

№5. N4

З. 873 523

Министерство образования Республики Беларусь

МАН

Учреждение образования
«Гомельский государственный университет
имени Франциска Скорины»

ПРАВЕРАНА
2014г.

И. В. МАКСИМЕЙ

**ОСНОВЫ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ**

Тексты лекций
по разделу «Технология имитационного
эксперимента на ЭВМ»
для аспирантов и магистрантов

Б/н

Установка опублікаві
«Гомельскі дзяржаўны ўніверсітэт
імя Францішка Скорины»
БІБЛІЯТЭКА

Гомель 2007

РЕПОЗИТОРИЙ ГРУППЫ

УДК 004.9 + 681.3 (075.8)
ББК 32.973 = 018 я73
М 171

Рецензенты:
кафедра математических проблем управления учреждения образования Гомельский государственный университет имени Франциска Скорины;
О.М. Демиденко, профессор, доктор технических наук.

Рекомендованы к изданию научно-методическим советом учреждения образования «Гомельский государственный университет имени Франциска Скорины»

М 171 Максимей, И.В.
Основы информационных технологий : тексты лекций по разделу «Технология имитационного эксперимента на ЭВМ» для аспирантов и магистрантов / И. В. Максимей ; М-во образ. РБ, Гомельский государственный университет им. Ф. Скорины. – Гомель: ГГУ им. Ф. Скорины, 2007. - 107 с.

Информационные технологии характеризуют уровень развития общества, возможность его интеграции в мировую цивилизацию. Именно этим определяется актуальность и необходимость изучения основ информационных технологий.

Целью текстов лекций по разделу «Технология имитационного эксперимента на ЭВМ» является оказание помощи аспирантам и магистрантам в овладении основами имитационного эксперимента как инструмента для решения научных и практических задач в предметной области и их использования в практической деятельности.

УДК 004.9 + 681.3 (075.8)
ББК 32.973 = 018 я73

© Максимей И.В., 2007
© УО «ГГУ им. Ф. Скорины», 2007

СОДЕРЖАНИЕ

Введение.....	4
Лекция 1 Исследование на вероятностных моделях сложных систем.....	5
Лекция 2 Этапы имитационного моделирования сложных систем.....	29
Лекция 3 Метод статистических испытаний как основа моделирования вероятностных систем.....	44
Лекция 4 Способы организации имитационного моделирования сложных систем.....	66
Литература.....	106

ВВЕДЕНИЕ

Информационные технологии характеризуют уровень развития общества, возможность его интеграции в мировую цивилизацию. Именно этим определяется актуальность и необходимость изучения основ информационных технологий.

Целью курса «Основы информационных технологий» является овладение аспирантами и магистрантами современными информационными технологиями как инструментом для решения на высоком уровне научных и практических задач в своей предметной области.

Задачами курса являются:

- ознакомление с вопросами программного обеспечения современных информационных технологий;
- освоение работы с основными программными продуктами информационных технологий: текстовыми, графическими и табличными процессорами, базами данных, средствами подготовки презентаций, сетевыми клиентскими программами, средствами поддержки математических вычислений;
- овладение методами и средствами решения задач в своей предметной области на базе использования информационных технологий.

Курс «Основы информационных технологий» включает пять разделов:
Раздел 1. Общие вопросы современных информационных технологий.
Раздел 2. Технология имитационного эксперимента на ЭВМ.
Раздел 3. Технология обработки и анализа данных исследований на ПЭВМ.
Раздел 4. Технология организации на ЭВМ баз данных.
Раздел 5. Технология разработки Internet приложений.

В результате усвоения материала аспиранты и магистранты должны иметь четкое представление о развитии информационных технологий в своей предметной области; освоить основные приемы работы на персональных компьютерах; ознакомиться с основами сетевой технологии использования компьютеров; освоить приемы работы в основных службах сети Интернет.

Целью текстов лекций по разделу «Технология имитационного эксперимента на ЭВМ» является оказание помощи аспирантам и магистрантам в овладении основами имитационного эксперимента как инструмента для решения научных и практических задач в предметной области и их использования в практической деятельности.

Лекция 1 ИССЛЕДОВАНИЕ НА ВЕРОЯТНОСТНЫХ МОДЕЛЯХ СЛОЖНЫХ СИСТЕМ

1.1 Кибернетический подход к исследованию на вероятностных моделях сложных систем.

1.2 Концептуальная модель исследования вероятностной сложной системы.

1.3 Основные определения планирования экспериментов на ЭВМ с моделями вероятностных систем.

1.4 Планы для построения регрессионной модели поведения сложной системы.

1.5 Другие планы эксперимента на ЭВМ моделями сложных систем.

1.6 Планирование имитационного эксперимента.

1.1 Кибернетический подход к исследованию на вероятностных моделях сложных систем.

Обычно математические модели (ММ) представляют собой формализованную запись сложной системы и процессов, происходящих в объектах исследования. Они служат как для исследования свойств СС, так и для предсказания их поведения в различных ситуациях. Если исходить из соотношений в этих ММ, то обратим внимание на то, что эти соотношения, параметры объекта моделирования и исходная информация, поступающая на вход объекта исследования, могут быть детерминированными, вероятностными и смешанными. При построении ММ исследователю должны быть заранее известны все составляющие ММ и вся сложность состоит в определении самих этих соотношений и в получении явного вида функциональных зависимостей между компонентами векторов откликов системы $\{Y_t\}$, векторов состояний системы $\{S_n\}$, в каждый момент времени t при изменении компонент вектора параметров объекта $\{H_n\}$, компонент вектора параметров $\{x_n\}$ и вектора начальных условий функционирования объекта $\{U_0\}$. В таких случаях СС представляется в виде «черного ящика», на вход которого поступают значения компонент этих векторов, а на выходе его определяются значения компонент векторов откликов $\{Y_{jt}\}$ и векторов состояний СС $\{S_{nj}\}$. На начальный момент исследования внутренняя структура и взаимосвязи между компонентами СС еще не установлены. Поэтому стоит задача нахождения функциональных зависимостей для детерминированных ММ в ином виде (и таким образом превратить «черный ящик» в «белый ящик»):

$$\begin{aligned} \tilde{Y} &= \Phi(\tilde{X}, \tilde{U}, \tilde{H}), \\ \tilde{S}_n &= \Phi_s(\tilde{S}_{n-1}, \tilde{X}, \tilde{U}, \tilde{H}), \end{aligned} \quad (1.1)$$

РЕПОЗИТОРИЙ ГРУШИН

где i – номер нового состояния СС, $i-1$ – номер предыдущего состояния объекта исследования.

В вероятностных ММ (и тем более в имитационных моделях) с помощью соотношений (4.1) можно определить лишь распределение вероятностей для компонент вектора состояний $F_{1m}(S)$ и вектора откликов $F_0(Y)$, если известны распределение вероятностей для начальных условий $F_1(U)$ и параметров $F_{ik}(\Pi)$ и входной информации $F_0(x)$. Из-за вероятностного характера компонентов векторов $\{S_m\}$, $\{Y\}$, $\{U\}$, $\{X\}$ в явном виде трудно найти зависимости вида (1.1). Поэтому в ММ замеряют множество статистик моделирования и в явном виде находят зависимости:

$$ST_{st} = \phi_s(\bar{X}, \bar{U}, \bar{\pi}). \quad (1.2)$$

Затем, используя процедуру Монте-Карло, определяют математическое ожидание и выборочную дисперсию статистик моделирования:

$$\overline{ST_{st}} = \frac{1}{N} \sum_{n=1}^N ST_{st}; \quad D_{st} = \frac{1}{N} \sum_{n=1}^N (ST_{st} - \overline{ST_{st}})^2. \quad (1.3)$$

По этим интегральным статистикам непосредственно определяются значения компонент векторов откликов и состояний:

$$Y_t = \varphi_{st}(\overline{ST}); S_m = \phi_m(\overline{ST}). \quad (1.4)$$

Существует принципиальное различие при получении откликов с помощью детерминированных моделей вида (4.1) и вероятностных моделей вида (1.4). Если для детерминированных ММ отклики не зависят от момента времени, т.е. и они полностью определяются значениями компонент этих векторов. В вероятностных же ММ отклики зависят от момента времени и поэтому они предварительно усредняются согласно процедуре Монте-Карло с помощью зависимостей (1.3). Это означает, что переход от «черного ящика» к «белому ящику» гораздо сложнее. Поэтому в имитационных моделях важную роль играют «наблюдатели» за течением во времени процессов внутри самих имитационных моделей СС. Отсюда в ряде случаев выражает роль определения интегральных состояний S_m с помощью этих состояний во время имитации S_m . В этом и состоит основная идея кибернетического подхода к исследованию вероятностных СС.

Если исходить из способа дальнейшего использования ММ для изучения СС, то ММ можно разделить на три класса:

- аналитические, полностью определяемые выражением (1.1);
- численные модели, когда в общем виде вместо явных зависимостей из-за их сложной структуры стараются получить числовые результаты ре-

шения уравнений вида (1.1) при конкретных начальных расчетных данных компонент векторов \bar{X}, \bar{U} и $\bar{\pi}$;

- имитационные модели, когда СС сложна, имеет вероятностную структуру и требуется получение результатов вида (1.4), или же необходим высокий уровень детализации представления процессов СС, возникающих во времени t_0 , в самой СС.

