменьшены.

Операционная пригодность менее категорична, чем техническая, и допускает большие рассогласования. Особое внимание обращается на робастность, включающую анализ чувствительности на ошибки в процессе ИМ при задании экстремальных значений входным пара-

Таблица 1.1. Критерии оценки пригодности

Вид пригодности ИМ	Оценка пригодности		Условия анализа пригодности
Техническая	Теоретическая		Учет математических содержательных и причинных допущений
	Пригодность данных	необработанных	Оценка точности, беспристрастности и репрезентативности данных
		структуризованных	Учет точности операции сравнимости позиций
	Структурная		Правильность отражения внутренних взаимосвязей ИС
	Прогнозная		Способность предсказывать будущее
Операционная	Репликативная		Точность воспроизведения характеристик ИС
	Робастность		Учет чувствительности модели
	Реализационная		Вероятность практического внедрения
Динамическая	—		Актуализация, практика успешного использования модели

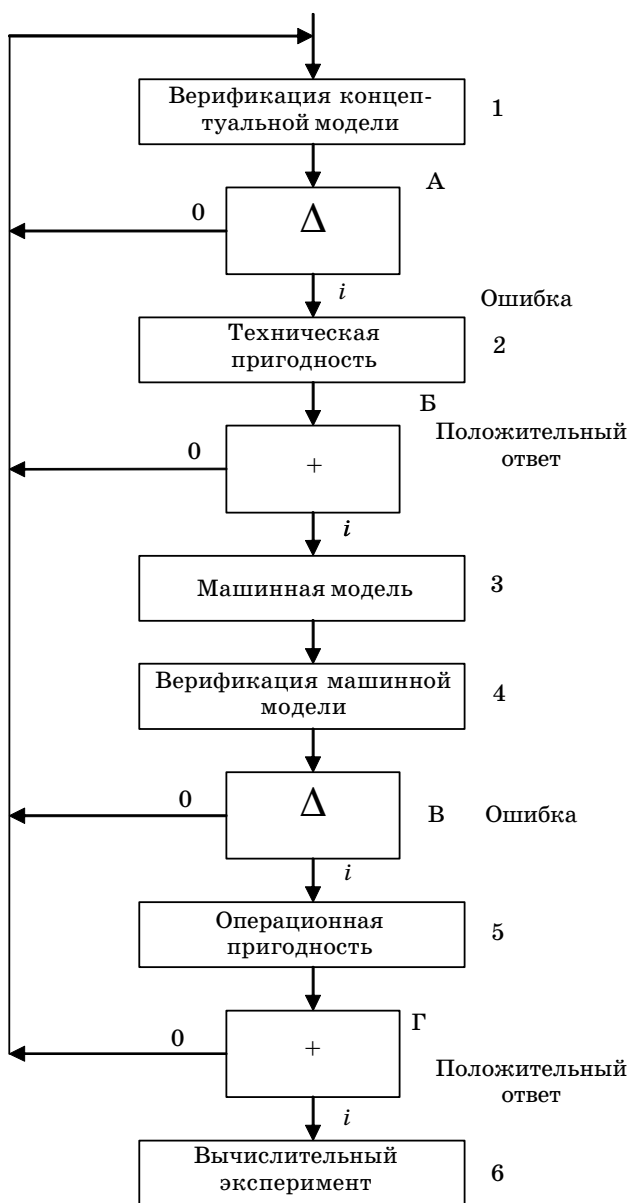


Рис. 1.4. Фрагмент блок-схемы алгоритма

метрам. Репликативная пригодность должна оценивать уровень точности воспроизведения характеристик индикатора состояния (ИС). Формально необходимо иметь две выборки и оценивать их статистическими методами (регрессионный и факторный анализ, хи-квадрат и F -критерии, тесты Тьюринга).

Динамическая пригодность. Расхождения во временном диапазоне, влияющие на операционную пригодность, оцениваются динамической пригодностью ИМ, а также возможностью актуализации и расширения данных.

В результате блок-схема алгоритма ИМ включает ряд действий, показывающий последовательность проведенных работ.

Методы оценки верификации и пригодности делятся на две группы:

- формальные (статистические);
- неформальные с привлечением пользователей и лиц, принимающих решение.

Из фрагмента блок-схемы алгоритма (рис. 1.4) видно, что оценка верификации и пригодности взаимосвязаны. Очевидно, что оценка адекватности является пошаговым процессом и обязательной частью процесса ИМ. Если необходимо внести коррективы в имеющуюся систему, то оценка адекватности приобретает реальные очертания. Так, результаты моделирования сравниваются с накопленными данными, и если совпадение находится в границах заданной статистической точности, то адекватность достаточно высока.

После этого в модель вносятся предполагаемые коррективы, и о проверке адекватности можно не беспокоиться. В случае вновь разрабатываемой системы приходится прибегать к сложным процедурам оценки адекватности, описанной выше.

Глава 2

КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ ДЛЯ GPSS/H

§ 2.1. КЛАССИФИКАЦИЯ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ

Существует большое число аналитических моделей описания процессов функционирования систем, решения задач оптимизации или нахождения рационального решения, принятия решений и т. п. Пусть не обидятся на нас специалисты-математики, которые считают аналитические описания истиной в последней инстанции, но, с позиций системного подхода, в естественных и искусственных сферах природы можно классифицировать любые понятия. Поэтому и аналитические методы по-крупному можно свести в классификационную таблицу аналитических моделей, исходя из двух основополагающих понятий — протяженности во времени и характера возникновения событий (отношения с внешней средой).

Рассмотрим эти понятия подробнее.

Протяженность во времени

Можно рассматривать только два вида протекания какого-либо процесса во времени. Время может рассматриваться либо как непрерывная переменная $t \in [0, T]$, либо как дискретная $t = i\Delta t$, $i = 0, 1, \dots, M$, $M = [T/\Delta]$, где Δ — шаг дискретизации. Соответственно припишем индексы Н и Д этим двум видам процессов, описываемых аналитическими моделями. Индекс Н соответствует аналоговым сигналам (постоянный, монотонный, синусоидальный и т. д.). Индекс Д — дискретным сигналам (импульсный — в виде отдельного импульса или их последовательности; цифровой, подобно 1 и 0 в ЭВМ, и т. п.).

Характер возникновения событий

При отсутствии случайных возмущений со стороны среды и полной определенности поведения компонентов системы имеем постоянную или детерминированную модель, для которой выберем индекс П (во избежание путаницы с ранее введенным индексом Д). В большинстве случаев необходимо учитывать вероятностные характеристики всех воздействий, для этого класса моделей введем индекс В.

С учетом сказанного, классификация позволит рассматривать четыре типа моделей: НП-, ДП-, НВ-, ДВ-модели. Очевидно, что все многообразие аналитических моделей можно разнести по этим клас-

Таблица 2.1. Классификация математических моделей

Характеристика	Тип ММ			
	НП	ДП	ДВ	НВ
Вид зависимости	Дифференциальные и интегральные уравнения	Теория разностных уравнений, конечные автоматы	Разностные стохастические уравнения, вероятностный автомат	Стохастические дифференциальные уравнения, теория массового обслуживания

сам. Обзор аналитических моделей невозможно провести в рамках одного пособия, определенная попытка была сделана в работах [4, 11]. Ограничимся сводной табл. 2.1, которая содержит только некоторые аналитические модели и определяет область их применения.

Данная таблица не претендует на полноту, а является лишь иллюстрацией предлагаемой классификации, и в ней для дальнейшего изложения представляет интерес только часть НВ-моделей — теория массового обслуживания. Именно теория массового обслуживания является концептуальной моделью для GPSS, поэтому в следующих параграфах кратко рассмотрены основные идеи теории МО.

§ 2.2. ОСНОВНЫЕ ОБОЗНАЧЕНИЯ ТЕОРИИ МАССОВОГО ОБСЛУЖИВАНИЯ

Системы массового обслуживания встречаются на каждом шагу — невозможно назвать область человеческой деятельности, где не возникают проблема обслуживания и создание очереди при занятости органа обслуживания (документооборот, телефонная связь, бизнес, торговля и т. д. В гл. 4 приведен список сфер, где можно применять ИМ на GPPS/Н, а значит, где применимы методы СМО).

В теории СМО разработаны идентичные модели для:

- проектирования и эксплуатации систем, состоящих из большого числа аналогичных или похожих, но отличающихся производительностью компонентов (количество персонала, число линий связи, количество таможенных пунктов и т. п.);
- отыскания оптимального количества оборудования (число лифтов в офисном здании, число грузовых терминалов и т. п.);
- определения производительности той или иной системы (производительность ЭВМ, пропускная способность канала связи и т. п.).

Очевидно, что можно назвать дополнительные возможности СМО и заключить, что во многих исследуемых системах сочетаются все указанные выше модели. Более того, теория СМО быстро развивается и появляются все новые модели. В связи с этим ниже будут изложены только основополагающие идеи СМО, не претендующие на какую-либо новизну, являющиеся ориентиром для читателей, не знакомых с теорией СМО, и приводимые для лучшего понимания концептуальных основ ЯИМ GPSS/H.

Систему МО можно описать рядом параметров.

- *Входной поток заявок или требований $v(t)$* (в GPSS/H — транзакты), задающий вероятностный закон поступления заявок на обслуживание. Заявки могут поступать либо по одиночке, либо группами (пакетами). В GPSS/H входной поток задается оператором блока (ОБ) GENERATE (см. гл. 4, 5).

- *Поток обслуживания $u(t)$* , задающий вероятностный закон процесса обслуживания заявок. В GPSS/H поток обслуживания задается ОБ ADVANCE (см. гл. 5).

- *Прибор обслуживания $P_i, i=1, 2, \dots, N$* , состоящий из накопителя N_i емкостью $0 \leq m \leq \infty$, при $m = 0$ происходит потеря обслуживания, а при $m = \infty$ все заявки ожидают обслуживания, промежуточные значения определяют емкость накопителя. В состав прибора также входит канал обслуживания $K = n_j, j = 1, 2, \dots, L$; при $n = 1$ обслуживание называется *одноканальным*, а при $n > 1$ — *многоканальным*.

Если приборы обслуживания соединяются параллельно, то такое обслуживание называется *однофазным*, а если приборы соединяются последовательно, — *многофазным* (ряд последовательных операций).

- *Очередь* — задержка в обслуживании поступающих заявок, характеризующаяся *дисциплиной очереди*, т. е. порядком обслуживания заявок. Можно назвать разные виды дисциплины обслуживания:

FIFO — первый пришел — первый вышел (обслужился), в англоязычной литературе эта известная аббревиатура все чаще заменяется на FCFS (first come first serve — первый пришел — первый обслужился);

LCFS — последний пришел — первый обслужился, эта дисциплина предназначена для заявок с более высоким приоритетом, но используется крайне редко, а чаще используется дисциплина следующего вида;

SPT (shortest processing time) — кратчайшее время обслуживания, которое применяется для заявок с приоритетом, в GPSS/H эта дисциплина реализуется ОБ PRIORITY;

случайная дисциплина, например система опроса слушателей на практических занятиях.

В пособии будут рассматриваться только дисциплины FCFS и SPT. Подчеркнем, что в GPSS/Н факт создания очереди реализуется связанной парой ОБ QUEUE / DEPART.

• *Выходной поток* $y(t)$ — функция распределения, представляющая собой сумму двух вероятностных законов: $y_1(t)$ — поток обслуженных заявок и $y_2(t)$ — поток потерянных (не обслуженных) заявок, который образуется за счет отказа в обслуживании из-за малого объема накопителя по принципу $m + 1 < K$, где K — число заявок на входе прибора. В отдельных случаях заявка может остаться в приборе из-за окончания времени моделирования, поэтому в неравенстве появляется единица. Все сказанное объединим в рис. 2.1.

Таким образом, однофазные (простые) СМО могут быть либо одноканальными, либо многоканальными, многофазные СМО, представляющие последовательность различных операций, выполняемых различными приборами обслуживания, могут представлять собой комбинацию одно- и многоканальных СМО.

В 1953 г. Г. Кендалл предложил стандартные обозначения для введенных выше определений, которые и используются исследователями без изменений. Для однофазных СМО символика Кендалла выглядит следующим образом:

$$A / B / n / m,$$

где A и B — входной поток и поток обслуживания соответственно; n — число каналов, $n \leq 1$; m — емкость накопителя.

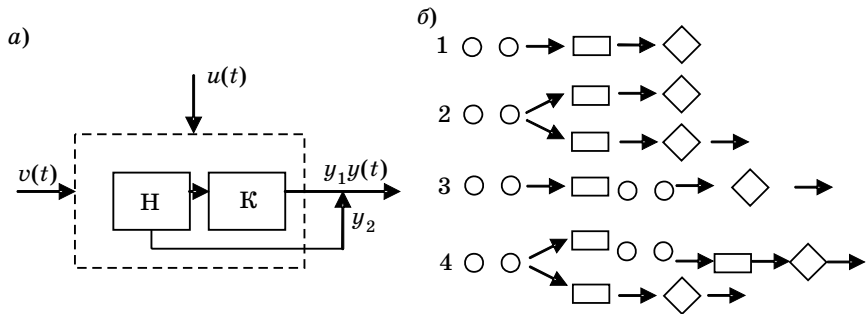


Рис. 2.1. Блок-схема элемента СМО: а — прибор обслуживания; б — виды СМО:

1 — одноканальная однофазная; 2 — многоканальная однофазная; 3 — одноканальная многофазная; 4 — многоканальная многофазная; Н — накопитель; К — канал обслуживания; ○ — поток входных заявок; □ — прибор обслуживания; ◇ — поток обслуженных заявок

Потоки случайных событий могут иметь различный вид:

M — экспоненциальное распределение длительностей интервалов поступления заявок или длительностей обслуживания (индекс M — от определяющего слова *марковский процесс*, т. е. такой, когда поведение процесса после момента времени t зависит лишь от состояния процесса в момент времени t и не зависит от поведения до момента времени t);

D — детерминированное распределение длительностей интервалов поступления заявок или длительностей обслуживания;

E_k — поток Эрланга k -го порядка для длительностей интервалов между приходами заявок или длительностей обслуживания;

GI — рекуррентный поток (длительности интервалов статистически независимы и имеют одинаковое распределение);

G — общий вид распределения.

Тогда в символах Кендалла вместо A и B подставляется символ одного из упомянутых потоков, например:

$M/M/1$ — экспоненциальные потоки с одним каналом обслуживания и неограниченной емкостью;

$D/GI/5/10$ — детерминированный входной поток, рекуррентный поток обслуживания, многоканальное СМО с пятью одинаковыми каналами, емкость накопителя 10 и т. д.