Для аналитических ММ процессы функционирования СС записываются в виде некоторых функциональных соотношений (алгебраических и интегро-дифференциальных). При этом всегда стремится в общем виде получить явные зависимости вида (1.1). Поэтому предсказательные возможности аналитических ММ очень велики. Если математические зависимости вида (1.1) сложны, то зачастую используется ЭВМ, позволяющая быстро вычислить значения компонент векторов \bar{S} и \bar{Y} для любых заданных значений X, U, π или их распределения (для вероятностных ММ).

Для численных ММ используются численные методы решения зависимостей φ_1 и φ_2 в (1.1). Причем, в явном виде эти зависимости удается получить на практике весьма редко. Поэтому исследователи вначале стремятся получить аналитическое решение задачи. При этом они умышленно идут на упрощение реальной ситуации, чтобы иметь возможность изучать некоторый общий свойства СС. В отдельных случаях приближенное решение задачи о поведении СС исследователей удовлетворяет. Обычно для них достаточно результатов, полученных с помощью качественных методов на численной модели. Для этой цели создан мощный математический аппарат (алгебра интегральные и разностные уравнения, теория вероятностных процессов, численные методы: методы оптимизации и т.д.)

В имитационных моделях моделируемый алгоритм поведения СС приближенно воспроизводит сам процесс-оригинал в смысле его функционирования во времени. При этом имитируются элементарные явления, составляющие процесс в СС, с сохранением их логической структуры и порядка его протекания во времени. Таким образом, на ЭВМ реализуется специальный алгоритм, который воспроизводит формализованный процесс поведения СС. Этот алгоритм по исходным данным $(\bar{U}, \bar{\pi}, \bar{\lambda})$ позволяет получить статистику об изменениях во времени t сначала ST , а затем вычислить значения компонент векторов \bar{S} и \bar{Y} . В этом алгоритме можно выделить три функциональные части: моделирование элементарных процессов; учет их взаимодействия и объединение их в единый процесс; обеспечение согласованной работы отдельных компонентов ИМ СС на ЭВМ. Влияние случайных факторов на течение процессов в ИМ СС имитируется с помощью ГСЧ с заданными вероятностными характеристиками. В ходе имитации постоянно фиксируется: статистика (\bar{ST}_t) , состояния системы (\bar{S}) и характеристики вектора откликов (\bar{Y}) . Эта статистика должна образом обрабатывается в ходе имитации, либо накапливается и по окончании задан-

ного интервала моделирования (T_h) обрабатывается статистическими методами. Как видим, идея имитации привлекательна по своей простоте, но дорога по реализации. Поэтому применяются ИМ только в тех случаях, когда другие способы моделирования СС неэффективны.

1.2 Концептуальная модель исследования вероятностной сложной системы.

Поскольку моделирование вероятностей СС реализуется с помощью имитационных или вероятностных моделей, построенных на основе многократного использования процедуры Монте-Карло, то особенную роль приобретает планирование имитационных экспериментов (ИЭ), позволяющее уменьшить ресурсоемкость исследований на ЭВМ. Перед построением планов натурных экспериментов (НЭ) с реальной СС или же перед построением планов ИЭ необходимо составить концептуальную модель исследование (КМИ). Блок-схема КМИ вероятностной СС приведена на рис. 4.1, построенная на основе использования кибернетического подхода к исследованию вероятностной СС.

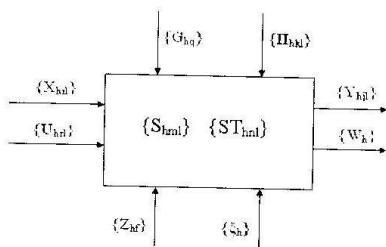


Рис. 1 - Блок-схема концептуальной модели исследования вероятностной СС

Согласно этому подходу вероятностная СС представлена «черным ящиком» на входы которого во время 1-ой реализации процедуры Монте-Карло ($i = \overline{1, N}$) действуют векторы: $\{X_{hi}\}$, $\{U_{hi}\}$, $\{\Pi_{hi}\}$, $\{G_{hi}\}$, $\{\xi_i\}$.

Здесь $\{X_{hi}\}$ -значение вектора входной информации 1-ой реализации, формируемые по функциям распределения $F_S(x)$ в h -ом варианте исследования СС;

$\{U_{hi}\}$ -вектор начальных условий моделирования h -го варианта СС 1-ой реализации модели, формируемый по функции распределения $F_u(U)$:

$\{\Pi_{hi}\}$ -вектор параметров в h -го варианта модели СС в 1-ой реализации модели, формируемый по функции распределения $F_{\Pi}(Pi)$:

$\{G_{hi}\}$ -множество постоянных характеристик модели СС, определяющих обычно влияние внешней среды, в которой функционирует исследуемая СС;

$\{\xi_i\}$ -вектор неконтролируемых случайных воздействий внешней среды о которых известно только об их существовании и их влияние оценивается суммарной ошибкой моделирования СС. В ходе моделирования 1-ой реализации ИМ фиксируется вектор статистик имитации $\{ST_{hmi}\}$ - h -го варианта модели СС. Для каждой компоненты вектора по выборке статистик определяются по формуле (4.3) математическое ожидание и выборочная дисперсия значений статистик $\{ST_{hmi}, D_{ST_{hmi}}\}$.

В общем случае значений компонент вектора, наряду с фиксацией статистик моделирования в 1-ой реализации h -го варианта модели фиксируется множество состояний модели СС $\{S_{hmi}\}$ в моменты времени i .

После усреднения согласно процедуре Монте-Карло определяется множество наиболее вероятных состояний системы в моменты времени i $\{\bar{S}_{hmi}\}$. Используя \bar{S}_{hmi} , формируются временные диаграммы (ВДГ) перехода модели системы в различные состояния. В совокупности ВДГ позволяют следить исследователю за динамикой функционирования СС.

По значениям ST_{hmi} с помощью формул (4.4) определяются значения компонентов вектора откликов h -го варианта 1-ой реализации модели СС $\{Y_{hi}\}$. Согласно процедуре Монте-Карло по завершении N реализаций ИМ СС по выборкам $\{Y_{hi}\}$, $i = \overline{1, N}$ определяются оценки математического ожидания и дисперсии компонентов вектора откликов по формулам:

$$\bar{Y}_h = \frac{1}{N} \sum_{i=1}^N Y_{hi}; D_{Y_h} = \frac{1}{N} \sum_{i=1}^N (Y_{hi} - \bar{Y}_h)^2. \quad (1.5)$$

Для сравнения друг с другом вариантов h модели СС необходимо «вернуть» вектор \bar{Y}_h к скалярной величине W_h по формуле:

$$W_h = Z(\bar{Y}_h). \quad (1.6)$$

Способы «вертки» вектора \bar{Y} к скалярной величине и выбор оптимального варианта h_0 модели СС рассматриваются в главе 6. В итоге обработки статистики имитации и вычисления откликов моделирования и обобщенного формируются интегральные характеристики модели и временные диаграммы изменения статистик и состояний модели системы.

Имея значения характеристик КМИ, приведенные на рис. 1.1, исследователь может предварительно оценить изменения во времени в системе и сравнивать варианты друг с другом по обобщенному показателю $W(Z)$ для каждой компоненты вектора откликов \vec{Y} .

1.3 Основные определения планирования экспериментов на ЭВМ с моделями вероятностных систем.

С точки зрения теории ПЭ **входными наблюдениями и управляемые** переменными модели являются компоненты векторов и множеств $\{X_n\}$, $\{U_n\}$, $\{P_n\}$ поскольку исследователь может варьировать их значениями по своему усмотрению. На систему (или на ее модель) может воздействовать множество неуправляемых переменных $\{Z_n\}$, поскольку их значения не зависят от исследователя, но в модели СС они должны быть также представлены в тех случаях, когда их влияние на поведение системы и ее отклики является значимым. Множества $\{G_n\}$ и $\{U_n\}$ можно также отнести к неуправляемым переменным, поскольку исследователь должен задать их значения такими, какими они были при исследовании реальной СС. Множество $\{\xi_n\}$ также представляет собой неуправляемые параметры, но их влияние исследователь должен оценить, определив при этом общую ошибку имитации. Входными характеристиками ИМ являются вектор откликов $\{Y_n\}$ и обобщенный показатель W_n . Назовем $\{X_n\}$ и $\{P_n\}$ факторами. Конкретное значение каждой компоненты этих векторов назовем уровнем фактора.

Первичными факторами назовем те входные переменные и параметры модели в изучении влияния которых исследователь заинтересован. Остальные факторы назовем **вторичными**, поскольку их влиянием нельзя пренебречь. Однако, качественная оценка этого влияния не входит в цели исследования СС. Наоборот, исследователь зачастую пытается ослабить это влияние путем специального подбора плана эксперимента (ПЭ), чтобы выделить эффекты первичных факторов. Неуправляемые переменные вместе с $\{\xi_n\}$ вносят случайные помехи.

Моделирование СС может преследовать следующие цели:

- определение вида зависимости (4.4) по данным натурного или модельного эксперимента;
- нахождение такого сочетания параметров и входных переменных, которое обеспечивает максимум выражения (4.6);
- изучение динамики изменения во времени состояний СС в виде множества $\{S_n\}$ и поиска узких мест в структуре модели СС.

В первом случае речь идет об определении регрессионной зависимости, либо по данным натурного эксперимента (ИЭ), либо по данным модельного эксперимента (ИЭ). Зависимости (4.4) уточняются с помощью системы регрессионных моделей:

$$Y_j = \phi_{jm} [\vec{X}, \vec{U}, \vec{G}, \vec{U}], \theta, \quad (1.7)$$

где $\theta = (\theta_0, \theta_1, \theta_2, \dots, \theta_d)$ – вектор параметров регрессионной модели, подлежащих определению.

Как правило, вид таких моделей инут среда компонентов вектора Y :

$$\cdot Y = b_0 \text{ (полином нулевой степени);} \quad (1.8)$$

$$\cdot Y = b_0 + b_1 + x_1 + b_3 + x_2 \text{ (полином первой степени);} \quad (1.9)$$

$$\cdot Y = b_0 + b_1 x_1 - b_2 x_2 - b_{12} x_1 x_2 + b_{11} x_1^2 + b_{22} x_2^2 \text{ (полином второй степени);} \quad (1.10)$$

Если вид полинома выбран, о необходимо составить план эксперимента и затем осуществить либо НЭ, либо МЭ. Отметим, что любой эксперимент представляет собой систему операций с реальной системой или ее моделью, возможностей и наблюдений, которые в совокупности направлены на получение информации о СС в ходе испытаний. Опыт является отдельной элементарной частью эксперимента. Он воспроизводит исследуемое явление в СС при определенных условиях проведения эксперимента (составе параметров модели) и при возможности регистрации результатов НЭ или МЭ, которые осуществляются согласно планов эксперимента. В общем случае **план эксперимента** (ПЭ) представляет собой совокупность данных, определяющих число, условия и порядок реализации опытов. Поэтому планирование эксперимента устанавливает совокупность действий, направленных на разработку тактики эксперимента от начальных (выбор НЭ) до заключительных этапов изучения СС (обработка экспериментальных данных и создание математической модели СС). ПЭ должен обеспечивать максимум информации при минимуме опытов на СС. При ПЭ модель СС должна удовлетворять двум основным требованиям:

- воспроизводимость на объекте исследования результатов эксперимента, т.е. разброс значений отклика при повторении опытов не превышает точности эксперимента;

- управляемость объекта исследования либо непосредственно, либо с помощью модели.

Исходя из второго требования, эксперимент бывает пассивным и активным. При пассивном эксперименте существуют только неуправляемые факторы. В этом случае осуществляется только регистрация статистик ST_i , состояний S_i , откликов Y_i , без каких-либо возможностей функционирования СС. В таких случаях решаются только задачи идентификации явлений в СС. Активные эксперименты является основным объектом планирования экспериментов. Они проводятся при условии, что существуют только управляемые параметры (\vec{X} и \vec{U}) и исследователь изменяет компоненты этих векторов целенаправленно в соответствии с ПЭ. Для активного эксперимента область действия определяется областью возможных значений компонент векторов \vec{X} и \vec{U} . Если область планирования задается интервалами

возможного изменения факторов $X_{i_{\min}} \leq X_i \leq X_{i_{\max}}, i = \overline{1, n}$, то она представляет собой при соответствующем масштабировании гиперкуб.

Точка плана означает упорядоченную совокупность численных значений факторов, соответствующих условиям проведения опыта. Точка плана с номером g соответствует вектор-строке:

$$X_g^T = \|X_{1g}, X_{2g}, \dots, X_{ng}\| \quad (1.11)$$

Общая совокупность таких векторов $X_g, g = \overline{1, l}$, образует ПЭ, а совокупность различных векторов определяет спектр плана. Вводится понятие об уровнях фактора. Основной уровень фактора $X_i^0, i = \overline{1, n}$ указывает такие условия эксперимента, которые представляют основной интерес для в данный момент времени. Обычно вектор-строка $(X^0) = \|X_1^0, X_2^0, \dots, X_n^0\|$ задает в факторном пространстве точку, являющуюся центром области планирования. Если координаты $X_i^0 = (X_{i_{\min}} + X_{i_{\max}})/2$, а интервал (шаг) формирования фактора X_i равен $\Delta X_i = (X_{i_{\max}} - X_{i_{\min}})/2$, то

$$X_{i_{\max}} = X_i^0 + \Delta X_i, \quad X_{i_{\min}} = X_i^0 - \Delta X_i. \quad (1.12)$$

В практике ПЭ осуществляется операция нормализации фактора, которая сводится к изменению начала отсчета координатных осей и их масштаба в соответствии с соотношением:

$$X_i = (X_i - X_i^0)/\Delta X_i, \quad (1.13)$$

где $i = \overline{1, N}$; N – число факторов

Для переменной X_i в стандартизованном масштабе начало координат совмещено с центром эксперимента, а в качестве единицы измерения используется шагарьрования фактора. Вклад фактора в величину отклика Y , при переходе от минимального (нижнего) к максимальному (верхнему) уровню называется эффектом фактора (основным или главным эффектом фактора). Эффект фактора может зависеть от уровня, на котором находится другой фактор. При этом имеет место эффект взаимодействия факторов.

Для организации ПЭ вначале составляется матрица плана, которая представляет собой стандартную форму записи условий проведения эксперимента в виде прямоугольной таблицы, строки которой отвечают номенклатурой опытов, а столбцами являются типы факторов. Размер матрицы плана равны $(L \times n)$, где L – число опытов, n – число факторов. Она может иметь совпадающие строки. Каждый (ij) -й элемент матрицы плана равен уровню j -го фактора в i -ом опыте. Матрица спектра плана составляется из всех строк матрицы плана и отличается наличием значений уровня хотя бы одного фактора. Поэтому размер матрицы спектра равен $N \cdot n$. Все строки

этой матрицы различны. Матрица дублирования представляет собой квадратную диагональную матрицу, у которой диагональные элементы ее равны числам параллельных опытов в соответствующих точках спектра плана и имеет вид:

$$R = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & n \end{pmatrix} \quad (1.14)$$

Итак, ПЭ может быть задан либо матрицей плана, либо матрицей спектра плана в совокупности с матрицей дублирования. Во втором случае речь идет о выборе такого сочетания уровней факторов $\{X_i\}$, при котором обеспечивается максимум обобщенного отклика W . Здесь возможны два случая. Либо вид функции (1.6) известен и тогда осуществляется поиск экстремума одним из способов векторной оптимизации, либо зависимость (1.6) неизвестна и исследователь ограничен конечным числом комбинаций уровневых факторов. В такой ситуации исследователь имеет только возможность простого перебора вариантов сочетания факторов и задача сводится к выбору той комбинации факторов, которая обеспечивает максимум обобщенного показателя качества вариантов организации СС (таких W), вычисляемому из выражения (1.6). В такой ситуации говорить об оптимизации структуры СС нельзя и поэтому мы будем говорить о выборе рационального варианта сочетания факторов (лучшего в смысле критерия (1.1) из рассмотренных вариантов).

В третьем случае речь идет о способе «свертки» вектора состояний СС $\{S_{nt}\}$, который позволяет при необходимости в любой момент времени t воспроизвести диаграмму состояний системы и на основе анализа этой диаграммы определить «кузки места» функционирования СС. Наиболее частым способом такой «свертки» $\{S_{nt}\}$, является построение матрицы вероятностей переходов СС из состояния m в состояние f $\|P_{mf}\|$ предполагая при этом возможность представления смен состояний СС как полумарковского процесса. Нахождение длительностей T_{mf} нахождения СС в состоянии m при переходах системы из состояния f описывается функцией распределения $\Gamma_{mf}(t)$. Поэтому, кроме матрицы вероятностей $\|P_{mf}\|$ для описания переходов СС как полумарковский процесс необходимо задание матрицы, элементами которой будут функции распределения $\|\Gamma_{mf}(t)\|$, а также задать начальные условия для таких переходов (U) и способов завершения полумарковского процесса (либо это будет вероятность перехода в конечное состояние P_{mf} , либо это будет общее число состояний $\{v_k\}$ исследуемой

СС). Ниже в данной главе мы рассмотрим более подробно все три случая анализа результатов эксперимента с моделью СС.

1.4 Планы для построения регрессионной модели поведения сложной системы.

В основе построения таких моделей лежит регрессионный анализ, позволяющий строить модели вида (1.8) и (1.10). Для использования регрессионного анализа нет необходимости различать факторы управляемые (X) и неуправляемые (Z). Поэтому полагаем, что Z входит в X . Кроме того, поскольку нельзя обнять необъятное ограничимся лишь наиболее распространёнными линейными по параметрам регрессионными моделями. Для линейных регрессий используются планы первого порядка. Такие планы используются для экспериментального получения регрессионных моделей вида:

$$\phi(x) = b_0 + b_1 x_1 + b_2 x_2 + \dots + b_n x_n, \quad (1.16)$$

$$\text{где } b_i = \frac{\beta_i}{\Delta X_i}; b_0 = \beta_0 + \sum_{i=1}^N \beta_i X_i^0, -1 \leq X_i \leq +1; i = \overline{1, n}.$$

Известно несколько разновидностей планов первого порядка: одинофакторный эксперимент (ОЭ); полный факторный эксперимент (ПФЭ); дробный факторный эксперимент (ДФЭ); насыщенный план (симплекс-план). Рассмотрим суть каждого из этих типов планов.

Одинофакторный эксперимент. Он предусматривает поочередное варирирование каждого из факторов, в то время как все остальные факторы стабилизированы на некотором уровне (чаще всего на нулем). Можно считать, что в ходе варирирования фактор X_i принимает значения $X_{iH} = X_i^0 + \Delta X_i$ и $X_{iL} = X_i^0 - \Delta X_i$, а все другие факторы фиксированы на своем базовом уровне $X_j^0 (j \neq i)$. Для нормированных величин $X_{iH}=+1; X_{iL}=-1$. Матрица спектра должна ОЭ иметь вид:

$$X = \begin{bmatrix} -1 & 0 & \dots & 0 \\ +1 & 0 & \dots & 0 \\ 0 & -1 & \dots & 0 \\ 0 & 0 & \dots & -1 \\ 0 & 0 & \dots & +1 \end{bmatrix}; \quad (1.17)$$

Геометрический смысл спектра данного плана для двумерного случая означает, что точки располагаются на серединах сторон ограничивающего

квадрата со сторонами равными 1. Обычно ОЭ редко используется в силу низкой его точности и отсутствия возможности оценить эффект взаимодействия факторов.