Описание любого варианта СМО на языке математики несложно, но практически малоценно, так как не отражает динамики процесса функционирования системы. Поэтому всегда существует необходимость, получив предварительные числовые ориентиры на основе аналитической модели, далее строить имитационную модель, для чего незаменим ЯИМ GPSS/H.

§ 2.3. НЕКОТОРЫЕ АНАЛИТИЧЕСКИЕ МОДЕЛИ СИСТЕМЫ МАССОВОГО ОБСЛУЖИВАНИЯ

Наиболее сложный случай оценки параметров потоков заявок и обслуживания — когда потоки являются *простейшими*. Простейшим называется такой поток, при котором время прихода заявок удовлетворяет условиям стационарности, отсутствия последствия и ординарности.

Стационарность случайного процесса означает, что на любом промежутке времени Δt вероятность прихода n заявок зависит только от числа n и величины промежутка Δt , но не изменяется от сдвига Δt по оси времени. При этом выполняется эргодическое свойство — статистическое равенство n заявок, полученных при ИМ одной системы,

или испытания n систем до прихода первой заявки. Формулирование этого свойства звучит достаточно просто: «Совокупное по времени наблюдений равняется совокупному по ансамблю наблюдений».

Отсутствие последствий означает, что вероятность прихода n заявок в течение промежутка времени Δt не зависит от того, сколько пришло заявок до этого момента времени. Выполнение этого условия гарантирует случайность и независимость событий.

Ординарность потока заявок означает невозможность появления более одной заявки в один и тот же момент времени. Как это реализуется в GPSS/H, будет рассмотрено в ч. 2.

Не будем рассматривать причины, приводящие к искажению сформулированных свойств простейшего потока, отметим только, что интервалы времени прихода заявок подчиняются в этом случае функции распределения Пуассона, а время обслуживания описывается экспоненциальной функцией распределения. Любые другие функции распределения приведут к лучшим параметрам потоков, поэтому считается, что если параметры СМО удовлетворяют условиям простейшего потока, то они наверняка обеспечат удовлетворительную работу СМО при всех других потоках. В связи с этим в рассматриваемых далее моделях используются функции распределения Пуассона и экспоненциальные.

2.3.1. Распределение вероятностей длительности интервалов между заявками

Пусть $f(t)$ — плотность распределения длительностей t интервалов между любой парой смежных заявок. Определим параметр потока λ как среднюю частоту появлений заявок, а $1/\lambda$ — как среднее значение длительности интервала, тогда

$$\int_0^{\infty} t f(t) dt = 1/\lambda. \quad (2.1)$$

Например, если за дискрету времени примем 1 ч, а $\lambda = 4$, то среднее количество поступлений равняется 15 мин ($1/\lambda = 0,25$) и наоборот, если каждые 10 мин в систему поступает одна заявка, то частота поступлений λ равняется 0,1 заявки в минуту.

Для стационарного потока плотность определяется как

$$f(t) = \lambda e^{-\lambda t}, \quad t \geq 0, \quad (2.2)$$

такое распределение называется экспоненциальным.

Вычисляя вероятность попадания n заявок в произвольно выбранный интервал T , приходим к распределению Пуассона:

$$P_n(t) = ((\lambda t)^n / n!) e^{-\lambda t}, n = 0, 1, 2, \dots \quad (2.3)$$

Полученные распределения отвечают всем свойствам простейшего потока, а именно:

— длительность интервалов взаимонезависима и одинаково распределена;

— ненулевая вероятность поступления заявок при $\Delta t > 0$;

— при интервале Δt , стремящемся к нулю, количество поступающих заявок не превышает единицы.

Впредь будем полагать, что отсчет времени начинается с момента $T \geq 0$.

Нетрудно показать, что экспоненциальная функция распределения заявок и пуассоновский процесс обладают одинаковыми статистиками, и их можно считать синонимами, поэтому и принято обозначение марковский процесс или М-процесс.

2.3.2. Распределение вероятностей длительностей обслуживания

Будем считать, что каждый канал в одно и то же время может обслуживать только одну заявку. Следующие друг за другом интервалы обслуживания независимы и имеют идентичное распределение.

Пусть плотность распределения равна $g(t)$, тогда среднее время обслуживания

$$T_0 = \int_0^{\infty} t g(t) dt = 1/\mu, \quad (2.4)$$

где μ — параметр (скорость) потока обслуживания.

Так, например, если за дискрету времени принять 1 ч, а $\mu = 5$, то в течение часа прибор обслужит 5 требований и среднее время обслуживания равно 12 мин и наоборот; если на обслуживание заявки уходит 30 мин, то скорость обслуживания $\mu = 2$. При расчете среднего времени обслуживания учитывается только время занятости прибора обслуживания.

Для получения верхней границы пропускной способности канала обычно полагают, что распределение длительностей обслуживания является экспоненциальным:

$$g(t) = \mu e^{-\mu t} \text{ при } t \geq 0, \quad (2.5)$$

при этом вероятность завершения обслуживания в интервале $(t + \Delta t)$ не зависит от того, сколько времени уже потрачено на обслуживание этой заявки (пример системы, не обладающей памятью). Таким образом, если в момент t заявка уже обслуживалась, то в силу (2.5) в момент $(t + \Delta t)$ вероятность того, что в этом интервале обслуживание не заканчивается:

$$P(t + \Delta t) \approx e^{-\mu\Delta t}. \quad (2.6)$$

Следовательно, при очень малых Δt вероятность того, что обслуживание в рассматриваемом интервале не заканчивается:

$$P(t + \Delta t) \approx 1 - \mu\Delta t, \quad (2.7)$$

а что заканчивается:

$$P(t + \Delta t) \approx \mu\Delta t. \quad (2.8)$$

2.3.3. Одноканальное обслуживание с пуассоновским входным потоком и экспоненциальным распределением длительностей обслуживания

Рассмотрим пример, в котором имеется возможность аналитического определения показателей эффективности функционирования СМО (М/М/1).

Пусть процесс обслуживания начинается при отсутствии заявок в накопителе, тогда состояние СМО описывается следующей системой уравнений:

$$\begin{aligned} P_n(t + \Delta t) &= P_n(t)(1 - (\lambda + \mu)\Delta t) + P_{n-1}(t)\lambda\Delta t + P_{n+1}(t)\mu\Delta t, \quad n \geq 1; \\ P_0(t + \Delta t) &= P_0(t)(1 - \lambda\Delta t) + P_1(t)\mu\Delta t, \end{aligned} \quad (2.9)$$

где $P_n(t)$ — вероятность нахождения системы в состоянии $x_n(t) \in X$ в момент t , т. е. когда в ней находятся n заявок.

Вероятность нахождения в системе n заявок в момент времени $(t + \Delta t)$ равна сумме трех вероятностей:

1) вероятности нахождения в системе n заявок в момент t , умноженной на вероятность того, что за время Δt в систему не поступает ни одной заявки и ни одна заявка не будет обслужена;

2) вероятности нахождения в системе $(n - 1)$ заявки в момент t , умноженной на вероятность поступления одной заявки за время Δt , и ни одна заявка не будет обслужена;

3) вероятности нахождения в системе $(n + 1)$ заявки в момент $t + 0$, где 0 — пренебрежимо малый интервал времени, умноженной на вероятность ухода одной заявки, при условии непоступления ни одной заявки.

Заметим, что

$$(1 - \lambda\Delta t)(1 - \mu\Delta t) = 1 - \lambda\Delta t - \mu\Delta t + o(\Delta t);$$

$$\lambda\Delta t(1 - \mu\Delta t) = \lambda\Delta t + o(\Delta t), \quad \mu\Delta t(1 - \lambda\Delta t) = \mu\Delta t + o(\Delta t). \quad (2.10)$$

Образуя разностное уравнение и переходя к пределу, получаем дифференциальные уравнения:

$$\begin{cases} \frac{dP_n(t)}{dt} = -(\lambda + \mu)P_n(t) + \lambda P_{n-1}(t) + \mu P_{n+1}(t), & t \geq 1; \\ \frac{dP_0(t)}{dt} = -\lambda P_0(t) + \mu P_1(t). \end{cases} \quad (2.11)$$

Найдем выражение среднего числа заявок, находящихся в накопителе, и среднего времени ожидания заявок в накопителе для стационарного состояния при загрузке $\rho = \lambda / \mu < 1$, коэффициент ρ иногда называют трафик-интенсивностью. Поскольку он также представляет долю полного времени, в течение которого прибор не простаивает, его иногда называют коэффициентом использования или загруженности.

Приравняв производные по времени t к нулю, получим уравнения:

$$\begin{cases} (\lambda + \mu)p_n = \lambda p_{n-1} + \mu p_{n+1}, & n \geq 1; \\ \lambda p_0 = \mu p_1; \end{cases} \quad \begin{cases} (1 + \rho)p_n = p_{n-1} + \rho p_{n+1}, & n \geq 1; \\ p_1 = \rho p_0. \end{cases} \quad (2.12)$$

Положим $n = 1$, тогда $(1 + \rho)p_1 = p_2 + \rho p_0$, $p_2 = \rho^2 p_0$, повторяя эти операции, имеем $p_n = \rho^n p_0$, причем:

$$\sum_{n=0}^{\infty} p_n = 1 \Rightarrow \sum_{n=0}^{\infty} \rho^n p_0 = 1 \Rightarrow p_0 \sum_{n=0}^{\infty} \rho^n = 1 \Rightarrow p_0 = 1 - \rho.$$

Следовательно, получим, что $p_n = \rho^n (1 - \rho)$ — геометрическое распределение. Среднее число заявок в системе равно

$$E\{x_n\} = l_n^S = \sum_{n=1}^{\infty} n p_n = (1 - \rho) \sum_{n=1}^{\infty} n \rho^n = \rho / (1 - \rho), \quad (2.13)$$

$$D\{x_n\} = \sum_{n=1}^{\infty} (n - l_n^S)^2 p_n = \sum_{n=1}^{\infty} n^2 p_n - (\rho/(1-\rho))^2 = \rho/(1-\rho) + 2\rho^2/(1-\rho)^2.$$

Среднее число заявок, находящихся в накопителе:

$$l_n^H = \sum_{n=1}^{\infty} (n-1)p_n = \rho^2/(1-\rho). \quad (2.14)$$

Среднее время ожидания заявок в накопителе:

$$E\{t_n\} = \frac{l_n^i}{\lambda} = \rho^2/(\lambda(1-\rho)). \quad (2.15)$$

Среднее время пребывания в системе

$$E\{w\} = 1/\mu(1-\rho) = 1/\mu - \lambda. \quad (2.16)$$

Сведем основные операционные характеристики рассматриваемой системы с дисциплиной FCFS (FIFO) в табл. 2.2. Следует обратить внимание, что при возрастании коэффициента использования такие параметры как число заявок в системе, длина очереди, время пребывания в системе начинают быстро возрастать. При заданной скорости обслуживания μ , когда коэффициент занятости невелик, основная доля среднего времени пребывания заявки в системе связана только с процедурой обслуживания, при возрастании интенсивности входного потока большая часть времени пребывания заявки в системе обусловлена ожиданием обслуживания.

Таблица 2.2. Операционные характеристики СМО

ρ	$1 - \rho$	$\rho/(1 - \rho)$	$\rho^2/(1 - \rho)$	$\mu = 10$			$\mu = 20$		
				λ	E_w	E_{tn}	λ	E_w	E_{tn}
0.1	0.911	10.11	0.011	111	110.11	0.01	12111	0.06	0.01
0.3	0.711	10.43	10.1311	113	110.14	0.04	16111	0.07	0.02
0.5	0.511	11.01	10.5111	115	10.2	0.11	10111	0.11	0.05
0.7	0.311	12.33	1.631	117	110.33	0.23	14111	0.17	0.12
0.8	0.211	4.0	3.211	118	10.5	0.41	16111	0.25	0.21
0.9	0.111	9.0	8.111	119	11	0.91	18111	0.51	0.45
10.95	0.051	19.01	18.0511	9.5	21	1.91	19111	111	0.95
10.99	0.011	99.01	98.011	9.9	1011	9.91	19.81	511	4.95
0.999	0.001	999.0	998.01	119.99	100	99.911	19.98	50111	49.951

Рассмотрим конкретный пример использования табл. 2.2. Пусть дискрета времени равна 1 ч, а $\rho = 0,8$, при этом прибор простаивает в среднем 0,2 ч (12 мин), а среднее количество требований в системе равно 4. При $\mu = 10$ (скорость обслуживания равняется 10 ед./ч) средняя продолжительность пребывания заявки в системе равняется 0,5 (30 мин), а пребывание в очереди из них занимает 24 мин.

Возможны ситуации, когда длина очереди ограничена. Если в СМО не может быть более L заявок, то длина очереди ограничена величиной $L - 1$ и любая заявка сверх этого значения теряется и статистическое равновесие в этом случае достигается при любом значении ρ [1].

2.3.4. Многоканальное обслуживание с пуассоновским входным потоком и экспоненциальным распределением длительностей обслуживания

Очевидно, что в большинстве случаев СМО являются многоканальными, символика Кендалла в этом случае имеет вид $M/M/L$, где входной поток имеет вид $f(t) = \lambda e^{-\lambda t}$, поток обслуживаний $g(t) = \mu e^{-\mu t}$, а число каналов равно L . Учтем, что режим функционирования любого канала не влияет на режимы функционирования других каналов, длительности обслуживания являются случайными величинами. Уравнения в конечных разностях аналогичны уравнениям (2.11). Решение системы уравнений приводится без доказательств:

$$P_n = \rho^n / n! P_0 \text{ при } L > n \geq 0; \quad (2.17)$$

$$P_n = \rho^n / L! L^{n-L} P_0 \text{ при } n \geq L, \quad (2.18)$$

где $\rho = \mu / \lambda$;

$$P_0 = \frac{1}{\sum_{j=0}^{L-1} \frac{\rho^j}{j!} + \frac{\rho^L}{L!(1-\rho/L)}}. \quad (2.19)$$

Установившийся режим функционирования СМО, определяемой выражениями (2.17) – (2.19), достигается при условии $\lambda < \mu L$. Когда P_n определено, то большинство операционных характеристик вычисляется с помощью элементарных алгебраических операций [1].