Полный факторный эксперимент 2^n . Спектр ПФЭ содержит в себе все возможные комбинации п факторов на всех уровнях их изменения. Если рассматривать вариант, когда каждый из п факторов меняется на двух уровнях (что достаточно для построения линейной регрессии), то общее число элементов спектра плана равно 2^n . Такой план обозначают в виде ПФЭ 2^n . Например, ПФЭ 2^2 имеет следующий спектр плана:

$$X = \begin{bmatrix} -1 & -1 \\ +1 & -1 \\ -1 & +1 \\ +1 & +1 \end{bmatrix}; \quad (1.18)$$

Отметим, что геометрически точки этого плана размещаются в вершинах квадрата с единичными ребрами.

С помощью ПФЭ 2^n возможно получение регрессионных моделей вида:

$$\begin{aligned} \phi(x) = b_0 &+ \sum_{i=1}^n b_i x_i + \sum_{i_1=1}^{n-1} \sum_{i_2=i+1}^n b_{i_1, i_2} x_{i_1} x_{i_2} + \\ &\dots \\ &+ \sum_{i_1=1}^n \sum_{i_2=i_1+1}^n \sum_{i_3=i_2+1}^n b_{i_1, i_2, i_3} x_{i_1} x_{i_2} x_{i_3} + \dots + b_{1, 2, \dots, n} x_1 x_2 \dots x_n; \end{aligned} \quad (1.19)$$

Как правило, такие модели содержат:

- один коэффициент b_0 (свободный член уравнения регрессии);
- п коэффициентов b_i (линейных членов уравнения регрессии);
- C_n^2 коэффициентов b_{i_1, i_2} (при парных взаимодействиях факторов);
- C_n^3 коэффициентов b_{i_1, i_2, i_3} (при тройных взаимодействиях факторов);
- и так далее, включая один коэффициент $b_{1, 2, \dots, n}$ при взаимодействиях факторов максимального n -го порядка.

Общее число коэффициентов регрессии в этой модели равно $2^n = 1 + n + C_n^2 + C_n^3 + \dots + C_n^n$, т.е. по его отношению к регрессии план 2^n является насыщенным. План называется **насыщенным**, если общее число наблюдений равно числу неизвестных параметров регрессионной модели. Желательно, чтобы любой реальный план был близок к насыщенному. Это важно на этапе предварительного исследования, когда требуется получить хотя бы представление об объекте исследования, но зато с минимальными затратами ресурса опытов. Различают четыре типа свойств матриц планирования:

1) *симметричность* относительно центра эксперимента, означающее, что алгебраическая сумма вектор-столбца каждого фактора равна нулю:

$$\sum_{i=1}^k X_p = 0, \quad (1.20)$$

где $i=1, k$ -номера вектора; L-общее число факторов;

2) *условие нормировки*, означающее, что сумма квадратов элементов каждого вектор-столбца равна числу опытов:

$$\sum_{i=1}^L X_q^2 = L; \quad (1.21)$$

3) *ортогональности*, показывающее, что сумма почленных произведений любых двух вектор-столбцов матрицы плана равна нулю:

$$\sum_{i=1}^L X_{ij} X_{iu} = 0; \quad j \neq u; \quad u = 0, n. \quad (1.22)$$

4) *ротатабельность*, предполагающее, что точки в матрице планирования подбираются таким образом, чтобы точность предсказания значений отклика была одинакова на разных расстояниях от центра эксперимента и не зависела от направления движения в факторном пространстве. Свойство ортогональности дает возможность снизить вычислительные трудности расчета коэффициентов регрессии. Ротатабельность обеспечивает равномерность распределения линсерии предсказанных значений отклика в изучаемой области факторного пространства. В этом смысле ПФЭ 2^n оптимальен при построении линейной модели. Недостатком ПФЭ 2^n является быстрый рост количества опытов при увеличении n , например при $n=10$ число опытов превышает 1024. Как правило, исследователь ограничивается на первых порах линейной моделью с учетом дополнительных парных взаимодействий. Количество опытов в ПФЭ значительно превосходит число определяемых коэффициентов линейной модели. Поэтому ПФЭ обладает большой избыточностью и целесообразно сократить число опытов за счет использования той информации, которая не очень существенна при построении линейных моделей.

Дробный факторный эксперимент 2^{n-p} . Дробный факторный эксперимент позволяет получить линейные регрессионные модели, а также модели, содержащие дополнительно некоторые взаимодействия факторов при меньшем числе наблюдений по сравнению с ПФЭ. Подобная экономия числа наблюдений достигается за счет использования априорной информации о свойствах СС. Эта информация должна быть сформулирована в виде списка переменных-факторов и их взаимодействий. Например, X_0, X_1, X_2, X_3 . Здесь X_0 – фиктивная переменная, соответствующая коэффициенту b_0 , в линейной регрессионной модели. Поэтому наибольшее распространение получили регулярные планы ДФЭ типа 2^{n-p} , где n - число факторов, p - степень дробности ПФЭ. Соответствующие типы ДФЭ называют репликами с указанием их степени дробности. Например, ДФЭ 2^{n-1} означает, что исследователем получена $\frac{1}{4}$ реплика ПФЭ 2^n . Аналогично ДФЭ 2^{n-2} означает $\frac{1}{4}$ реплика ПФЭ 2^n . *Регулярность* плана ДФЭ 2^{n-p} состоит в том, что матрица спектра плана является некоторой неслучайной частью матрицы спектра типа ПФЭ 2^n .

Основная идея ДФЭ состоит в том, чтобы использовать такой экономичный план, который обеспечивал бы оценку влияния только переменных, находящихся в списке существенных, и не учитывал возможное воздействие прочих переменных. Если список существенных переменных содержит $(d+1)$ элемент, то для того, чтобы оценить по отдельности коэффициенты регрессии при каждой из них, необходимо выполнение условия $N=2^{n-1} \geq d+1$. Это условие при заданном числе факторов n и известном числе существенных переменных $d+1$ позволяет найти минимально допустимую степень дробности p . Тем самым и определяется и общее число строк в матрице спектра плана ДФЭ. ДФЭ строится следующим образом. Для (p) факторов, условно называемых основными строятся матрица ПФЭ, а для r факторов, называемых дополнительными, уровни выраживания в опытах выбираются на основании априорной информации. Генеральное соотношение – это формальное равенство, показывающее, знак каких основных переменных, стоящих в правой части равенства (например, $X_3=-X_1 X_2$), необходимо перемножить для получения знака дополнительного фактора (уровня вариации), чтобы матрица ДФЭ оказалась ортогональной, нормированной и симметричной. Можно, например, построить и другую реплику ПФЭ $X_3=-X_1 X_2$. Аналогичным образом для ПФЭ 2^3 можно выбрать две 1/2 реплики, не имеющих общих строк. Поэтому, подобная разновидность планов получила широкое практическое применение.

Насыщенный план первого порядка называется план, который позволяет оценить коэффициенты линейной модели относительно n факторов, а его спектр содержит $(n+1)$ точку. Суммарность $n+1$ точек насыщенного линейного плана задает в n -мерном факторном пространстве вершины геометрической фигуры, называемой n -мерным симплексом и поэтому подобные планы называются симплекс-планами. Это простейшая фигура в евклидовом пространстве. Например, одномерный симплекс – это отрезок прямой, двумерный симплекс – треугольник, n -мерный – выпуклый многоугольник с $(n+1)$ вершинами. Симплекс-планы используются, как правило, на стадии предварительного исследования СС, когда желательно применить наиболее экономичные насыщенные планы, а также при построении процедуры поиска экстремальных значений целевой функции.

БІБЛІЯТЭКА

При практической реализации планов первого порядка исследователь выполняет следующие действия:

- 1) выбирает центр плана X^0 , т.е. значения базовых уровней $x_i^0, i=1..n$;
- 2) определяет шаги варьирования Δx по каждому фактору (обычно $\Delta x = 0,3; 0,5$ допустимого диапазона изменения фактора x_i);
- 3) устанавливается общее число опытов, т.е. формируется план эксперимента в целом, что включает также построение спектра плана и выбор элементов матрицы дублирования $R(1,4)$. В планировании первого порядка применяются, как правило, равномерное дублирование опытов, т.е. в матрице R все диагональные элементы равны между собой: $r_1=r_2=\dots=r_N=m$. Обычно $m=2,4$, а общее число опытов равно N_m . Причем, эксперименты реализуются отдельными сериями. Каждая серия включает проведение N опытов, соответствующих всем строкам матрицы спектра плана. Количества серий равно m . Внутри каждой серии порядок реализации опытов должен быть случайным. Для этой цели процедура *randomизация*, т.е. обеспечивается случайный порядок реализации строк матрицы плана. Таким способом устраивается в математических моделях смещения, вызванные действием неконтролируемых систематических переменных.

Конечным результатом проведения эксперимента является табл. I значений отклика Y . Данные, полученные с помощью планов первого порядка, сосредоточенные в табл. I, обрабатываются согласно семи-шагового алгоритма.

Таблица I - Представление результатов эксперимента

Серия опытов				
Y_1	Y_2	Y_3	Y_4	
-	-	-	-	Y_{m_1}
-	-	-	-	Y_{2m_1}
-	-	-	-	Y_{3m_1}
-	-	-	-	Y_{4m_1}
Y_{N_1}	Y_{N_2}	Y_{N_3}	Y_{N_4}	Y_{Nm_1}

Шаг 1. Проверка воспроизводимости эксперимента. Находят средние значения и выборочную дисперсию:

$$\bar{Y} = \sum_{i=1}^m Y_i / m ; S_g^2 = \sum_{i=1}^m (Y_{gi} - \bar{Y})^2 / (m-1). \quad (1.23)$$

Проверяется гипотеза $H_0: \delta_1^2 = \delta_2^2 = \dots = \delta_k^2$, где $\delta_g^2, g = 1, l$ значения дисперсии плума, соответствующие спектрам плана.

Эта проверка проводится с помощью критерия Кохрена, статистика которого имеет вид:

$$G = S_{\max}^2 / \sum_{g=1}^N S_g^2 . \quad (1.24)$$

где S_{\max}^2 - наибольшая из оценок $S_g^2; g = 1, N$.

Шаг 2. Вычисление оценок коэффициентов регрессии по методу наименьших квадратов.

Шаг 3. Определение дисперсии оценок коэффициентов регрессии.

Шаг 4. Проверка значимости коэффициентов регрессии.

Шаг 5. Расчет предсказанных по уравнению регрессии значений отклика в точках спектра (4.23) и анализ точности предсказания.

Шаг 6. Проверка адекватности уравнения регрессии.

Шаг 7. Проверка работоспособности регрессионной модели.

Перед проведением эксперимента устанавливается каких из факторов нужно включать в состав (4.23), которые влияют на выходную величину отклика СС. При этом определяют минимальный набор факторов. Если число предлагаемых факторов невелико (порядка 6-8), то для предварительного изучения СС можно применить ПФЭ или ДФЭ. Далее определяют оценки коэффициентов модели и проверяют их статистическую значимость. По абсолютному значению факторов осуществляется их ранжирование согласно степени влияния на величину отклика.

Планы второго порядка используются для получения регрессионной модели в виде полинома второй степени. Они используются тогда, когда план первого порядка не позволяет получить адекватную регрессионную модель или априори известно, что регрессия обладает существенно нелинейными свойствами. Квадратичная модель содержит $k = 1 + 2n + C_g^2 = ((n+1)(n+2))/2$ членов, что в $(n+2)/2$ раз больше, чем для линейной модели. Соответственно возрастает и минимальное необходимое количество точек в спектре плана. Для получения квадратичной зависимости каждый из факторов должен изменяться, по крайней мере, на трех уровнях. Это вызывает необходимость постановки большого числа опытов. Полный факторный эксперимент содержит 3^n точек. Например, для четырехфакторного эксперимента число точек плана в пять раз больше, чем в плане первого порядка.

На практике часто используется *центральный композиционный план* (ЦКП), позволяющий сократить число точек до $L = n_c + 2n + n_b$, где n_c - число точек ПФЭ 2^{n_c} , или дробной реплики; n_b - число опытов в центре плана. План называется композиционным, если в его спектр в качестве составной части входят точки спектра плана, который был реализован при построении первого порядка.

ния более простой модели. Это дает возможность применить точки спектра плана одного из предшествующих этапов исследования в качестве точек части спектра плана следующего этапа, если модель, полученная на последнем этапе (например, полной степени k), не устраивает исследователя и требует усовершенствования (например, перехода к полному степени $k+1$). ЦКП состоит из трех частей. Первая часть – ядра плана – это ПФЭ 2^n или ДФЭ 2^n . Вторая часть ЦКП – «звездные» точки, расположенные на координатных осях на расстоянии $\pm d$ от центра эксперимента. Общее число таких точек равно 2^n . Третья часть ЦКП – постановка опытов в центре плана (число таких опытов $= 1$). Например, если рассматривать трехфакторную модель, то идея ЦКП состоит в том, что ставят серию из 8 опытов, располагая опыты в вершине куба, т.е. реализуется линейный план ПФЭ 3^3 . Затем к этим точкам добавляют еще шесть «звездных» точек, находящихся на линиях, которые выходят из центра куба. В центре ставят еще одну точку. В результате общее количество равно 15. Для ПФЭ 3^3 равно 27. Как видим, удается сэкономить почти половину опытов. В общем случае ЦКП не ортогонален. Ортогональность достигается специальным образом выбранным «звездным» планом λ и преобразованием квадратичных значений факторов. Недостатком ЦКП является неравномерная точность математической модели в факторном пространстве. Оптимальным планированием второго порядка считается *ротатабельное планирование*. Такие планы годятся для любого числа факторов. Их можно построить, комбинируя вместе вершины правильных геометрических фигур и центральные точки. Использование ротатабельных планов гарантирует, что в любой точке пространства факторов стандартное отклонение отклика зависит только от расстояния этой точки до центра и не зависит от направления. Построение ротатабельных планов требует высокой квалификации исследователя. Отметим, что композиционное планирование представляет собой один из возможных способов построения ротатабельных планов, в этом случае выбирают такое «звездное» плечо, при котором $n_c = 2^n$ и $\lambda = 2^{n-4}$. Такое планирование называют центральным ротатабельным униформ-планированием. При этом величина, обратная дисперсии отклика, остается постоянной в области факторного пространства, ограниченной радиусом, равным единице.

1.5 Другие планы эксперимента на ЭВМ моделями сложных систем.

Решение большого числа задач управления объектами связано с оптимизацией, т.е. нахождением наилучших значений факторов по отношению к отклику. Поиск экстремума функции отклика происходит путем исследования поверхности отклика. Эти исследования осуществляются посредством измерения поверхности отклика в различных точках факторного пространства. Возникает вопрос: какой должна быть стратегия планирования

эксперимента, чтобы число опытов, необходимых для нахождения экстремума обобщенного отклика W или близких к нему значений, было бы как можно меньше. Решение этой задачи обычно осуществляется в два этапа:

- осуществляются поисковые движения к области экстремума;
- уточняются расположения экстремальной точки либо с помощью дополнительных поисковых опытов, либо с помощью математической модели СС в области экстремума откликов, которая определяется с помощью организованных опытов.

Сами по себе поисковые методы определения экстремума при активном эксперименте бывают двух типов: классические и факторные. Среди классических методов поиска оптимальной области наиболее старым является метод Гаусса-Зейделя, при котором все факторы, кроме одного, поочередно фиксируются, т.е. эксперимент проводится по схеме одиночного эксперимента. Ставится серия опытов при различных значениях незадействованного фактора. Двигаясь параллельно одной из осей факторного пространства, исследователь находит наилучшее значение для рассматриваемого разреза поверхности отклика. Затем в этой наилучшей точке ставится следующая серия опытов при различных значениях следующего фактора (остальные факторы при этом являются зафиксированными). Таким образом, осуществляется движение параллельно следующей от факторного пространства. Последовательное прохождение всех осей факторного пространства составляет первый цикл исследования. Этой процедуре повторяют до получения оптимума или до попадания в некоторую точку, любое движение из которой ухудшает значение отклика. Данный метод требует большого количества опытов.

Применение градиентных методов позволяет быстрее отыскать экстремум. Широко распространён метод *кругового восхождения*. Основная идея этого метода заключается в том, что на начальном этапе на основании ПФЭ и ДФЭ получают простейшие линейные модели в качестве приближенного описания некоторой части функции отклика (может быть далекой от экстремума). Коэффициенты модели, полученной в результате ПФЭ или ДФЭ, пропорциональны проекциям вектор-градиента и позволяют определить само направление градиента, т.е. направление кругового склона поверхности отклика $W(z)$. Затем вдоль этого направления совершаются постепенное пошаговое движение в области экстремума. При достижении области экстремума проводится ее исследование. В этом случае строятся планы более высокого порядка, так как поверхность отклика $W(z)$ вблизи экстремума обычно плохо аппроксимируется гиперплоскостью. Поэтому используются центральные композиционные и ротатабельные планы второго порядка.

На практике исследователь не имеет информации о виде функции отклика $W(X, \pi, z)$. В таких случаях за центр плана принимается множество значений входной величины $\{X_0\}$, имевшее место при натурном экспери-

менте на СС при фиксированных значениях вектора параметров π и существующих неуправляемых воздействий $\{z\}$ на СС. Кроме того, число возможных комбинаций уровней фактора $\{x_i\}$ ограничено и поэтому речь может идти только о выборе наилучшей комбинации уровней факторов x_i , т.е. определяется *рациональный состав комбинаций $\{x_i\}$* . Здесь может помочь метод «секущих плоскостей» суть которого заключается в следующем. Во-первых, вместо трех уровней (x_1^0, x_1^1, x_1^2) рассматривается еще два дополнительных уровня, означающих, что между центром фактора и крайними значениями уровней выбираются по два уровня $(x_1^-, x_1^0, x_1^0, x_1^+)$. Поэтому число всевозможных комбинаций уровней факторов резко возрастает, но из них выбирается целенаправленно те подбирают комбинации, которые практически достигны. Во-вторых, фиксируют все факторы кроме одного, который варьируется на уже 5 уровнях. Например, пусть изменяется x_1 на пяти уровнях, а для всех остальных факторов фиксируются значения центральной точки плана $(x_{ij}^0, i \neq j)$. В третьих, из всех 5 уровней фактора x_1 выбирают то значение (например, x_n), которое обеспечивает максимум $W(z)$. Для фактора x_1 фиксируют величину его уровня значением x_{1j} . Далее переходят к варьированию уже фактора x_1 (i -й фактор зафиксирован на уровне x_{1j}). Для всех остальных x_i уровни остаются фиксированным в центре области пространства параметров. Далее, осуществив 5 вариаций фактора x_1 , находят тот уровень (например, пусть это будет x_{1k}), который обеспечивает увеличение $W(z)$ и фиксируют это значение фактора x_1 . Теперь уже два фактора зафиксированы соответственно на уровнях x_{1j} и x_{1k} , а остальные факторы находятся на уровнях x_{ij}^k , ($k = i, k \neq j$). Эту процедуру повторяют до тех пор, пока не будут модифицированы уровни всех факторов. Если позволяет ресурс исследований, то новая центральная точка уже будет находиться на модифицированных уровнях факторов и весь процесс модификации повторяется до тех пор, пока не кончится у исследователя ресурс опытов.