Глава 3 ПРИНЦИПЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

§ 3.1. ОБЩИЕ ПРЕДСТАВЛЕНИЯ

В настоящей главе будут рассмотрены некоторые основополагающие принципы построения любой имитационной модели, не обязательно реализуемой с помощью ЯИМ GPSS/Н. В табл. 3.1 в качестве иллюстративного примера приведена классификация некоторых известных ЯИМ [4].

С появлением персональных ЭВМ интенсивно развиваются специализированные системы моделирования для этих типов компьютеров. Часто они представляют собой развитие известных систем моделирования для универсальных ЭВМ, например: GPSS, SIMSCRIPT II.5, SIMULAP, MiniDYNAMO, MicroDYNAMO/DOS, SLAM II/PC. В книге Е. Киндлера «Языки программирования» приведен каталог наиболее известных зарубежных систем ИМ, включающий 200 наименований. Некоторые из этих систем вошли в табл. 3.2. В таблице указаны, в основном, системы, предназначенные для использования на IBM-совместимых универсальных (MF) и персональных (PC) ЭВМ. Дополнительно указываются требуемый транслятор (с базового языка программирования), операционная система, а также некоторые

Таблица 3.1. Классификация ЯИМ по способу имитации

Тип модели	Способ имитации	Примеры ЯИМ
Дискретный	Событийный Просмотр активностей Процесный Транзактный	SIMSCRIPT SMPL, FORSIM SIMPULA GPSS
Агрегативный	Агрегатный	АИС, САПАС
Непрерывный	Аппроксимация дифференциальных уравнений	DYNAMO, MIMIC
Непрерывно-дискретный	Комбинированный (возможны различные комбинации вышперечисленных способов имитации)	SLAM, НЕДИС
Сетевой	Имитации сетевых моделей	SIMMET, NETWORK

Таблица 3.2. Классификация ЯИМ

Название	Тип ЭВМ, транслятор	Примечания
Дискретные модели		
GPSS/PC	PC	Интерактивная графика и анимация Версия системы GPSS для ПЭВМ
GPSS/H	PC	
SIMSCRIPT II.5	PC, MF	Графика и анимация
SIMULA	PC, MF	
SIMULAP	PC, MS DOS, Windows	
MIC-SIM	PC, PS/2	
OPTIC	PC	
PC-SOL 4.0	PC, Паскаль 4.0	
Непрерывные модели		
DYNAMO III /F+/ 370	MF, Фортран	Нелинейные модели
MicroDYNAMO	PC, 16-разрядные, PC	
MiniDYNAMO	PC	Дополнительные возможности оптимизации моделируемых систем
ProDYNAMO+		
CSSL-IV		
ISIM		
OPT ISIM		
SYSL/M		
ACSL	PC, Фортран	
Сетевые модели		
SIMNET	PC, Фортран-77	—
Network II.5	PC, MF, VAX	
BEST-NETWORK	MF, Фортран	
Непрерывно-дискретные модели		
SLAM II	MF, Фортран	Комбинированные модели дискретного, непрерывного и сетевого типа
SLAM II/PC	PC	
MicroPASSIM	PC, Паскаль	Расширенные возможности статистического анализа; ориентация на производственные системы; интерактивная графика и анимация
PASION	PC, Паскаль	
SIMAN + Cinema	PC, Фортран-77	
Динамические системы (ДС)		
ENPORT-7	MF, Фортран	Нелинейные ДС
TSIM	VAX	
LSMP	PC	Линейные ДС
DSL/VS	MF, Фортран, Фортран-77, ПЛ /1	

Продолжение табл. 3.2

Название	Тип ЭВМ, транслятор	Примечания
Системы реального времени		
NET Real Time	PC	Ориентация на моделирование автоматизированных производств Непрерывные модели; интерактивная графика
DATE-INTER-ACTIVE	VAX	
Системы, ориентированные на производственные процессы и контроль качества		
GEMS-II MAST SIMPLE SPAR	PC, Фортран-77 PC, Фортран-77 PC PC	Сетевые модели Анализ производственных процессов Оценка возможности роста производства Быстрое построение прототипов производственных систем средствами интерактивной графики Моделирование производственных и материальных потоков Моделирование систем управления качеством продукции
XCELL+	PC	
SIMIS III	PC, MF	
Process Quality Simulator	PC	
Системы моделирования экономических процессов		
LIBRA	PC	Построение моделей равновесия экономических систем Многоуровневые модели рынка
MULTISIM	DEC-20, Simula	
Моделирование вычислительных и коммутационных систем (КС)		
CIRCUITS	Macintosh	Моделирование мощных вычислительных систем; интерактивная графика
Моделирование робототехнических систем		
ROSCAD ROSY	Silicon, Sun VAX	Графическое моделирование в реальном времени

Название	Тип ЭВМ, транслятор	Примечания
Системы статистического моделирования и обработки данных		
СТАТМОД	РС	Статистическое моделирование случайных величин, векторов, процессов
СТАН	РС	Статистический анализ результатов экспериментов

особенности или возможности систем (в графе «Примечания»). По своему назначению системы разбиты на разделы. Первые четыре раздела включают ЯИМ общего назначения, ориентированные на определенный тип модели. В последующих разделах системы объединены по типу предметной области либо условиям применения.

Таблицы 3.1, 3.2 дают общее представление о системном мире ЯИМ, однако существуют принципы, одинаковые для всех типов моделей:

- понятие о модельном времени;
- структура процесса и способы представления параллельно происходящих в реальной системе событий в виде квазипараллельных событий при реализации на компьютере;
- генерирование базовых случайных величин (чисел, векторов, процессов).

Все упомянутые принципы будут рассмотрены в последующих параграфах настоящей главы.

§ 3.2. МОДЕЛЬНОЕ ВРЕМЯ

Как уже упоминалось (§ 1.5), содержание любой задачи ИМ получается в результате реализации прогона или ряда прогонов модельного файла, получения и обработки собранной статистики и принятия решений на основе статистического анализа. Длительность испытаний зависит либо от заданной статистической точности, либо от заданного числа реплик в одном прогоне, либо от времени рассмотрения функционирования реальной системы (см. гл. 8). В связи со сказанным необходимо четко понимать, какие времена рассматриваются при ИМ, и представлять их различие.

- T_p — реальное время функционирования исследуемой системы S , которое может быть очень большим, например, 10^n лет при иссле-

довании космогонических процессов, либо, наоборот, очень малым — 10^{-n} с при исследовании процессов, происходящих в микромире.

- $T_{\text{и}}$ — машинное время имитации, отражающее затраты ресурса времени ЭВМ на организацию ИМ. В случае использования суперкомпьютеров, производительность которых превышает сотни гигафлоп, минута машинного времени может стоить несколько тысяч долларов. Это ограничение должно учитываться создателями ИМ и далее не рассматривается.

- $T_{\text{м}}$ — модельное время (МВ), используемое в ИМ. Оно может быть сжато при исследовании процессов макромира или растянуто при оперировании со сверхбыстрыми процессами в реальной системе. Кроме того, именно МВ позволяет избежать сложностей моделирования поведения реальной системы. Так, в реальной системе события могут происходить одновременно в разных компонентах системы; в обычных, не мультипроцессорных, ЭВМ параллельные события воплотить нельзя. Модельное время позволяет синхронизовать все события и реализовать квазипараллелизм (см. § 3.2). При создании ИМ задание временной дискреты модельного времени является обязательным условием до начала процесса ИМ. *Естественно, что разные значения времени процессов обязательно должны быть выражены в едином масштабе временной дискреты модельного времени.* Так, например, если временная дискрета задана в минутах, другие временные периоды: часы, сутки, годы — должны быть также представлены в минутах.

Введем обозначение временного интервала моделирования системы S (интервала МВ)

$$\tau = [t_0, T_i],$$

где t_0 — время начала моделирования (обычно полагают $t_0 = 0$); $T_{\text{м}}$ — время окончания моделирования; $t \in \tau$ — текущее значение МВ.

Построение модели системы S начинается с определения параметров системы и переменных, определяющих процесс функционирования системы.

1. **Параметры системы** $\theta_1, \theta_2, \dots, \theta_m$ — это характеристики системы, остающиеся постоянными на всем интервале моделирования τ . Если значения $\{\theta_i\}$ определены на некотором множестве Θ

$$\theta = (\theta_1 \dots \theta_m) \in \Theta \subset R^m,$$

то говорят, что имеется параметрическое семейство систем.

2. **Множество переменных** разбивают на два подмножества — независимых и зависимых переменных.

К *независимым* переменным отнесем следующие характеристики.

- Входные воздействия на систему (сигналы) u_1, u_2, \dots, u_{n_1} . Входные воздействия в момент $t \in T$ характеризуются вектором

$$\mathbf{u} = \mathbf{u}(t) = (u_1(t), \dots, u_{n_1}(t)) \in U \subset R^{n_1}.$$

Среди $\{u_i\}$ могут быть управляющие воздействия, например $u_1, u_2, \dots, u_{n_1'}$ ($n_1' \leq n_1$), а остальные $n_1 - n_1'$ воздействий — неуправляющие.

- Воздействия внешней среды: среди них могут быть контролируемые (наблюдаемые) и неконтролируемые (ненаблюдаемые), детерминированные и случайные воздействия. В момент $t \in T$ они характеризуются вектором

$$\mathbf{v} = \mathbf{v}(t) = (v_1(t), \dots, v_{n_2}(t)) \in V \subset R^{n_2}.$$

- Переменные, характеризующие состояние системы: x_1, x_2, \dots, x_{n_3} . В отличие от $\{\theta_i\}$ состояния $\{x_i\}$ характеризуют свойства системы, изменяющиеся во времени. Состояние системы в момент времени $t \in T$ описывается вектором

$$\mathbf{x} = \mathbf{x}(t) = (x_1(t), \dots, x_{n_3}(t)) \in X \subset R^{n_3},$$

где X — пространство состояний или фазовое пространство системы (множество возможных значений вектора \mathbf{x}). Если $t_1 < t_2 < \dots$ — моменты изменения состояния системы, то последовательность $\mathbf{x}(t_1), \mathbf{x}(t_2), \dots$ называется фазовой траекторией системы.

К *зависимым* переменным отнесем следующие характеристики.

- Выходные характеристики (сигналы) системы y_1, y_2, \dots, y_{n_4} , определяемые в момент времени $t \in T$ вектором

$$\mathbf{y} = \mathbf{y}(t) = (y_1(t), \dots, y_{n_4}(t)) \in Y \subset R^{n_4}.$$

- Выходные показатели системы q_1, q_2, \dots, q_k характеризуют ее цели (т. е. характеризуют достижение системой заданных или оптимальных величин) и образуют вектор

$$\mathbf{q} = \mathbf{q}(t) = (q_1(t), \dots, q_k(t)) \in Q \subset R^k, t \in \tau.$$

Выходные показатели представляются в виде некоторых функций

$$q = q(t) = q_1(t) = (q_1(t), \dots, q_k(t)) \in Q \subset R^k, t \in \tau.$$

При наличии в системе случайных факторов (например, случайных воздействий внешней среды) значения $\{q_i\}$ являются также слу-

чайными, при этом в качестве Q используют средние значения $\{Q_i\}$, определяемые соотношениями

$$Q_i = E\{q_i\}, \quad i = \overline{1, k},$$

где $E\{\cdot\}$ — символ математического ожидания.

Связи между зависимыми и независимыми переменными показаны на рис. 3.1.

Процесс функционирования системы во времени описывается *операторными соотношениями* (заданными аналитически или алгоритмически) для состояний и выходных характеристик системы:

$$\begin{aligned} x(t) &= F_1(u^{(t)}, v^{(t)}, \theta, t); \\ y(t) &= F_2(u^{(t)}, v^{(t)}, x^{(t)}, \theta, t); \\ q(t) &= F_3(u^{(t)}, v^{(t)}, x^{(t)}, y^{(t)}, \theta, t), \quad t \in \tau, \end{aligned} \quad (3.1)$$

где $u^{(t)}$ обозначают реализацию процесса $u(t)$ на отрезке $[0, t]$, аналогично обозначены $x^{(t)}, y^{(t)}$; $F_1(\cdot), F_2(\cdot), F_3(\cdot)$ обозначают соответствующие операторы, описывающие динамику зависимых и независимых переменных и показателей эффективности.

Зависимости (3.1) называются *законами функционирования системы S*; зависимость $y = y(t), t \in \tau$ называется *выходной траекторией* системы, а зависимость $x = x(t), t \in \tau$ — *фазовой траекторией*.

Под фазовой траекторией будем понимать отображение движения системы из одного состояния фазового пространства в другое в каждый момент МВ. В общем случае фазовое пространство при N компонентах системы S характеризуется $3N$ координатами и $3N$ импульсами, т. е. имеет размерность $6N$, что не изобразить графически. Простейшим случаем фазового пространства является *фазовая плоскость*.

Например, для маятника с трением имеем уравнение

$$d^2x/dt^2 + \eta/m dx/dt + k/mx = 0, \quad (3.2)$$

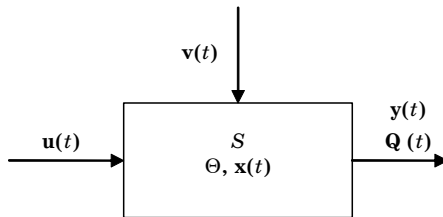


Рис. 3.1. Связь между переменными

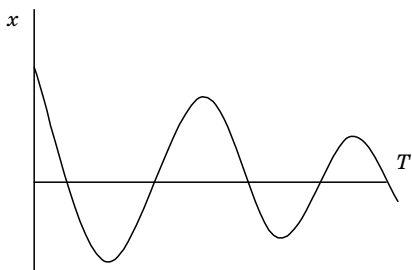


Рис. 3.2. Зависимость координаты

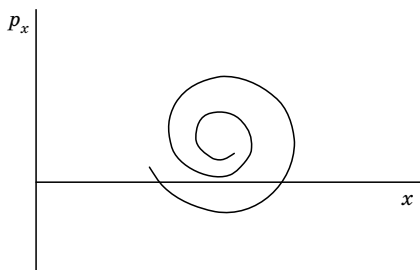


Рис. 3.3. Фазовая траектория

где x — координата; η — сила вязкого трения; m — масса маятника; k — жесткость.

Решением этого уравнения являются затухающие колебания. Зависимость координаты от времени приведена на рис. 3.2, а на рис. 3.3 — фазовая траектория в виде спирали, имеющая *аттрактором* (устойчивым состоянием) фокус в нуле.