Во многих практических задачах влияние некоторых величин на отклик объекта исследования нельзя оценить качественно. Однако, исследователя может интересовать вопрос, насколько существенно влияние того или иного фактора (или их комбинации) на обобщенный отклик $W(z)$ модели СС. Кроме того, исследователь часто заинтересован в том, чтобы ослабить влияние вторичных, наблюдаемых и управляемых факторов. Сделать это можно путем специального планирования и привлечения для обработки в случае качественных факторов дисперсионного анализа. План типа «классический латинский квадрат» позволяет исследовать влияние трех факторов, варьируемых на 4 уровнях. Размерность латинского квадрата определяется числом уровней варьирования факторов. Например, пусть $n=4$, тогда размерность латинского квадрата равна 4×4 . В табл. 2 приведен план

типа латинского квадрата. Уровни двух факторов, представлены в виде строк (фактор 1) и столбцов (фактор 2), а фактор 3, уровня которого обозначены буквами латинского алфавита (A, B, C, D), заносится в ячейки квадрата 4×4 . В первую строку (первый уровень фактора 1) в упорядоченной последовательности заносится буквенные обозначения уровней фактора 3. Порядок расположения уровней фактора 3 в последующих строках определяется путем циклического сдвига элементов предыдущей строки вправо, при этом (буквы) расположены таким образом, что в каждой строке и в каждом столбце любая буква встречается только один раз. Отметим, что этот эксперимент состоит из 16 опытов, тогда как соответствующий ПФЭ 4^3 потребовал бы 64 опытов.

Таблица 2 - Пример плана типа «Латинский квадрат»

Фактор 1	Фактор 2			
	1	2	3	4
1	A	B	C	D
2	D	A	B	C
3	C	D	A	B
4	B	C	D	A

1.6 Планирование имитационного эксперимента.

При ограниченном ресурсе времени моделирования необходимо так спланировать эксперименты, чтобы выборка статистик, необходимых для определения откликов, была достаточной для вычисления зависимости откликов $W(X)$ от вектора параметров моделирования X . Поскольку разбросы выборочных значений случайны, то обусловленная ими неточность результатов ИЭ определяется размером выборки значений отклика. На практике размер выборки обычно является функцией ресурсов, отпущенных на ИЭ. При этом предполагается независимость и нормальность распределений откликов Y_i как представляющий собой сумму большого числа небольших эффектов и переменная отклика является результатом действия всех этих эффектов. Поэтому необходимо спланировать количество опытов, которые осуществляются обычно внутри модели по правилу «автоматической остановки», базирующееся на методе доверительных интервалов. Этот метод предполагает задание точности представления d_n математического ожидания $(M Y_n)$ и б., дисперсии (D_n) у n -ой компоненты вектора отклика Y_n и уровня значимости λ , гарантирующим попадание $M Y_n$ и D_n внути интервалов $(\bar{Y}_n \pm d_n)$, $(S_n^2 \pm b_n)$ с вероятностью $(1-\lambda)$. Здесь \bar{Y}_n и S_n^2 представляют собой среднее значение и выборочную дисперсию, вычисленных по выборке объема N и являющимися оценками $M Y_n$ и D_n .

В ходе испытания и исследования свойств ИМ исследователь определяет векторы точностей $\{d_1, \dots, d_n\}$ или $\{b_1, \dots, b_n\}$ представления компонент вектора отклика Y . Выполнение правила «автоматической остановки» представляет собой итеративную процедуру, суть которой состоит в следующем. До начала проведения серии ИЭ известны уровни значимости λ и векторы точностей представления компонент вектора отклика b , или d . Из априорных сведений или опыта устанавливается количество начальных экспериментов N_1 , необходимых для получения выборок значений компонент вектора откликов модели $\{y_{nk}\}$, $n = 1, \dots, k = 1, N_1$. Если исследователь предполагает, что в ходе исследования свойств ИМ число опытов будет не слишком большим, то в качестве начального значения он полагает $N_1=5$. Далее алгоритм выбора числа экспериментов состоит из следующих шагов.

Шаг 1. По выборке $\{y_{nk}\}$ находит среднее значение \bar{Y}_n и выборочную дисперсию S_n^2 .

Шаг 2. Для очередного номера n компоненты вектора откликов модели определяют достаточную точность оценки \bar{Y}_n и S_n^2 при выполнении N_1 экспериментов. Здесь возможны следующие случаи.

Если выборка малого объема ($N_1 \leq 30$), то для вычисления доверительного интервала используют t -статистику, имеющую распределение Стьюдента, и определяют:

$$d_{1n} = t_{kp} \sqrt{S_n^2 / (N_1 - 1)}, \quad (1.25)$$

где t_{kp} – критическое значение, t – статистика при $(N_1 - 1)$ степенях свободы и заданном уровне значимости λ .

Если размер выборки большой ($N_1 > 30$) и можно предположить нормальность Y_n , то для вычисления доверительного интервала MY_n используют двухстороннюю статистику нормированного распределения

$$d_{1n} = Z_{\lambda/2} \sqrt{S_n^2 / N_1}, \quad (1.26)$$

где $Z_{\lambda/2}$ – значение нормированного нормального распределения на уровне значимости $\lambda/2$.

Если нормальность Y_n предложить нельзя, но N_1 большое, то применяют неравенство Чебышева:

$$P(|\bar{Y}_n - MY_n| \geq h \cdot S_n \sqrt{N_1}) \leq 1/h^2, \quad (1.27)$$

где h – некоторая заранее заданная константа, означающая число среднеквадратичных отклонений, удовлетворяющих исследователю.

В этом случае с достаточной для практике точностью доверительный интервал отклика можно вычислить по формуле:

$$d_{1n} = \sqrt{S_n^2 / N_1 (1 - \lambda)}. \quad (1.28)$$

При оценивании дисперсии откликов формула отыскания оценки D_n с достоверностью $(1 - \lambda)$ имеет вид:

$$P((1 - b_{1n})S_n^2 \leq D_n \leq (1 + b_{1n})S_n^2) \geq 1 - \lambda. \quad (1.29)$$

Поскольку N_1 достаточно велико, то эту статистику аппроксимируют нормальным распределением и получают форму оценки дисперсии:

$$b_{1n} = Z_{\lambda/2} \sqrt{2(N_1 - 1)}, \quad (1.30)$$

где $Z_{\lambda/2}$ – значение нормированного нормального распределения при заданном уровне значимости $\lambda/2$.

Шаг 3. Сравнивают достигнутую точность d_{1n} или b_{1n} оценок MY_n или D_n в N_1 опытах с заданными значениями d_n или b_n . Если выполняется неравенство $d_{1n} \leq d_n$ или $b_{1n} \leq b_n$, то это означает, что по n -й компоненте вектора откликов Y достигнута требуемая точность оценки в N_1 экспериментах и переходят к шагу 5.

Шаг 4. Выполняют еще один модельный эксперимент, увеличив N_1 на единицу ($N_1=N_1+1$) и переходят к шагу 1.

Шаг 5. Проверяют, все ли компоненты вектора откликов проверены на удовлетворение точности оценки MY_n и D_n . Если точность по всем (d_1, \dots, d_n) достигнута, это означает завершение ИЭ. В противном случае меняют компоненты вектора откликов модели и переходят к шагу 2.

Достигимость заданной точности d_n или b_n оценки некоторой статистики (или группы статистик) может быть одним из условий окончания ИЭ. Необходимо при этом считать, что для достижения требуемой точности математического ожидания $\{MY_n\}$ требуется N_{12} равное максимальному значению для оценки всех компонентов.

Для достижения требуемой точности $\{D_n\}$ требуется N_{12} , равная максимальному значению для оценки всех компонентов. Кроме того, поскольку имитация в модели СС осуществляется с помощью процедуры Монте-Карло, то требуется повторение опыта N_2 раз с одинаковыми и теми же значениями параметров модели (но с разными начальными величинами базового генератора $\{s_{nk}\}$; k -номер места, использующего базовый генератор случайных чисел).

Итак, объем выборки $\{Y_{nl}\}$ в одной реализации для достижения требуемой точности вектора (1.23) равен $\max(N_1, N_2)$. Потом этот эксперимент в следующей реализации ИМ должен быть повторен N_2 раз согласно процедуре Монте-Карло.