Сделаем важное замечание, которое используется при моделировании: процесс моделирования всегда начинается в момент времени 0.0 или после точки 0.0. Если операндом С ОБ GENERATE задано смещение, то начало координат перемещается из точки 0.0 в точку, заданную операндом С.

Прежде чем дать описание способов представления модельного времени, рассмотрим взаимосвязь таких компонентов ИМ, как *событие* $A_j^{(i)}$, *действие* $d_j^{(i)}$, *локальное модельное время* $\tau_j^{(i)}$ (ЛМВ) и *процесс*. Под событием будем понимать изменение фазового состояния системы при совершении каких-то действий в течение локального момента времени.

В реальной системе процесс функционирования происходит в реальном времени, в модели — в течение МВ T_m . На рис. 3.4 показана взаимосвязь компонентов ИМ [4].

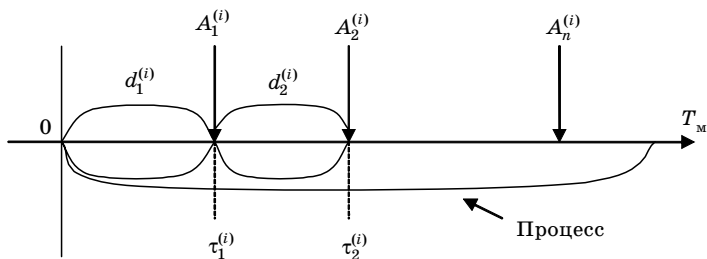


Рис. 3.4. Взаимосвязь компонентов ИМ

Очевидно, что возможны три способа изменения вектора состояний $x(t)$ системы S :

- 1) в моменты наступления событий $\{A_j^{(i)}\}$;
- 2) в результате выполнения действий $\{d_j^{(i)}\}$, на что требуются затраты МВ $\{\tau_j^{(i)}\}$. Пара $\{d_j^{(i)}; \tau_j^{(i)}\}$ называется (i, j) -активностью (Р. Шеннон определил это сочетание как «работа»);
- 3) в результате выполнения хронологической последовательности событий и действий, называемых процессом.

Для осуществления изменения состояния системы одним из указанных способов необходимо каким-то образом управлять изменением МВ.

Существуют два способа формирования конечного множества моментов времени T_M , известных как принципы организации изменения МВ Δt и Δx .

Принцип Δt заключается в изменении МВ с фиксированным шагом Δt .

Принцип Δx заключается в изменении МВ при скачкообразном изменении вектора состояния x системы S на некоторую величину Δx ($\Delta x \neq 0$).

Для моментов времени t^* из множества T_M , сформированного по принципу Δx , справедливо

$$x(t^* + 0) = x(t^*) + \Delta x, t^* \in T. \quad (3.3)$$

Для моментов времени из множества $[0, T] \setminus T_M$ вектор состояний изменяется непрерывно (либо остается неизменным).

Заметим, что скачкообразные изменения состояния системы S происходят при наступлении таких «особых» событий, как поступление управляющих сигналов и внешних воздействий, выдача выходных сигналов и т. п.

Приведем более строгое описание принципов Δt и Δx и поясним их особенности. Пусть система S состоит из N элементов: $A^{(1)}, \dots, A^{(N)}$, поведение которых предполагается моделировать:

$$S = \{A^{(1)}, \dots, A^{(N)}\}.$$

Для каждого элемента $A^{(i)} \in S$ ($i = 1, \dots, N$) определим ЛМВ $t^{(i)} \in [0, T_M]$. Поведение элемента $A^{(i)} \in S$ в течение интервала моделирования определяется некоторой последовательностью действий

$$g_1^{(i)}, g_2^{(i)}, \dots, g_M^{(i)}, g_j^{(i)} \in G, j = 1, \dots, M_j,$$

где G — множество всевозможных действий для элементов S . На множестве G будем выделять подмножество действий D : $D \subset G$, для вы-

полнения которых в ИМ требуется некоторое ненулевое модельное время.

Будем обозначать такие действия $d_1^{(i)}, \dots, d_{m_i}^{(i)}$ ($d_j^{(i)} \in D \subset G$, $j = \overline{1, m_i}$, $m_i \leq M_i$, $i = \overline{1, N}$), а интервалы МВ, затрачиваемые на выполнение этих действий, соответственно, $\tau_1^{(i)}, \dots, \tau_{m_i}^{(i)}$. Последовательность $\{\tau_j^{(i)}\}$ ($j = \overline{1, m_i}$) является последовательностью случайных величин с заданными законами распределения $L\{\tau_j^{(i)}\}$, $i = \overline{1, N}$.

Действия $\{d_j^{(i)}\} \subset D$ приводят к наступлению в системе S *особых событий* $\{A_j^{(i)}\}$. События $\{C_j^{(i)}\}$, к которым приводят действия $\{g_j^{(i)}\} : \{g_j^{(i)}\} \in G/D$, не требующие затрат МВ, считаются *неособыми*.

Момент ЛМВ наступления события $A_j^{(i)}$ для $A_i \in S$ определяется по формуле

$$t_j^{(i)} = t + \tau_j^{(i)}, \quad j = 1, 2, \dots, i = 1, \dots, N, \quad (3.4)$$

где t — текущее значение МВ; $\tau_j^{(i)}$ имитируется на ЭВМ в соответствии с законом распределения $L\{\tau_j^{(i)}\}$.

Состояние системы S в момент времени $t \in [0, T_M]$ определяется вектором состояния $\mathbf{x}(t) \in X \subset R^n$. Состояния системы в моменты наступления особых событий будем называть *особыми состояниями*, а состояние $\mathbf{x}(0)$ — *начальным состоянием* системы.

Для иллюстрации принципов Δt и Δx используем временную диаграмму (рис. 3.5).

Описание временной диаграммы.

Пусть число моделируемых элементов в S равно 2, т. е. $N = 2$, и $S = \{A^{(1)}, A^{(2)}\}$.

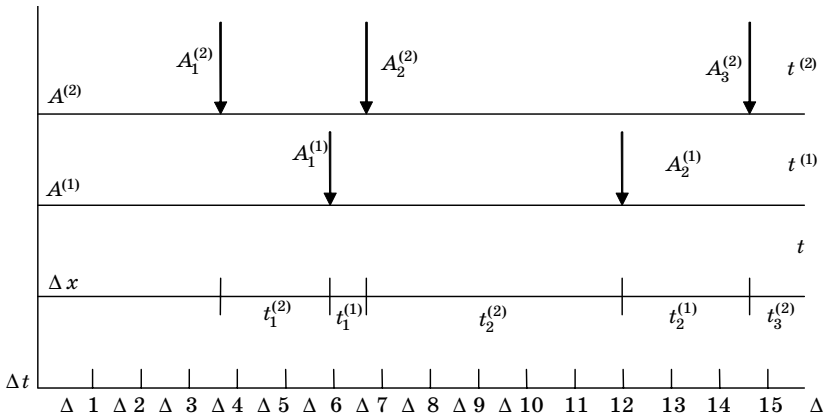


Рис. 3.5. Временная диаграмма

Временная диаграмма включает:

- временную ось ЛМВ $t^{(1)}$ для элемента $A^{(1)}$;
- временную ось ЛМВ $t^{(2)}$ для элемента $A^{(2)}$;
- временную ось модельного времени по принципу Δx ;
- временную ось модельного времени по принципу Δt .

Временные оси будем помечать символами $A^{(1)}, A^{(2)}, \Delta t, \Delta x$.

Пусть в течение рассматриваемого интервала моделирования $[0, T_m]$ для элемента $A^{(1)}$ произошло 2 события: $A_1^{(1)}, A_2^{(1)}$ в моменты $t_1^{(1)}, t_2^{(1)}$, для элемента $A^{(2)}$ — 3 события: $A_1^{(2)}, A_2^{(2)}, A_3^{(2)}$ в моменты $t_1^{(2)}, t_2^{(2)}, t_3^{(2)}$.

Предположим, что хронологическая последовательность событий такова:

$$0 \leq t_1^{(2)} < t_1^{(1)} < t_2^{(2)} < t_2^{(1)} < t_3^{(2)} \leq T.$$

Принцип Δt .

В соответствии с принципом Δt изменение модельного времени t происходит через промежутки времени, равные Δt , т. е. t в течение времени моделирования T принимает конечное множество значений:

$$T_a = \{0, \Delta t, 2\Delta t, \dots, v\Delta t = T\}.$$

При этом событиям, которые попадают в интервал постоянства МВ $\delta_r = ((r-1)\Delta t, r\Delta t)$, $r = 1, v$, в ИМ присваивается один и тот же момент наступления: $t = r\Delta t$. Выбор величины Δt существенно влияет как на быстродействие ИМ, так и на точность аппроксимации системы S с помощью ИМ. Пусть Δt выбран таким, как указано на диаграмме (см. рис. 3.5), т. е. моменты наступления событий в S принадлежат следующим интервалам:

$$t_1^{(1)}, t_2^{(2)} \in \delta_6, t_2^{(1)} \in \delta_{10}, t_3^{(2)} \in \delta_{15}.$$

Это означает, что соответствующим событиям в ИМ будут присвоены следующие моменты наступления:

$$A_1^{(2)} \sim 3\Delta t, A_1^{(1)}, A_2^{(2)} \sim 6\Delta t, A_2^{(1)} \sim 10\Delta t, A_3^{(2)} \sim 15\Delta t.$$

При этом фазовая траектория системы S с вектором состояний $x(t) \in X$ будет иметь вид

$$\begin{aligned} x(0), x(\Delta t) = x(2\Delta t) = x(0), x(3\Delta t) = x(t_1^{(2)}), x(4\Delta t) = x(5\Delta t) = x(3\Delta t), \\ x(6\Delta t) = x(t_2^{(2)}), x(7\Delta t) = x(8\Delta t) = x(9\Delta t) = (6\Delta t), x(10\Delta t) = x(t_2^{(1)}), \\ x(11\Delta t) = \dots = x(14\Delta t) = x(10\Delta t), x(15\Delta t) = x(t_3^{(2)}). \end{aligned}$$

На основании полученной фазовой траектории можно сделать следующие выводы относительно выбора Δt :

1) если Δt мало, то выполняется много лишних вычислений состояний системы в моменты, когда вектор $x(t)$ не изменяется (за счет этого возрастает время выполнения ИМ);

2) даже при сравнительно малом значении Δt моменты наступления событий в системе (а следовательно, и моменты изменения состояния системы) не совпадают с моментами наступления событий в ИМ, поэтому фазовая траектория, построенная с помощью ИМ, на множестве $T \subset [0, T_M]$ не совпадает с фазовой траекторией системы S .

Принцип Δx .

В соответствии с принципом Δx изменение модельного времени происходит в моменты наступления событий или, что то же самое, в моменты особых состояний, т. е. для нашего примера

$$T_i = \{0, t_1^{(2)}, t_1^{(1)}, t_2^{(2)}, t_2^{(1)}, t_3^{(2)} \leq T_i \},$$

а фазовая траектория, построенная с помощью ИМ, будет совпадать на множестве $T_M \subset [0, T_M]$ с фазовой траекторией системы S :

$$x(0), x(t_1^{(2)}), x(t_1^{(1)}), x(t_2^{(2)}), x(t_2^{(1)}), x(t_3^{(2)}).$$

Приведем более строгие формулировки правил изменения МВ по принципам Δt и Δx .

Пусть $t^* < T_M$ — некоторый момент особого состояния системы S ;

r_i — число событий, произошедших с элементом $A^{(i)} \in S$ до момента t^* включительно ($i = 1, \dots, N$); $t_{r_i}^{(i)} (t_{r_i}^{(i)} \leq t^*)$ — момент наступления последнего для элемента $A^{(i)}$ события до момента t^* включительно; $t_{r+1_i}^{(i)} > t^*$ — момент наступления ближайшего после r_i будущего события;

$r = \sum_{i=1}^N r_i$ — общее число событий в момент t^* ; t^{**} и τ^{**} — моменты

ближайших будущих событий в ИМ, вычисленные по принципам Δt и Δx соответственно.

Модельное время t в ИМ можно рассматривать как функцию от числа событий, происходящих в ИМ. Очевидно: $t(r) = t^* < T_M, r = 0, 1, 2, \dots,$

$$t(r+1) = t^{**} = \min \{t_{r+1}^{(1)}, \dots, t_{r+1}^{(N)}, T_M\} = t(r) + \min \{t_{r+1}^{(N)}, T_M - t(r)\}; \quad (3.6)$$

$$t(r+1) = \tau^{**} = r\Delta t, \text{ если } t^{**} \in \delta_r, (r = \overline{1, v}, v\Delta t = T_M), \quad (3.7)$$

где $\{t_j^{(i)}\}$, $\{\tau_j^{(i)}\}$ определяются соотношением (3.6). Заметим, что моменты $t^{**} = T_m$ и $\tau^{**} = v\Delta t$ (если $T \in \delta_v$) являются моментами завершения моделирования. Правила (3.6) и (3.7) называются правилами изменения *модельного времени* по принципам Δt и Δx соответственно.

На практике отдается предпочтение принципу Δx . Принцип Δt используется лишь в случаях, когда:

1) события $\{A_j^{(i)}\}$ таковы, что $t_j^{(i)} - t_{j-1}^{(i)} \cong \text{const}$ на всем интервале моделирования T_m , и, следовательно, можно подобрать интервал Δt изменения МВ, обеспечивающий минимальную погрешность аппроксимации (например, для разностных уравнений);

2) событий очень много и они появляются группами. В этом случае за счет групповой обработки событий $\{A_j^{(i)}\}$, попавших внутрь очередного шага изменения Δt , удастся уменьшить затраты машинного времени.

В большинстве практически важных случаев события $\{A_j^{(i)}\}$ наступают через случайные интервалы времени $\{t_j^{(i)}\}$. Поэтому способ задания шага до следующего события экономичнее (в смысле затрат машинного времени) и точнее (в смысле точности аппроксимации) фазовой траектории способа фиксированного изменения МВ. В связи с реализацией такого способа представления МВ при моделировании на GPSS/Н в англоязычной литературе он называется моделированием дискретных событий (Discrete Event Simulation). Такое название никоим образом не противоречит выбранному типу концептуальной НВ-модели, а лишь отражает способ представления модельного времени.

§ 3.3. СПОСОБЫ СОЗДАНИЯ КВАЗИПАРАЛЛЕЛИЗМА ПРИ ИМИТАЦИОННОМ МОДЕЛИРОВАНИИ

По Р. Шеннону, каждая ИМ представляет собой комбинацию компонентов, параметров, переменных, функциональных зависимостей, ограничений, целевых функций.