В любом ИЭ важно определить интервалы изменения компонент вектора параметров модели X . Поэтому до начала имитации необходимо найти предельные значения статистик моделирования. Существуют три фактора, влияющие на выбор интервала изменения параметров модели: необходимость получения одинаковой относительной точности статистик на разных участках области изменения параметров ИМ; характер функции отклика; назначение ИЭ, определяющее способ планирования эксперимента. Если анализ погрешностей показывает, что на каком-то участке моделируемого процесса статистик вызывает сомнения, то в ходе следующей имитации на этом участке необходимо спланировать более частную функцию статистики. Обычно стремится к тому, чтобы точность функции отклика модели всех ее участков была одинакова. Если известно, что в ИЭ имеются особенности, которые можно обнаружить при получении данных в регулярной последовательности, используют классический последовательный план. В этом случае все параметры ИМ полагают постоянными в центральной области плана, а один из них меняют во всей области его значений. Выбор интервалов между соседними значениями параметра осуществляется с учетом баланса точности. Аналогичным образом планируют изменения второго параметра, третьего параметра и т.д. Затем составляют план изменения параметров при переходе от одного имитационного опыта к другому по методике, которая ранее была рассмотрена в вопросах 1.3–1.5.

Ранее были рассмотрены случаи, когда в качестве рабочей точки серии ИЭ выбирались уровни факторов, $\{x_i^0\}$, имевшие место во время мониторинга реальной СС.

Рассмотрим способы задания $\{x_i^0\}$ на основе априорной информации о моделируемой СС. Наиболее распространенный способ задания основного уровня имитации на основе анализа функции отклика $Y = \phi(X, G)$. Если у исследователя имеются сведения о координатах только одной точки на функции отклика и нет информации о границах изменения факторов, то ему ничего не остается, как рассматривать эту точку в качестве основного уровня плана ИЭ. Когда границы изменения компонент вектора факторов известны исследователю и он знает, что наилучшее значение X находятся внутри области изменения фактора, то в качестве основного уровня он может выбрать любую из точек факторного пространства. Хуже обстоит дело, когда известная точка лежит на границе факторного пространства. В этом случае исследователю придется выбирать уровень со слэвигом внутрь области. Когда исследователю известно, что имеется много наилучших

значений функции отклика (мультимодальная зависимость), то в качестве основного уровня он может выбирать любую случайную точку внутри факторного пространства. Наконец, когда исследователю по постановке ИЭ известна под область в факторном пространстве, где исследуемый процесс протекает оптимально и эксперименты ведутся только для уточнения места этого оптимума, тогда в качестве основного уровня он выбирает центр этой подобласти.

Назначение ИЭ существенно влияет на выбор интервалов изменения параметров модели. Если предметом имитации является поиск узких мест в функционировании СС или выбор гипотез о механизмах явлений, то применяется многоступенчатое факторное планирование. В этом случае на выбор уровня факторов влияет относительная точность фиксации статистик имитации ST_{ij} на разных участках изменения параметров ИМ. Когда предметом моделирования является поиск оптимальных условий функционирования СС, либо выбор существенных для управления СС параметров, либо нахождение вида функции: аппроксимирующей поведение откликов СС, то задача сводится к выбору для каждого фактора его уровня изменения. Обычно на интервал варьирования факторов накладываются ограничения сверху и снизу. Интервал варьирования не может быть меньше той погрешности, с которой исследователь фиксирует уровни, чтобы они были различими. Но интервал варьирования не может быть настолько большим, чтобы эти уровни оказались за пределами изменениями факторов. Для выбора интервала варьирования исследователь обычно использует следующую априорную информацию: сведения о точности фиксации факторов; данные о кривизне функции откликов (линейная или нелинейная); результаты оценки чувствительности откликов $\{Y_i\}$ к изменениям параметров модели $\{x_i\}$. Поэтому часто проводятся серии «затравочных» ИЭ для знакомства с исследуемой СС и для получения недостающей априорной информации о влиянии выбранных интервалов варьирования факторов на вид искомой функции откликов.

При составлении плана анализа результатов ИЭ необходимо предусмотреть несколько проверок точности и приемлемости результатов имитации. В ряде случаев для определения источников погрешностей можно использовать уравнение баланса точности. Рассмотрим суть технологии составления этого уравнения на следующем примере. Пусть в ходе ИЭ на модели измеряются переменные А, В, С, Д, которые связаны между собой уравнением сохранения значений между парами переменных $A \cdot B = C \cdot D$. Установлено, что из-за неточности описания реальной СС одна из этих переменных (неизвестно, какая из них) является источником систематической погрешности. Для коррекции алгоритма ИМ нужно выявить не только виновника погрешности, но и место ее возникновения в алгоритме модели. Пусть в ходе имитации не меняется значение переменной А, и остальные переменные изменяются следующим образом: $C \rightarrow m \cdot C$, $D \rightarrow n \cdot D$,

$B \rightarrow mB$. Уравнение баланса при этих условиях имеет вид: $A + mB = mC + nD$. Далее используется следующее правило. Если одна из переменных в этом уравнении баланса, например, может быть представлена в виде суммы $(A + \varphi(A))$, где $\varphi(A)$ - систематическая погрешность, то эту переменную можно обнаружить, рассматривая поочередно случаи с фиксированным значением каждой переменной. Переменная, относительная погрешность которой при фиксированном ее значении не меняется, и является причиной систематической погрешности. Исключением является случай, когда переменная имеет пропорциональное приращение ошибки $A+kA$ (k - постоянная величина). В этом случае погрешность предлагаемым способом обнаружить невозможна.

Часто, для поиска погрешности ИЭ используется предварительный эксперимент, по результатам которого исследователь строит график зависимости $Y=f(X)$. Система координат и сама функция f выбираются такими, чтобы график был линейным или хотя бы не имел большой кривизны вблизи начала координат для последующей экстраполяции графика. Обычно используются линейные или полупаррафнические шкалы, позволяющие строить прямую зависимость отклика Y от вектора параметров X . Например, пусть исследователю известно, что один из графиков линейной зависимости отклика имеет систематическую погрешность. Необходимо определить, какой из графиков верен, и оценить при этом эту погрешность. Пусть для простоты каждый набор результатов ИЭ характеризуется одинаковым показателем степени, хотя сами графики не совпадают. Однако известно, что в области малых значений зависимость $Y=f(X)$ должна проходить через начало координат. Тогда для идентификации и оценки систематической погрешности ИЭ строят оба графика в области малых значений. Тот график, который не удовлетворяет условию прохождения через начало координат и имеет систематическую погрешность.

Отметим, что среднее значение отклика \bar{Y}_n при $X=0$ является оценкой этой погрешности.

Лекция_2 ЭТАПЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ СЛОЖНЫХ СИСТЕМ

- 2.1 Использование имитационных моделей при проектировании СС.
- 2.2 Этапы разработки программ имитации функций СС.
- 2.3 Этапы испытания имитационных моделей СС.
- 2.4 Этапы эксплуатации программ ИМ СС.
- 2.5 Составление содержательного описания объекта имитации.

2.1 Использование имитационных моделей при проектировании СС.

В производстве понятие «технология» включает в себя совокупность производственных процессов и операций, с помощью которых создаются определенные виды изделий, а также описание способов производства. Поскольку программы для ЭВМ также являются изделиями, создаваемыми коллективами специалистов разной квалификации при значительных трудовых затрат, то актуальной является задача создания современной технологии производства таких программ. Имитационные модели являются специфичными и достаточно сложными программными изделием, и их разработка должна вестись с применением прогрессивной технологии. Технология проектирования ИМ включает в себя методы и средства, обеспечивающие их создание и развитие в течение всего периода их жизни. Этот период включает этапы проектирования, изготовления и эксплуатации ИМ. Он начинается с формирования назначения и принципов построения ИМ и завершается после полного прекращения эксплуатации модели. Как правило, ИМ является средством проектирования сложных систем. Технология их изготовления и использования связана с технологией проектирования и эксплуатации объекта моделирования. Поэтому вначале рассмотрим вопросы применения ИМ на различных этапах проектирования сложных систем, затем уделим внимание собственно технологии создания и эксплуатации ИМ.

Проектирование сложной системы представляет собой весьма трудоемкий процесс, в котором участвуют разные специалисты. Большие размеры системы, сложность поведения ее компонент, высокая стоимость разработки требуют методов математического моделирования на всех этапах проектирования такой системы. Поэтому моделирование сопровождает и процесс проектирования, и процесс испытания, и процесс эксплуатации сложной системы. ГОСТами на разработку программных изделий, например ЕСПД, предусматриваются следующие этапы проектирования: формулировка технического задания, разработка технических предложений по созданию изделия, разработка эскизного проекта, разработка технического проекта, рабочее проектирование, испытание изделия. На всех этапах про-

ектирования специалистам приходится рассматривать две стороны объекта проектирования: требования к системе со стороны внешней среды (внешнее проектирование) и организацию функционирования сложной системы (внутреннее проектирование).

Формулировка технического задания. Начало работы над проектом сложной системы имеет целью сформулировать задачи проектирования и организовать рабочие группы по решению поставленных задач проектирования. Вначале образуется небольшая инициативная группа специалистов, которая на основе всесторонних обсуждений вырабатывает документ, который содержит в себе следующую информацию:

- формулировку проблем (результаты обсуждений);
- предлагаемые варианты решений (с указанием достоинств и недостатков каждого из них);
- предложения по необходимому составу специалистов, комплектованию и срокам подключения рабочих групп;
- грубую оценку времени и денежных средств, необходимых для разработки проекта.

В результате создается группа специалистов-проектировщиков, имеющая четко поставленную задачу проектирования сложной системы.