Под *компонентами* будем понимать составную часть исследуемой системы, величина которой зависит от степени детализации рассмотрения системы (элемент, блок, устройство, подсистема).

Параметры выбираются исследователем, и обычно в течение прогона, а иногда и в процессе всего ИМ они сохраняются неизменными.

Переменные бывают внешние и внутренние.

Функциональные зависимости связывают переменные и параметры внутри компоненты или между компонентами.

Ограничения представляют собой пределы значений переменных или условия их изменений.

Целевая функция представляет отображение целей или задач системы и выражается в терминах, интересных исследователю (мощность, быстродействие, экономические показатели и т. д.).

Для рассмотрения особенностей ИМ воспользуемся примером И. В. Максимея [8], который представляется наиболее наглядным для иллюстрации связи между процессами в реальной системе и в модели (рис. 3.6). Пример подчеркивает то важное обстоятельство МВ, что при моделировании надо осуществлять создание квазипараллелизма одним из способов, рассмотренных ниже.

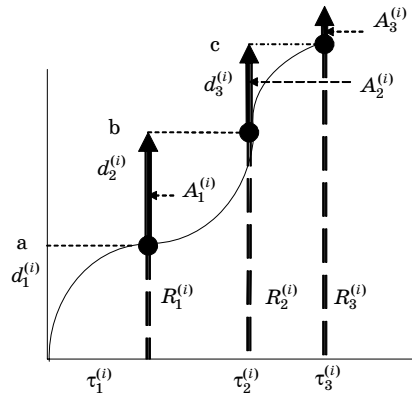


Рис. 3.6. Процессы в модели и системе: $A_j^{(i)}$ — события в системе и модели; $R_j^{(i)}$ — реальные действия в системе; $d_j^{(i)}$ — имитирующие действия в модели; $\tau_j^{(i)}$ — отрезки времени между событиями

Пример. Объектом имитации является движение космической ракеты. В режиме запуска можно выделить последовательность реальных действий от $R_1^{(i)}$ на стартовой площадке до $R_3^{(i)}$ — сброса второй ступени. Движение множества ракет представляет собой систему, а каждая i -я ракета является компонентой. В результате выполнения действий возникают события $A_j^{(i)}$.

Отличие $R_j^{(i)}$ от $d_j^{(i)}$ определяет уровень детализации модели и порождает ошибки имитации реальной системы. Очевидно, что каждое действие $d_j^{(i)}$ описывается алгоритмом $AE_j^{(i)}$. Тогда при переходе от события к событию реализуется действие при неизменном значении времени, а потом изменяется время на $\tau_j^{(i)}$. В принципе, возможно и обратное поведение, т. е. вначале изменяется время, а затем реализуется действие посредством алгоритма. Для реальной системы фазовая траектория запишется в виде $0 - A_1^{(i)} - A_2^{(i)} - A_3^{(i)}$; в имитационной модели — в виде $0 - a - A_1^{(i)} - b - A_2^{(i)} - c - A_3^{(i)}$ или $\tau_1^{(i)} - A_1^{(i)} - \tau_2^{(i)} - A_2^{(i)} - \tau_3^{(i)} - A_3^{(i)}$.

В любом случае мы имеем активность $AK_j^{(i)}(d_j^{(i)}, \tau_j^{(i)})$ как некую молекулу, содержащую описание алгоритма, приводящего к действию, и оператор изменения временной координаты $I_{t_j}^{(i)}$ (временной модификатор).

На рис. 3.7 дается представление о различиях между исследуемой системой и ее моделью. Каждому компоненту системы соответствует аналогичный компонент модели, причем каждый компонент может

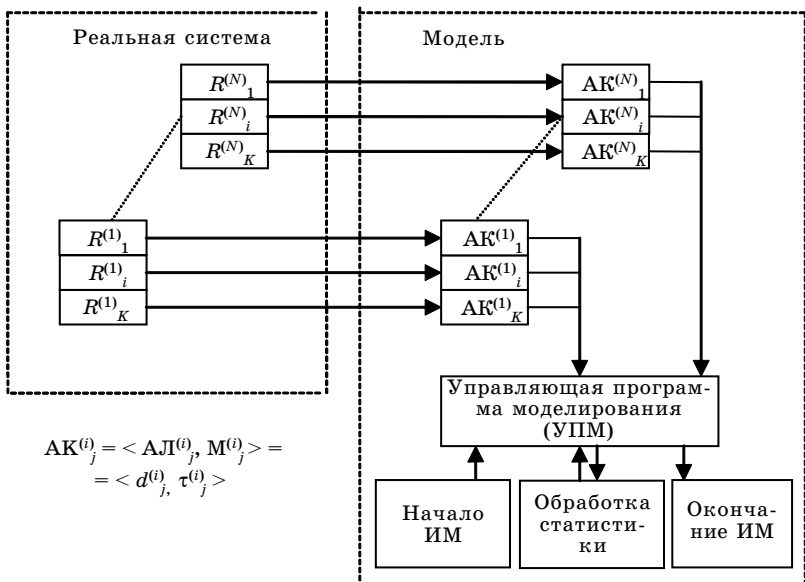


Рис. 3.7. Отличия между системой и ее моделью

включать ряд элементов со своими действиями, которым в модели соответствующий алгоритм и модификатор времени.

Общее управление ведется с помощью УПМ, связывающей запуск алгоритмов, временных модификаторов, проверку условий имитации, обработку статистики, окончание испытаний, выдачу результатов. Таким образом, описание ИМ разрастается в объеме по сравнению с описанием системы. ИМ состоит из двух частей: первая часть является переменной, объектно-ориентированной и создается исследователем с помощью определенных процедур; вторая часть практически инвариантна к виду исследуемой системы и представляет собой реализацию процедур синхронизации модели, запуска, сбора статистики и завершения имитации.

В зависимости от состава алгоритмов, связей между компонентами, целей и задач моделирования можно выбрать различные способы представления квазипараллелизма, а соответственно, способы имитации (приставка «квази» отражает последовательный характер обслуживания событий в ИМ, одновременно возникающих в разных компонентах реальной системы).

В табл. 3.3 приведены типы описания ИМ и способы организации квазипараллелизма в ИМ.

Одну и ту же систему можно представить одним из способов табл. 3.3, но ИМ на их основе отличаются размерами и количеством

Таблица 3.3. Способы организации квазипараллелизма

Тип описания ИМ	Способ организации квазипараллелизма
Активностями	Просмотр активностей в УПМ
Событиями	Составление расписания событий
Транзактами	Управление обслуживанием транзактов
Агрегатами	Управление агрегатами
Процессами	Синхронизация процессов

ресурсов, затрачиваемых на их создание, испытания и использование. Рассмотрим особенности и принципы организации квазипараллелизма в ИМ каждым из указанных способов.

Просмотр активностей

Система, описываемая этим способом, характеризуется следующим:

- все действия для элемента $A_j^{(i)}$ системы S различны и приводят к наступлению разных событий;
- каждое действие $d_j^{(i)}$ характеризуется набором условий его выполнения, представляемых алгоритмически;
- времена выполнения действий являются случайными величинами с известными законами распределения.

Имитационная модель описывается в виде двух частей: множества активностей $\{AE_j^{(i)}\}$ и набора процедур выполнимости условий инициализации активностей. (Инициализация — передача управления от УПМ на выполнение алгоритма данной активности.) Затем происходит модификация временной координаты $M_{jt}^{(i)}$. Таким образом, ИМ представляет собой чередование выполнения алгоритмов активностей, операторов модификации временной координаты t_i и алгоритма УПМ. При этом способе приходится проверять много условий попадания активностей в список инициализированных, поэтому затраты машинного времени весьма велики. Этот способ применяется, когда важно оценить влияние действия на поведение системы.

Составление расписания событий

Используется для систем, характеризующихся следующим:

- множество событий разбивается на небольшое число типов;
- определяют условия перехода от одного события к другому для всех типов событий;
- для каждого типа событий определена последовательность действий, приводящая к изменению состояния системы;
- интервалы времени между последовательными наступлениями событий — случайные величины с известными законами распределения.

Имитационная модель также состоит из двух частей: множества активностей и набора процедур проверки появления событий и инициализации соответствующих активностей. Объединение нескольких активностей в группу существенно сокращает размеры начальных циклов и уменьшает расходы на организацию ИМ. Большим недостатком этого способа является то, что из-за объединения активностей различных подсистем в процедуру событий описание ИМ может потерять сходство с реальной системой. Так, в одной процедуре могут обслуживаться активности, не связанные друг с другом, но приводящие к одним событиям, что затрудняет анализ результатов ИМ.

Транзактный способ

При этом способе действия подсистем одинаковы, активности лишь корректируют значения временных координат. Кроме того, существует зависимость действий друг от друга, которую можно представить в виде СМО. Инициаторами появления событий являются заявки (транзакты) на обслуживание. В ИМ должна быть схема рождения транзактов, их перемещения, уничтожения обслуженных. Для описания ИМ создается фиксированный набор операторов, в GPSS/H совместное число всех операторов (блоков управления и описания немногим превышает 100) и УПМ сканирует списки транзактов, инициализирует блоки, сдвигает модельное время. Событием в ИМ является момент инициализации транзакта, в результате транзакт выступает в роли активности. (Поскольку далее будет рассматриваться только этот способ создания квазипараллелизма, то подробное описание порядка действия при ИМ приводится в гл. 4, 5).

Агрегатный способ

При описании агрегата (под агрегатом понимается объединение ряда устройств для унификации концептуального описания [4]) применим любой из рассмотренных способов, так что выделение агрегатного способа достаточно условно. Функцией УПМ является проверка условий перехода агрегата в одно из особых состояний и моделирование выходных сигналов агрегата. Агрегатный способ удобен при интерпретации системы, но требует больших затрат машинного времени.

Процессный способ

Используется при моделировании систем с различными компонентами, события в которых возникают в различное время, и у каждого компонента своя последовательность действий. При большой степени детализации описания структуры реальной системы и ее модели хорошо совпадают. При этом существует связь не только между компонентами, но и между алгоритмами. Этот метод обычно используют при проектировании новых систем большой размерности, он сочета-

ет в себе черты событийного способа и просмотра активностей. Однако метод требует создания специализированных языков.

В пособии не проводится сравнительный анализ методов создания квазипараллелизма, так как при дальнейшем изложении рассматривается только транзактный метод, для более развернутого изучения этих методов необходимо обратиться к специальной литературе, например [4].

§ 3.4. МЕТОДЫ ИМИТАЦИИ СЛУЧАЙНЫХ ЧИСЕЛ

3.4.1. Исторический экскурс

Любой процесс ИМ прежде всего зависит от качества случайных чисел, векторов или функций, вводимых в модель системы. В нашем случае они должны корректно представить входной поток заявок и поток их обслуживания (см. § 2.3). Любое случайное число или вектор, попадающий на вход модели системы, будем называть случайным элементом Θ^* (СЭ). Во всех случаях СЭ Θ^* должен удовлетворять двум основным принципам:

1) сходство между оригиналом Θ^* и его моделью Θ состоит в совпадении вероятностных законов распределения или числовых характеристик;

2) всякий СЭ Θ конструируется как некая борелевская функция на основе БСВ, генерируемых тем или иным путем.

Первым способом получения БСВ можно назвать попытку У. Госета (псевдоним Стьюдент) в 1908 г., когда он использовал семизначные телефонные номера, отбрасывая первые три цифры, которые не являлись случайными по определению.

Предположим, получалась комбинация из 16 чисел 2127210128891172, если требовалось создать равномерно распределенную (РР) 8-разрядную БСВ, то эта комбинация давала только две РР БСВ: $\Theta_1 = 0.21272101$ и $\Theta_2 = 0.28891172$.

Первая *таблица случайных чисел* была разработана в 1927 г. Л. Типпетом и содержала 41600 БСВ, достаточных для создания 5200 8-разрядных БСВ. Кульминацией таблиц БСВ явился выпуск в 1955 г. корпорацией RAND одного миллиона БСВ, достаточных для получения 125000 8-разрядных БСВ. Очевидно, что для проведения даже простого имитационного эксперимента такого количества БСВ явно недостаточно. Кроме того, хранение и воспроизведение табличных БСВ весьма сложно и длительно, поэтому табличные датчики используются только для назначения номеров телефонов, автомашин и практически не используются при ИМ на ЭВМ.

Следующим возможным вариантом датчиков БСВ является *физический датчик* — специальное радиоэлектронное устройство, являющееся приставкой к ЭВМ, выходной сигнал которого имитирует БСВ. Он состоит из источника флуктуационного шума (например, «флуктуационно шумящей» радиолампы), значение которого в произвольный момент времени является случайной величиной $\eta \geq 0$ с плотностью $p_\eta(y)$, и нелинейного преобразователя

$$\alpha = \{\eta\}_\Delta / \Delta, \quad (3.8)$$

где $\{\eta\}_\Delta = \eta - \Delta[\eta/\Delta]$ — дробная часть числа η относительно заданного $\Delta > 0$ ($[\eta]$ — целая часть числа η).

Исследуем вероятностные свойства α . По правилам функционального преобразования случайных величин, из (3.8) следует, что плотность распределения α

$$p_\alpha(x) = \begin{cases} \sum_{j=0}^{\infty} p_\eta((x+j)\Delta)\Delta, & 0 \leq x < 1; \\ 0, & x \notin [0, 1). \end{cases} \quad (3.9)$$

Будем предполагать, что $p_\eta(y)$ непрерывно дифференцируема и

$$p_\eta(0) < \infty, p_\eta(\infty) < \infty. \quad (3.10)$$

Применим к j -му слагаемому из (3.10) в окрестности точки $y = j\Delta$ линейную формулу Тейлора с остаточным членом Лагранжа:

$$p_\alpha(x) = \begin{cases} \sum_{j=0}^{\infty} \Delta p_\eta(j\Delta) + x\Delta \sum_{j=0}^{\infty} \Delta p'_\eta((\theta x + j)\Delta)\Delta, & 0 \leq x < 1; \\ 0, & x \notin [0, 1), \end{cases} \quad (3.11)$$

где $0 < \theta < 1$. При $\Delta \rightarrow 0$ по свойствам плотности распределения

$$\sum_{j=0}^{\infty} \Delta p_\eta(j\Delta) \rightarrow \int_0^{\infty} p_\eta(y) dy,$$

$$\sum_0^{\infty} \Delta p'_\eta((\theta x + j)\Delta) \rightarrow \int_0^{\infty} p'_\eta(y) dy = p_\eta(\infty) - p_\eta(0),$$

поэтому из (3.9), (3.10) следует, что

$$p_\alpha(x) \rightarrow \begin{cases} 1, & x \in [0, 1); \\ 0, & x \notin [0, 1). \end{cases} \quad (3.12)$$

Таким образом, выбирая Δ достаточно малой величиной, удается получить БСВ α .

Недостатки физического датчика БСВ:

1) невозможность повторения некоторой ранее полученной реализации a (поскольку $P\{\alpha = a\} = 0$);

2) схемная нестабильность, приводящая к необходимости контролировать работу датчика при очередном его использовании.

По этим причинам на современных компьютерах физические датчики БСВ используются весьма редко.

Указанными недостатками не обладает *программный датчик БСВ* — это программа, служащая для имитации на ЭВМ реализации a_1, a_2, \dots БСВ. Он может быть получен из физического датчика БСВ введением обратной связи. Будем рассматривать функционирование датчика во времени и обозначим: η_t — случайную величину, подвергаемую преобразованию (3.8) в момент времени t ; α_t — выходную величину датчика в момент t (случайное число). Источник флуктуационного шума в физическом датчике заменяется обратной связью

$$\eta_t = \psi(\alpha_{t-1}, \alpha_{t-2}, \dots, \alpha_{t-p}), \quad (3.13)$$

использующей p ранее полученных выходных значений датчика. В (3.13) $t = 1, 2, \dots$, а $\alpha_0, \alpha_{-1}, \dots, \alpha_{1-p}$ фиксируются заранее: $\alpha_i = a_i (i = 1 - p, 0)$ и называются исходными (стартовыми) случайными числами.

Согласно (3.8), (3.13):

$$\alpha_t = \frac{\{\eta_t\}\Delta}{\Delta} = \frac{\{\psi(\alpha_{t-1}, \alpha_{t-2}, \dots, \alpha_{t-p})\}\Delta}{\Delta} = \Phi(\alpha_{t-1}, \alpha_{t-2}, \dots, \alpha_{t-p}). \quad (3.14)$$

Рекуррентная формула (3.14) определяет последовательность *псевдослучайных чисел* $a_{1-p}, a_{2-p}, \dots, a_0, a_1, \dots, a_t, \dots$. Термин «псевдослучайные» используется по следующим причинам:

1) по происхождению эти числа не случайные; они получаются по известному детерминированному закону (3.14);

2) при специальном выборе функции $\Phi(\cdot)$ по вероятностным характеристикам эти числа похожи на реализации независимых БСВ.

Отметим, что понятие случайности последовательности можно связать со сложностью моделирующего алгоритма и, в частности, со сложностью функции $\Phi(\cdot)$ в (3.14).

Программные датчики, или, как их чаще принято называть, генераторы случайных чисел, встроены непосредственно в ЯИМ и должны отвечать следующим основным условиям:

— скорость генерации БСВ;

- требования к памяти ЭВМ;
- количество и качество генерируемых независимых РР чисел.

Прогресс генераторов БСВ весьма стремителен, и в настоящее время существует много конкурирующих методов создания генераторов, равно как и самих типов программных генераторов. Не менее стремителен и прогресс ЭВМ, поэтому первые два условия давно перестали служить ограничениями, и на первый план выдвинулось требование количества и качества БСВ. Простые расчеты показывают, что для моделирования даже несложной системы может потребоваться несколько миллионов БСВ.

3.4.2. Принципы моделирования БСВ

Простейшим для моделирования на ЭВМ случайным экспериментом является эксперимент, заключающийся в бросании точки наудачу в промежуток $[0, 1)$. Результатом этого эксперимента является координата точки. Математической моделью такого эксперимента является вероятностное пространство (Ω, F, P) , где $\Omega = [0, 1)$ — пространство элементарных событий (элементарное событие $\omega \in \Omega$ заключается в том, что координата брошенной точки равна ω); $F = \delta$ — алгебра, порожденная интервалами из δ ; P — вероятностная мера, определенная для событий (подмножеств) $A \in F$ и совпадающая с мерой Лебега, так что для события $A = \{\omega : \omega \in [0, x)\}$

$$P(A) = P\{\omega \in [0, x)\} = x, \quad x \in [0, 1). \quad (3.15)$$

Базовой случайной величиной на (Ω, F, P) будем называть непрерывную случайную величину

$$\alpha = \alpha(\omega) = \omega, \quad (3.16)$$

равномерно распределенную на полуинтервале $[0, 1)$.

Функция, распределенная БСВ, имеет вид

$$F_\alpha(x) = P\{\alpha < x\} = \begin{cases} 0, & x \leq 0; \\ x, & 0 < x < 1; \\ 1, & x \geq 1, \end{cases} \quad (3.17)$$

а плотность распределения определяется формулой

$$P_\alpha(x) = \begin{cases} 1, & 0 \leq x < 1; \\ 0, & x \notin [0, 1). \end{cases}$$

Будем обозначать закон распределения $\alpha : R [0, 1]$.

Математическое ожидание БСВ (первый начальный момент)

$$\mu = E\{\alpha\} = 1/2;$$

дисперсия (второй центральный момент)

$$\sigma^2 = D\{\alpha\} = E\{(\alpha - \mu)^2\} = 1/12.$$

Наряду с простейшим экспериментом будем рассматривать составной случайный эксперимент, получающийся в результате r -кратного ($r \geq 1$) повторения независимо друг от друга простейших экспериментов. Результатом составного случайного эксперимента является последовательность из r независимых БСВ $\alpha_1, \dots, \alpha_r$ таких, что

$$\alpha_i = \alpha_i(\omega) = \omega_i, \quad i = \overline{1, r}; \quad \omega = (\omega_1, \dots, \omega_r) \in \Omega^r,$$

где ω_i — координата точки, брошенной наудачу в $[0, 1]$ в i -м простейшем эксперименте.

Совместная плотность распределения вероятностей $\alpha_1, \dots, \alpha_r$

$$p_{\alpha_1, \dots, \alpha_r}(x_1, \dots, x_r) = \begin{cases} 1, & x_i \in [0, 1), \quad i = \overline{1, r}; \\ 0 & \text{в противном случае.} \end{cases}$$

Согласно второму принципу моделирования случайных элементов, любой СЭ Θ представляется для некоторого натурального r в виде функции $f(\cdot)$ от r независимых БСВ:

$$\Theta = f(\alpha_1, \dots, \alpha_r).$$

Таким образом, задача моделирования произвольного СЭ Θ^* разбивается на две подзадачи:

- 1) моделирование на ЭВМ независимых БСВ $\alpha_1, \dots, \alpha_r$;
- 2) нахождение функции $f(\cdot)$ такой, чтобы СЭ Θ обладал требуемыми вероятностным законом распределения и числовыми характеристиками.

Поэтому моделирующий алгоритм состоит из двух блоков (рис. 3.8):

Для имитации одного и того же СЭ Θ^* может быть предложено несколько вариантов функциональных преобразований. Обычно пред-

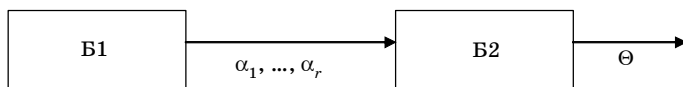


Рис. 3.8. Моделирующий алгоритм БСВ: B1 — блок моделирования БСВ (общий для всех Θ); B2 — блок функционального преобразования $f(\cdot)$ БСВ (различный для различных законов распределения вероятностей)

почтение отдается варианту $f(\cdot)$, требующему меньших вычислительных затрат; для этого применяется понятие коэффициента использования БСВ.

Коэффициентом использования БСВ K назовем величину, обратную числу r базовых случайных величин, используемых для моделирования одной реализации случайного элемента Θ^* :

$$K = 1/r, \quad 0 < K \leq 1.$$

Величина K является мерой вычислительных затрат на моделирование Θ^* . Чем меньше K , тем больше затраты. Целесообразно выбирать такую функцию $f(\cdot)$, для которой K принимает наибольшее значение.

Очевидно, чтобы моделировать на ЭВМ случайные элементы с заданным вероятностным законом распределения, необходимо уметь моделировать БСВ. БСВ α является абсолютно непрерывной случайной величиной. Однако на ЭВМ приходится иметь дело с *дискретными случайными величинами* (ДСВ). Поэтому моделирование БСВ основано на аппроксимации непрерывной случайной величины α ДСВ α' . Опишем способ построения ДСВ α' .

Рассмотрим случай, когда представление целых неотрицательных чисел на ЭВМ осуществляется с помощью k двоичных разрядов (битов). Тогда $C = \{0, 1, 2^k - 1\}$ — множество 2^k неотрицательных целых чисел, представимых в ЭВМ. Определим на (Ω, Φ, P) ДСВ $\beta = \beta(\omega)$ следующим образом:

$$\beta = \beta(\omega) = \begin{cases} 0, & 0 \leq \omega \leq 2^{-k}; \\ \vdots \\ i = i2^{-k} \leq \omega < (i+1)2^{-k}, & i \in C; \\ \vdots \\ 2^k - 1, & 1 - 2^{-k} \leq \omega < 2^k 2^{-k} = 1. \end{cases} \quad (3.18)$$

Построение β проиллюстрируем с помощью рис. 3.9. Разобьем промежуток $[0, 1)$ на 2^k отрезков одинаковой длины 2^{-k} ; для ω , попадающих в промежуток $[i2^{-k}, (i+1)2^{-k})$, полагаем $\beta = \beta(\omega) = i, i \in C$.

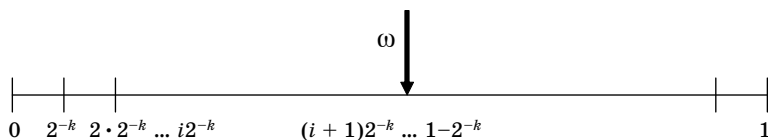


Рис. 3.9. Построение β

По построению распределение ДСВ β является равномерным на множестве C , т. е. все значения β равновероятны, действительно:

$$P\{\beta(\omega) = i\} = P\left\{\omega: \omega \in \left[i2^{-k}, (i+1)2^{-k} \right)\right\} = 2^{-k}. \quad (3.19)$$

Теперь перейдем от СВ β к искомой ДСВ α' :

$$\alpha' = \alpha'(\omega) = \beta(\omega) / 2^k = \beta(\omega) 2^{-k}. \quad (3.20)$$

Согласно (3.19):

$$\alpha' = \begin{cases} 0, & 0 \leq \omega \leq 2^{-k}, (\beta = 0); \\ \vdots \\ i2^{-k}, & i2^{-k} \leq \omega < (i+1)2^{-k}, (\beta = i), i \in C; \\ \vdots \\ 1 - 2^{-k}, & 1 - 2^{-k} \leq \omega < 1, (\beta = 2^k - 1), \end{cases}$$

т. е. от целочисленной ДСВ β мы перешли к ДСВ α' со значениями в $[0, 1)$.

Очевидно, все возможные значения α' определяются множеством $C' = \{0, 2^{-k}, \dots, 1 - 2^{-k}\}$ и являются равновероятными:

$$P\{\alpha'(\omega) = i2^{-k}\} = P\{\beta(\omega) = i\} = 2^{-k}, i \in C,$$

т. е. закон распределения α' является равномерным на C' .

Точность аппроксимации α с помощью α' устанавливается с помощью леммы.

Лемма 3.1. Для СВ $\alpha = \alpha(\omega)$ и $\alpha' = \alpha'(\omega)$, определенных на (Ω, Φ, P) и имеющих вид (3.16), (3.17) соответственно, равномерное уклонение удовлетворяет выражению

$$\sup_{\omega \in \Omega} |\alpha'(\omega) - \alpha(\omega)| < 2^{-k}. \quad (3.21)$$

Доказательство. Разобьем $\Omega = [0, 1]$ на 2^k промежутков согласно рис. 3.9. Пусть $\omega \in [i2^{-k}, (i+1)2^{-k})$, $i \in C$. Тогда, согласно (3.16), справедливо представление

$$\alpha = \alpha(\omega) = \omega = i2^{-k} + \delta, 0 \leq \delta < 2^{-k}, i \in C,$$

а согласно (3.20):

$$\alpha' = \alpha'(\omega) = i2^{-k}.$$

Поэтому $|\alpha - \alpha'| = \delta < 2^{-k}$, $\omega \in \Omega$. Отсюда заключаем справедливость (3.21).

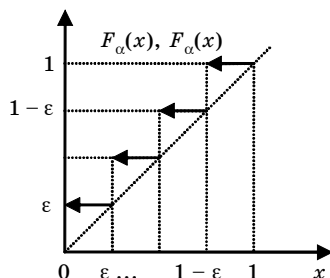


Рис. 3.10. Функция распределения $F_{\alpha}(x)$

Таблица 3.4. Соотношения между дисперсиями величин α и α'

k	2	3	5	10	15
$\left(\frac{D\{\alpha'\}}{D\{\alpha\}}\right)^{1/2}$	1.290	1.140	1.030	1.001	1.00

Из (3.21) следует, что если $k \rightarrow \infty$, то последовательность $\alpha'_k \rightarrow \alpha$ равномерна по $\omega \in \Omega$. Таким образом, случайная величина $\alpha' = \alpha'(\omega)$ является аппроксимацией для БСВ¹ $\alpha = \alpha(\omega)$; α' называется в связи с этим *квазиравномерной случайной величиной*. Ее функция распределения $F_{\alpha'}(x)$ показана на рис. 3.10 и аппроксимирует $F_{\alpha}(x)$ с точностью $\varepsilon = 2^{-k}$.

Между математическими ожиданиями величин α , α' справедливо соотношение

$$E\{\alpha'\} = 0,5(1 - 2^{-k}) \rightarrow E\{\alpha\} = 0,5,$$

представленное в табл. 3.4.

При достаточно больших значениях k (например, для ПЭВМ IBM PC AT 486, Pentium $k = 31$, $\varepsilon = 0.5 \cdot 10^{-9}$) величины α и α' отождествляют.

3.4.3. Методы построения программных генераторов

Существует ряд методов и алгоритмов, используемых при построении генераторов БСВ [4, 11]. Наиболее широкое распространение получили генераторы, построенные на использовании различных

¹ Напомним, что аппроксимация непрерывных случайных величин дискретными широко используется в теории вероятностей при построении интеграла Лебега.

модификаций метода вычетов (мультипликативный конгруэнтный (совпадающий) метод), который в разных модификациях используется в разных редакциях GPSS/Н. В конце параграфа будут приведены параметры генераторов GPSS/Н. Рассмотрим этот метод подробнее.

Любое следующее число образуется на основании выражения

$$x_{i+1} = ax_i + c \pmod{m} \quad i = 1, 2, \dots, \quad (3.22)$$

где a, m — положительные целые числа; c — целое число.

Исторически этот генератор строился из принципа

$$x_n = x^n \pmod{m}. \quad (3.23)$$

Чтобы не возникало вычислительных проблем, (3.23) можно заметить

$$x_n = xx_{n-1} \pmod{m}. \quad (3.24)$$

При $c = 0$ как раз и получается мультипликативный конгруэнтный генератор и генерируемые числа $U_i = x_i / m$. Методы проверки случайности генерируемых чисел будут рассмотрены в п. 3.5.3. Выражение (3.24) основывается на значении $m = 2^k$, где k — целое число, обычно равное длине машинного слова. Любые генерируемые БСВ характеризуются *периодом повторения* или *циклом генератора* $d = 2^{k-2}$ при нечетном x_0 (примем это без доказательства).

В качестве примера рассмотрим генератор $x_n = 3x_{n-1} \pmod{32}$ (2^5).

При $c = 0$ [см. (3.22)] и $a = \pm 3 \pmod{8}$ период этого генератора $d = 8^{5-2} = 8$. Приняв $x_0 = 1$, найдем числовую последовательность, создаваемую генератором:

$x_0 = 1$	00001
$x_1 = 3$	00011
$x_2 = 9$	01001
$x_3 = 27$	11011
$x_4 = 81 \pmod{32} (81 - 2 \cdot 32) = 17$	10001
$x_5 = 51 \pmod{32} = 19$	10011
$x_6 = 57 \pmod{32} = 25$	11001
$x_7 = 75 \pmod{32} = 11$	11001
$x_8 = 33 \pmod{32} = 1$	00001

Последовательность повторилась, т. е. период равен 8, как и было подсчитано, в случае четного x_0 *период сокращается вдвое!*

Современные ЭВМ (Pentium 1 и выше) сняли все ограничения по быстродействию и объемам памяти, необходимым для генерации качественных БСВ. Потому при выборе генератора БСВ надо руководствоваться следующими правилами:

- 1) период повторения должен быть более миллиона;

2) БСВ, производимые генератором, должны отвечать всем тестам на случайность;

3) стараться пользоваться стандартными генераторами типа RANDU (URN01 — URN42), встроенными непосредственно в ЯИМ.

Рассмотрим более подробно схему повторения БСВ. На интервале $(0, m)$ должны быть такие числа r и s ($0 \leq r < s < m$), для которых $x_r = x_s$, а следовательно, и $x_{r+l} = x_{s+l}$ для всех положительных l . Если s наименьшее положительное число, для которого $x_r = x_s$, тогда выражение (3.22) состоит из начального сегмента x_0, x_1, \dots, x_{r-1} , предшествующего второму сегменту $x_r, x_{r+1}, \dots, x_r = x_s$, который затем постоянно повторяется. Тогда целое число $d = s - r$ называется *периодом повторения, последовательностью* или *циклом генератора*. Последовательность имеет максимальный период, если $d = m$. Проиллюстрируем зависимость периода d от a, c, m, x_0 на двух простых примерах.

Пример 1. Пусть $x_0 = 1, a = 5, c = 0, m = 10$, тогда определяем последовательность $x_1 = 9 \bmod 10 = 9, x_2 = 81 \bmod 10 = 1, x_3 = 9 \bmod 10 = 9, \dots$. Последовательность имеет вид $1, 9, 1, 9, \dots$. В данном случае $r = 0, s = 2$ и период $d = s - r = 2$.

Пример 2. Пусть $x_0 = 1, a = 5, c = 0, m = 10$, тогда определяем последовательность $x_1 = 5 \bmod 10 = 5, x_2 = 25 \bmod 10 = 5, \dots$. Последовательность имеет вид $1, 5, 5, \dots$. В данном случае $r = 1, s = 2, d = 1$. У данного генератора начальный сегмент 1 предваряет последовательное число 5.

В табл. 3.5 приведены данные генератора для различных значений x_0, a для уравнения (3.23) — при $c = 0$ и $m = 10$.

Таблица 3.5. Данные генератора

x_0	a								
	1	2	3	4	5	6	7	8	9
1	1	1,2,4,8,6,	1,3,9,7,	1,4,6,	1,5,	1,6,	1,7,9,3,	1,8,4,2,6,	1,9,
2	2	2,4,8,6,	2,6,8,4,	2,8,	2,0,	2,	2,4,8,6,	2,6,8,4,	2,8,
3	3	3,6,2,4,8,	3,9,7,1,	3,2,8,	3,5,	3,8,	3,1,7,9,	3,4,2,6,8,	3,7,
4	4	4,8,6,2,	4,2,6,8,	4,6,	4,0,	1,4,	4,8,6,2,	4,2,6,8,	4,6,
5	5	5,0,	5,	5,0,	5,	5,0,	5,	5,0,	5,
6	6	6,2,4,8,	6,8,4,2,	6,4,	6,0,	6,	6,2,4,8,	6,8,4,2,	6,4,
7	7	7,4,8,6,2,	7,1,3,9,	7,8,2,	7,5,	7,2,	7,9,3,1,	7,6,8,4,2,	7,3,
8	8	8,6,2,4,	8,4,2,6,	8,2,	8,0,	1,8,	8,6,2,4,	8,4,2,6,	8,2,
9	9	9,8,6,2,4,	9,7,1,3,	9,6,4,	9,5,	9,4,	9,3,1,7,	9,2,6,8,4,	9,1,

По данным табл. 3.5, наибольшее значение периода d равно 4, а плохой выбор a и x_0 значительно уменьшает этот период, в таблице подчеркнуты повторяющиеся сегменты.

Для выбора генератора, обладающего наибольшим периодом, рассмотрим несколько лемм.

Лемма 3.2. Если a и m — простые числа, всегда есть положительное целое d , для которого $x_0 = x_d$. Для простых чисел наибольшим общим делителем (НОД) является единица.

Лемма 3.3. Если $\text{НОД}(a, m) = 1$, то тогда период последовательности (3.22) является наименьшим целым положительным числом d , для которого

$$\sigma_d((a-1)x_0 + c) = 0 \pmod{m}, \quad (3.25)$$

где $\sigma_d = 1 + a + a^2 + \dots + a^{d-1}$.

Необходимо отметить, что эти леммы действительны только для случая, когда НОД равен 1, поэтому для гарантии надо выбирать только такой генератор, у которого НОД равен 1.

Стремление увеличивать период является необходимым, но не достаточным условием. При увеличении периода может произойти ухудшение качества БСВ, поэтому их необходимо проверять на случайность, что и рассмотрено в § 3.5.

В заключение параграфа рассмотрим тип генератора БСВ, применяемый в GPSS/H. В первой версии языка использован генератор URN 27, работающий на сдвиговом принципе с обратной связью и использующий выражение $x^{31} + x^3 + 1$. Это выражение комбинируется с последовательностью $x_i = 69069 x_{i-1} \pmod{2^{32}}$. Сдвиг вправо на 15 разрядов и влево на 17 обеспечивает получение 32-разрядного числа y_i . Эти x_i и y_i складываются по правилам алгебры логики и создают последовательность z_i , так, например:

$$\begin{aligned} x_i &= 01001\dots01010 \\ y_i &= 00101\dots10110 \\ z_i &= 01100\dots11100 \end{aligned}$$

Этот генератор не проходил по некоторым тестам случайности, поэтому в более поздних версиях использован усовершенствованный генератор, имеющий в качестве минимального 32-разрядного числа (положительный ноль и 31 значащий разряд) минимальное значение 1 и максимальное 2 млрд 147 млн 483 тыс. 646 ($2^{31} - 2$), а выражение принимает вид $x_{i+1} = 742938285 x_i \pmod{2147483647}$, при этом $x_0 = 266301881$. Таким образом, период повторения этого генератора весьма велик (более 2 млрд), что вполне достаточно для моделирования сложных систем. Независимость потоков БСВ достигается тем,

что потоки заявок и обслуживания берутся с разных генераторов. В ранних версиях использовались 8 встроенных генераторов, стартующих с одинаковых начальных позиций, что делало потоки идентичными, а задачу сбора статистики достаточно сложной. В последних версиях статистическая независимость достигается практически неограниченным числом потоков и отличием каждой последующей выборки на 100 000. Такое отличие потоков практически исключает возможность их корреляции. Более тонкие моменты исследования регулярности этих процессов в пособии не рассматриваются. Кроме того, регулировка сдвига начальных значений легко осуществляется с помощью оператора управления (ОУ) RMULT и ОБ BRMULT, которые позволяют сдвигать начальные значения на один миллион и более! Следует подчеркнуть некоторые методические особенности использования генераторов БСВ. Разброс времени $A \pm B$ в ОБ GENERATE и ADVANCE всегда берется с генератора RN1, с этого же генератора берется случайное значение ОБ TRANSFER в статистическом виде.

§ 3.5. ОЦЕНКА КАЧЕСТВА БАЗОВЫХ СЛУЧАЙНЫХ ВЕЛИЧИН, ПОЛУЧАЕМЫХ ОТ ПРОГРАММНЫХ ГЕНЕРАТОРОВ

3.5.1. Общие представления

Рассмотрим кратко вопросы оценки качества БСВ, т. е. соответствие последовательности БСВ равномерности распределения в интервале (0,1). Такую оценку позволяют осуществлять статистические тесты. Поскольку причин отклонения БСВ от случайности много, то и тестов проверки также много [15]. Идеальным генератором считается тот, который проходит все статистические тесты. Каждый из таких тестов ориентирован на определение частной причины отклонения последовательности от независимости и случайности. Будем считать, что существует k возможных причин отклонений, и при выполнении гипотезы случайности вероятность появления i -й причины равняется π_i , $i = 1, 2, \dots, k$. Предположим, что рассматривается достаточно большая последовательность, в которой по разным причинам возникло n несоответствий по всем k причинам. Количество несоответствий по каждой группе обозначим как f_i , тогда $f_1 + \dots + f_k = n$; так как f_i имеет вероятность π_i , то число несоответствий равно $n\pi_1 = e_1$. Измерение различия между f_i и e_i определяется из выражения

$$D = (f_1 - e_1)^2 / e_1 + \dots + (f_k - e_k)^2 / e_k = \sum_{i=1}^k (f_i - e_i)^2 / e_i. \quad (3.26)$$

Таблица 3.6. Значения c для различного числа степеней свободы γ

γ	1	2	3	4	5	10	20	50	100
c	3.841	5.992	7.814	9.487	11.071	18.307	31.411	67.504	124.342

При истинной случайности разность между f_i и e_i должна стремиться к нулю, а D должно быть достаточно малым, следовательно, гипотеза о случайности отвергается при большом значении D . При уровне значимости 0.05 вероятность отвергнуть истинную гипотезу о случайности равна

$$P_{H_0}(D > c) = 0,05. \quad (3.27)$$

Оценить точное распределение D достаточно сложно, однако при $n > 5$ D приблизительно распределено по хи-квадрат-распределению с числом степеней свободы $\gamma = k - 1$. Для большей уверенности общее число несоответствий обычно принимают равным 10. В табл. 3.6 приведены значения c для различных значений числа степеней свободы с тремя значащими цифрами после точки, при которых справедливо равенство (3.27).

Тесты позволяют исключить процедуру статистического оценивания, при которой из-за ошибок 1-го и 2-го рода оценка вероятности принятия правильной гипотезы равномерности распределения оказывается достаточно сложной. Ниже кратко рассмотрим два типа тестов: тесты на равномерность БСВ и тесты проверки качества самих генераторов.

3.5.2. Тесты оценки качества БСВ

1. Тест равномерности распределения (хи-квадрат-тест).

Второго названия этого теста следует избегать, чтобы не возникло путаницы из-за совпадения названий при рассмотрении распределения D . При этом тесте интервал $(0, 1)$ делится на сто равных и не пересекающихся интервалов длиной 0.01, вероятности появления несоответствий в каждой подвыборке из n равны $e_1 = e_2 = \dots = e_{100} = n/100$.

Затем проверяется равенство вероятностей в каждом интервале.

2. Тест замещения (сбора купонов).

Имеем последовательность случайных чисел U_1, U_2, \dots , произведем замену чисел на целочисленные значения 1, 2, ..., M по правилу:

Заменим U_i на 2, если $1/M \leq U_i < 2/M$

Заменим U_i на 1, если $0.0 \leq U_i < 1/M$

·
·
·

Заменим U_i на M , если $(M - 1)/M \leq U_i \leq 1.0$

Эти новые целые числа от 1 до M должны быть случайными, если U_i действительно случайные числа. Рассмотрим последовательно всю полученную совокупность и обозначим через Q новые числа, полученные после M , например $M + 1, M + 2, \dots$. При соблюдении случайности вероятности $P(Q = M), P(Q = M + 1), P(Q = M + 2), \dots$ известны. Если вычислить такие значения Q повторно, то тогда тест 2 может быть применен для проверки случайности начальных чисел U_i .

3. Тест расхождения.

Положим, что α и β лежат между 0.0 и 1.0 при условии, что $\alpha < \beta$. Имеющуюся последовательность чисел U_1, U_2, \dots , будем рассматривать с позиции их нахождения: либо в интервале $[\alpha, \beta]$ включительно границы, либо вне его. Будем считать, что каждое число, лежащее в интервале, равно 1, а вне его равно 0. Эта операция преобразует последовательность в набор 0 и 1. Положим, что $p = \beta - \alpha$, тогда, если последовательность чисел случайна, то вероятность того, что j -й 0 появится после 1, прежде чем появится следующая 1, равняется

$$p_j = p(1 - p)^j, \quad j = 0, 1, 2, \dots \quad (3.28)$$

После определения, сколько раз появится каждая категория, можно применить тест 1, который оценивает случайность в соответствии с (3.28). Этот тест введен Кендаллом и Бабингтон-Смитом и предусматривает следующие правила для выбора α и β :

вариант 1 $\alpha = 0.0, \beta = 0.5$ (ниже среднего);

вариант 2 $\alpha = 0.5, \beta = 1.0$ (выше среднего);

вариант 3 $\alpha = 0.333, \beta = 0.667$ (нахождение в середине).

4. Тест перестановки.

Считается, что полученная последовательность случайных чисел U_1, U_2, \dots , содержит наборы T чисел. Каждый такой набор содержит $T!$ возможных перестановок, которые можно рассматривать как T чисел, расположенных в порядке убывания от наибольшего к наименьшему. При случайности вероятность появления каждого набора $T!$ равна $1/T!$. При $T!$ -категориях можно определить, сколько кортежей образовано из соответствующих наборов T и сколько из них попало в соответствующую $T!$ -категорию. Затем применяется тест 1, для этого теста рекомендованы значения $T = 3, 4, 5$.

5. Покер-тест.

В полученной последовательности случайных чисел U_1, U_2, \dots , произведем преобразование в целые числа 1, 2, ..., 10 по правилу:

Заменим U_i на 1, если $0.0 \leq U_i < 0.1$

Заменим U_i на 2, если $0.1 \leq U_i < 0.2$

·
·
·

Заменим U_i на 10, если $0.9 \leq U_i \leq 1$

Новые числа должны быть случайными, если U_i — случайные числа, в этом случае используется тест 2 с $M = 10$.

Вариант 1. Рассмотрим последовательные наборы из пяти чисел:

— пять любых одинаковых чисел из 1–10 обозначим $AAAAA$;

— одно измененное число типа $AAAAB$;

— сочетание типа $AAABB$;

— сочетание типа $AAABC$;

— сочетание типа $AABBC$;

— сочетание типа $AABCD$;

— сочетание типа $ABCDE$.

Только такие сочетания могут встречаться при проверке чисел. Вероятности появления сочетаний из пяти цифр при истинной случайности:

$$P(AAAAA) = 0.0001$$

$$P(AAAAAB) = 0.0045$$

$$P(AAABB) = 0.0090$$

$$P(AAABC) = 0.0720$$

$$P(AABBC) = 0.1080$$

$$P(AABCD) = 0.5040$$

$$P(ABCDE) = 0.3024$$

Обычно в связи с малостью значений вероятности первое и второе сочетания объединяются вместе, в результате получается шесть сочетаний из пяти цифр (отсюда карточное название теста).

Вариант 2. Для варианта 1 можно подсчитать, сколько различных цифр будет в пяти выбранных. Значения легко подсчитываются по данным варианта 1:

$$P(1) = 0.0001$$

$$P(2) = 0.0135$$

$$P(3) = 0.1800$$

$$P(4) = 0.5040$$

$$P(5) = 0.3024$$

В этом варианте первое и второе значения чаще всего также объединяются.

Вариант 3. Часто ограничиваются выбором не пяти цифр в сочетании, а четырех, в этом случае:

$$P(AAAA) = 0.001$$

$$P(AAAB) = 0.036$$

$$P(AABB) = 0.027$$

$$P(AABC) = 0.432$$

$$P(ABCD) = 0.504$$

И здесь первое и второе значения чаще всего объединяются.

6. Тест последовательных пар.

Имеем последовательность чисел U_1, U_2, \dots . Пусть имеется целое число M ($M \geq 2$). Произведем замену каждого числа U_i на выражение $1 + \text{ЦЧ}(U_i M)$, где $\text{ЦЧ}(\cdot)$ — целая часть произведения случайного числа на M ; в случае равенства произведения выбранному числу M $\text{ЦЧ}(\cdot) = M - 1$.

Новые числа будут случайными целыми от 1 до M , если U_i случайны. Теперь образуем матрицу из пар полученных чисел вида

$$(1, 1) (1, 2) \dots (1, M - 1) (1, M)$$

$$(2, 1) (2, 2) \dots (2, M - 1) (2, M)$$

.

.

.

$$(M, 1) (M, 2) \dots (M, M - 1) (M, M)$$

и определим, в какой из M^2 -категорий лежит каждое число. При истинной случайности каждая пара должна иметь одинаковую вероятность, равную $1/M^2$, и после этого можно использовать тест 1. Чаще всего принимают $M = 3, 10, 20$ для первого, второго и третьего варианта соответственно.

Каждый из этих шести тестов оценивается статистиками, которые при случайности последовательности U_i примерно подчиняются хи-квадрат-распределению с числом степеней свободы, заданным в каждом тесте. Обычно выборка проверяемых чисел ограничивается количеством от 1000 до 10000. Если все тесты приводят к значению $D(3.26) \geq 0,95$, то качество случайных чисел достаточно высокое. Если хотя бы один из тестов не проходит, то такой генератор не должен использоваться. Однако при этом следует помнить, что при увеличении объема выборки все тесты могут дать положительный ответ.

Проверка по критерию хи-квадрат, распределений хи-квадрат, получаемых при использовании каждого теста, носит название теста СНСН (chi-square on chi-square). Его введение позволяет нивелиро-

вать непрохождение на случайность по какому-либо из тестов. В настоящее время существует программа TESTRAND, проверяющая 10 млн случайных чисел по всем названным критериям, однако ее применение прежде всего должно интересовать разработчиков генераторов и в меньшей степени — пользователей. *Вместе с тем при ИМ сложных, ответственных систем бесполезно проводить тщательную проверку генерируемых БСВ!*

3.5.3. Теоретическая оценка качества генераторов

В некоторых случаях до проверки последовательности БСВ на случайность проводится теоретическая проверка конгруэнтных генераторов, реализующих последовательность (3.22).

1. Последовательный корреляционный тест.

При случайности последовательности U_1, U_2, \dots , корреляция строго равна 0, т. е. $R(U_i, U_{i+1}) = 0$.

Подставив $U_i = x_i/m$ в (3.22), можно показать, что

$$R(U_i, U_{i+1}) = 1/a(1 - 6c/m + 6(c/m)^2 + \varepsilon) \quad (3.29)$$

при

$$|\varepsilon| \leq (a + 6)/m. \quad (3.30)$$

Если a и m простые числа, возможно (3.30) заменить на равенство. Дадевичем [14] показано, что не рекомендуется применять генератор, у которого правая часть (3.29) больше 0.01, а правая часть (3.30) меньше 0.005. Например, для генератора URN13 при $c = 0$, $a = 663\,608\,941$ и $m = 2^{32}$, проводя оценку по (3.29), (3.30), получаем $1/a = 0.0000000015$, $(a + 6)/m = 0.1545$, следовательно, генератор отвечает сформулированным правилам и проходит корреляционный тест.

2. Спектральный тест (межплоскостных расстояний).

Этот тест основан на утверждении, что n выборок (U_1, \dots, U_n) , (U_2, \dots, U_{n+1}) , (U_3, \dots, U_{n+2}) , ..., полученных от генератора по формуле (3.22), лежат в конечном и малом числе параллельных, одинаково расположенных гиперплоскостей [15]. То, что они лежат в конечном числе плоскостей, очевидно, а то, что количество плоскостей мало, является предметом обсуждения. Каждая плоскость должна содержать по меньшей мере три точки, тогда наименьшее число плоскостей будет равно $(n!m)^{1/n}$. В табл. 3.7 приведены значения этого произведения для разного числа выборок n при $m = 2^{32}$.

Естественно, что существуют различные наборы плоскостей, содержащие все эти точки. Рассмотрим такой набор, когда расстояние

Таблица 3.7. Значения произведения

n	2	3	4	5	6	7	8	9
$m = 2^{32}$	92681	2953	566	220	120	80	60	48

между плоскостями будет наибольшим, тогда критерием спектрального теста будет являться *расстояние между плоскостями* d_n при n измерениях. Существует алгоритм TSTM4, входящий подпрограммой в TESTRAND, легко подсчитывающий это расстояние при различных значениях коэффициентов (3.22). Очевидно, что при фиксированном n , чем меньше d_n , тем лучше генератор. Критерий, применяемый в литературе, выглядит следующим образом: $d_n \leq 2^{-30/n}$, в табл. 3.8 это значение приводится в зависимости от n .

Например, для генератора RANDU, широко применяемого в России, при данных $c = 0$, $a = 2^{16} + 3$, $m = 2^{32}$ имеем: $d_2 = 0.00002$, $d_3 = 0.092$, $d_4 = d_5 = d_6 = d_7 = d_8 = d_9 = 0.093$, т. е. этот генератор соответствует спектральному тесту только для двумерного случая. Кстати, генератор, используемый в GPSS/H, проходит спектральный тест до $n = 9$.

Название «спектральный тест» пришло из теории распространения волн, где наибольшее значение $v_n = 1/d_n$ служило критерием наилучших условий распространения (а в нашем случае — случайности).

Все векторы чисел (U_i, \dots, U_{i+n-1}) лежат в n -мерном единичном кубе. Наименьшее расстояние между двумя плоскостями этого куба равно расстоянию от начального значения $(0, 0, \dots, 0, 0)$ до $(0, 0, \dots, 0, 1)$ и вычисляется как

$$d_L(n) = ((0 - 0)^2 + (0 - 0)^2 + \dots + (0 - 1)^2)^{0.5} = 1,$$

а наибольшее расстояние до $(1, 1, \dots, 1, 1)$ — из выражения

$$d_U(n) = ((0 - 1)^2 + \dots + (0 - 1)^2)^{0.5} = \sqrt{n}.$$

Поскольку расстояние длиной l может быть разделено r плоскостями на $r - 1$ равных промежутка длиной $l/(r - 1)$, то отсюда следует, что генератор номера плоскости $(n!m)^{1/n}$ будет иметь расстояние между плоскостями, лежащее между

$$1/((n!m)^{1/n} - 1) \text{ и } \sqrt{n}/((n!m)^{1/n} - 1). \quad (3.31)$$

Таблица 3.8. Зависимость d_n от n

n	2	3	4	5	6	7	8	9
d_n	0.00003	0.001	0.005	0.016	0.03	0.05	0.07	0.1

Таблица 3.9. Значения наилучшего расстояния

n	2	3	4	5	6	7	8	9
$m = 2^{16}$	0.00281	0.0141	0.0291	0.044	0.056	0.065	0.071	0.076
$m = 2^{32}$	0.00001	0.0003	0.0018	0.005	0.008	0.013	0.017	0.021

Первое значение является наилучшим возможным расстоянием при n -мерной выборке для генератора модуля m . В табл. 3.9 представлены значения наилучшего расстояния в зависимости от n и m , и можно считать, что они являются стандартом для оценки добротности генератора, выраженной в расстоянии между плоскостями d_2, \dots, d_9 при заданном m .

Если поделить d исследуемого генератора на лучшее расстояние, приведенное в табл. 3.9 для выбранного m , то полученное значение r_i в пределе равно 1 и генератор тем лучше, чем ближе результат деления к 1. Например, проверим на основе этого правила генератор RANDU, пользуясь значениями d_L , полученными выше:

$$r_2 = 0.00002/0.00001 = 2, r_3 = 307, r_4 = 52, r_5 = 19, \\ r_6 = 12, r_7 = 7, r_8 = 5, r_9 = 4.$$

Из расчета видно, что наиболее плохое поведение генератора — при размерностях 3 и 4. Из (3.31) найдем число плоскостей для 3-размерного случая для этого генератора. В этом случае расстояние $d_3 = 0.092$, следовательно, нижний предел равен $1/0.092 + 1 = 11.8$, а верхний — $\sqrt{3}/0.092 + 1 = 19.8$. На самом деле, для этого генератора возможное число плоскостей (см. табл. 3.7) равно 2953, поэтому с определенными оговорками этот генератор можно принять.

Пример. Проиллюстрируем сказанное на примере генератора с коэффициентами

$$x_{i+1} = 7x_i \bmod 11, x_0 = 1.$$

Из (3.29) оценим корреляцию $R(U_{i+1}, U_L) = 1/7 + \epsilon$, а из (3.30) $\epsilon \leq 13/11 = 1.18$

Таким образом, такой генератор может пройти тест благодаря большому значению ошибки. При $n = 2$ получим последовательность 1, 7, 5, 2, 3, 10, 4, 6, 9, 8, 1, которая имеет период $m - 1 = 10$. В результате получим 10 пар случайных чисел (1, 7), (7, 5), (5, 2), (2, 3), (3, 10), (10, 4), (4, 6), (6, 9), (9, 8), (8, 1). Спектральный тест предусматривает получение 10 плоскостей с интервалом $1/11 = 0.909$, каждая из которых содержит одну из пар (рис. 3.11, линии А).

Но для конгруэнтного генератора при данных параметрах плоскостей должно быть не больше $(2! 11)^{1/2} = 4$ (рис. 3.11, линии В) — это, естественно, предельный — худший — случай. Линии В (рис. 3.11) представляют

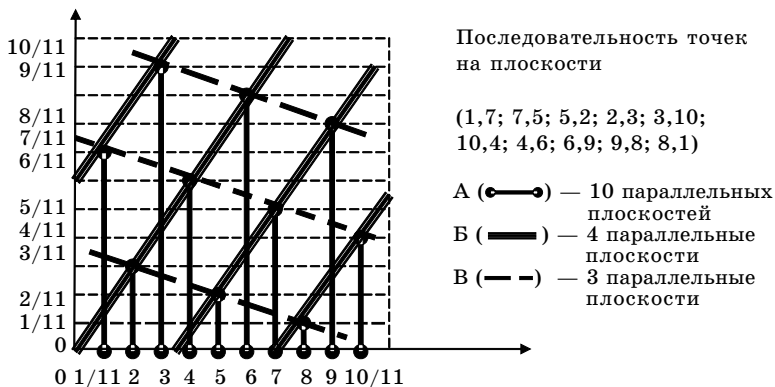


Рис. 3.11. Расположение пар чисел

три плоскости, содержащие все точки. Расстояние между параллельными плоскостями $d_2 = 0.3162$. Лучшее расстояние для этого генератора при $m = 11$, $n = 2$ равно $1 / (4 - 1) = 0.33$; таким образом, $r = 0.362 / 0.333 = 0.95$. Следовательно, рассматриваемый генератор для заданных параметров весьма хорош. Однако, если принять $m = 2^{32}$, что естественно для современных компьютеров, у которых в этом случае лучшее расстояние равно 0.00001 , то значение $r = 0.3162 / 0.00001 = 31620$. Это отношение очень большое, и такой генератор не может использоваться категорически!

Приведенный пример подчеркивает, что рекомендации по выбору генератора должны всегда соблюдаться, среди них главными являются:

- период генератора должен быть не менее 1 млрд;
- генератор должен пройти все проверки.

Использование плохого генератора обесценит все усилия исследователя даже при идеальной модели.