Разработка технических предложений. На этом этапе предполагается: выбор наилучшего решения, укомплектование группы проектировщиков системы до полного состава, составление плана всех работ над проектом. Для выбора наилучшего решения часть сил и средств направляется на поиск альтернативных решений. Для этого используются аналитические и имитационные модели. Затем наступает момент выбора решения. Многократно пересматриваются формулировки задачи проектирования сложной системы. Основными средствами исследования являются математические модели проектируемого объекта. При этом выполняются следующие виды моделирования компонент: проектирование единичной нити, проектирование большой нагрузки, состязательное проектирование.

В процессе проектирования единичной нити прорабатываются вопросы выполнения функций системы. При проектировании большой нагрузки просматриваются действия системы, которые зачастую случайно распределены во времени. Цель состязательного проектирования – найти на модели ответные действия системы на тот случай, когда внешняя среда пытается затруднить работу алгоритма системы. Эти три вида работ должны осуществляться одновременно в течение проектирования. Этот этап характеризуется появлением большого количества новых идей решения проблемы, новых проработок по частным вопросам отдельных направлений работы.

Новые идеи должны документироваться в виде рабочих записок или отчетов. С ними необходимознакомить всех разработчиков. К этому времени должна быть укомплектована группа проектировщиков системы.

Обязательно постоянство основного состава группы в процессе проектирования сложной системы. В результате их работы формулируется проблема, изыскиваются пути ее реализации, составляется план выполнения проекта системы, который должен:

- содержать график распределения времени по этапам работ с указанным необходимым средств и состава исполнителей;
- предусматривать консультации крупных специалистов по сложным вопросам;
- содержать заявки на проведение натурного и модельного экспериментов.

Эскизное проектирование. Этап начинается с разработки первого варианта того, что называется системой. Для этих целей требуется большое число экспериментов, в результате которых получают данные, позволяющие создать лучшую математическую модель сложной системы. На этапе эскизного проектирования решаются вопросы, допускающие многозначные ответы. Отчетная документация этапа должна содержать:

- подробное описание работы всей системы в целом;
- четкое описание подсистемы (форма, количество и времена ее создания, алгоритм ее функционирования);
- перечень допустимых значений характеристик системы;
- хотя бы один метод физической реализации предложенного способа функционирования сложной системы;
- информацию об исследованиях других методов физической реализации предложенного способа.

Техническое проектирование. Разработка технического проекта выполняется, когда система уже «заморожена» и идет отладочная работа. Уточняются и детализируются алгоритмы функционирования компонент системы. На имитационных моделях проверяется, соответствуют ли компоненты системе своему назначению. На этом этапе наряду с множеством подмоделей компонент системы обязательны создание и испытание обобщенной имитационной модели самой системы, с помощью которой решаются все вопросы внешнего и внутреннего проектирования. Отчетность аналогична отчетности предыдущего этапа, но содержит более детализированные описания системы и ее компонент.

Рабочее проектирование. Этот этап характеризуется созданием опытного образца, предназначенного для испытания свойств системы. Опытный образец системы должен иметь в своем составе контрольную аппаратуру. Зачастую при конструировании контрольной аппаратуры используются ЭВМ, позволяющие заменить ряд имитаторов-устройств соответствующими программами-имитаторами и таким образом автоматизировать процесс исследования свойств системы с помощью образца. На этом этапе должна быть разработана техническая документация, включающая в себя: технические характеристики по эксплуатации опытного образца системы; пол-

ное описание всех технических решений, использованных при создании опытного образца.

Испытание изделия. На этом этапе осуществляются испытание и оценка принятых проектных решений по созданию сложной системы. Основная цель испытания состоит в получении подтверждения, что система работает так, как предусматривалось при проектировании. В ходе испытания система отлаживается, исключаются неизбежные дефекты ее функционирования. Испытание проводится по плану испытаний, предусматривающему усложнение условий работы опытного образца в процессе испытания по специально разработанным методикам. В результате испытаний составляются отчеты. В них даются рекомендации по улучшению использования системы в производственных условиях и приводятся результаты анализа различий между реальным образцом системы и ее макетом.

2.2 Этапы разработки программы имитации функций СС.

Независимо от способа проектирования сложной системы и назначения моделирования можно выделить следующие восемь этапов создания и использования математических моделей:

- определение объекта имитации, установление границ и ограничений моделирования, выбор показателей для сравнения эффективности вариантов системы (составление *содержательного описания* объекта моделирования);
- формулировка замысла модели, переход от реальной системы к логической схеме ее функционирования (составление *концептуальной модели*);
- реализация описания объекта в терминах математических понятий и алгоритмизации функционирования ее компонент (составление *формального описания* объекта);
- преобразование формального описания объекта в описание имитационной модели (составление *описания имитационной модели*);
- программирование и отладка модели (*программирование модели*);
- проверка модели, оценка ее свойств и затрат ресурсов на имитацию (*испытание и исследование модели*);
- организация модельного эксперимента на ЭВМ (*эксплуатация модели*);
- интерпретация результатов моделирования и их использование в ходе проектирования сложной системы (*анализ результатов*).

Рассмотрим порядок действий разработчиков на каждом из перечисленных этапов создания и использования имитационных моделей на ЭВМ.

Составление содержательного описания объекта моделирования представляет собой выполнение следующих действий.

Вначале определяются объект имитации и состав исходной технической информации, достаточной для изучения тех сторон его функционирования, которые представляют интерес для исследователя.

Устанавливаются границы изучения функционирования объекта. Составляется возможный список ограничений модели, которые допустимы при организации имитации или при наличии которых еще имеет смысл имитации функционирования объекта моделирования.

Перед разработчиками ИМ ставится вполне конкретные цели моделирования и формулируются основные критерии эффективности, по которым предполагается проводить сравнение на модели различных проектных решений или вариантов организации сложной системы.

Результатом работ на данном этапе является содержательное описание объекта моделирования с указанием целей имитации и аспектов функционирования объекта моделирования, которые необходимо изучить на ИМ. Обычно они представляют собой техническое описание объекта моделирования, описание внешней среды, с которой он взаимодействует, и временнную диаграмму этого взаимодействия.

Составление концептуальной модели. Производится в следующей последовательности. На основе анализа поставленной задачи определяется общий замысел модели. Выдвигаются гипотезы и фиксируются все допущения, необходимые для построения ИМ.

На основании содержательного описания уточняется задача моделирования, определяются процедура и график ее решения. Уточняется метаполитика всего имитационного эксперимента в зависимости от наличия ресурсов, выделенных для имитации. Общая задача моделирования разбивается на частные подзадачи. Устанавливаются приоритеты решения подзадач моделирования. Обосновываются требования в рабочей силе и выделение ресурсов ЭВМ. Затем проводится тщательный анализ задач, стоящих перед имитацией.

Выполняются такие работы, как выбор параметров и переменных системы, представляющих интерес для моделирования; уточнение критерев эффективности функционирования различных вариантов проектируемой системы; выбор типов аппроксимации отдельных компонент модели. Проводятся также: предварительный анализ требований к модели сложной системы; определение необходимых математических уравнений, описывающих реальные процессы; поиск возможных методов проверки правильности функционирования модели.

Определяются методы проверки программной реализации модели системы, формулируются технические требования на программирование модели. Изучаются возможности применения известных методов обработки и анализа результатов, выбираются способы представления ожидаемых результатов моделирования. Одновременно с этим исследователь должен выбрать язык будущей formalизации процессов в объекте моделирования.

Результатом выполнения работ являются: концептуальная модель, выбранный язык formalизации и способ организации имитации, реализуемый языком formalизации. В состав концептуальной модели входит: уточнение, необходимые для построения ИМ.

ненное содержательное описание объекта моделирования, свободное от всего того, что не представляет интереса для имитации поведения системы; список параметров и переменных моделирования; критерии эффективности функционирования вариантов системы; список используемых методов обработки результатов имитации и перечисление способов представления результатов моделирования.

Отметим, что при создании небольших моделей данный этап работ совмещается с этапом составления содержательного описания моделируемой системы. Только с усложнением объекта моделирования и задач имитации появляется необходимость в данном этапе, главная цель которого состоит в определении того способа формализации, который наиболее подходит для решения конкретной задачи проектирования.

Составление формального описания объекта моделирования представляет собой ответственный этап создания имитационной модели сложной системы. Цель - получение исследователем формального представления алгоритмов поведения компонент сложной системы и отражение вопросов взаимодействия между собой этих компонент. При составлении формального описания модели исследователь использует тот или иной язык формализации. В зависимости от сложности объекта моделирования и внешней среды могут использоваться три вида формализации: аппроксимация явлений функциональными зависимостями, алгоритмическое описание процессов в системе, смешанное представление в виде последовательности формула и алгоритмических записей.

В зависимости от выбранного способа организации квазипараллелизма используются свой язык формализации и своя методика составления формального описания объекта имитации. После составления формального описания объекта моделирования приступают к его проверке. Это первая главная проверка достоверности будущей модели сложной системы в процессе проектирования. Для обеспечения контроля правильности функционирования модели вводят классические модели, достоверность которых доказана. Они фигурируют в модели в виде составных частей. На вход таких моделей поступают данные, вычисляемые в других частях модели, достоверность которых проверяется. Если результат работы классической модели окажется недостоверным, то считают, что предпредставляющая часть формального описания системы также недостоверна.

В процессе проверки достоверности модели необходимо ответить на следующие вопросы: позволяет ли модель решить поставленные задачи моделирования, насколько полна предложенная схема модели и отражает ли она фактическую последовательность развития процессов в реальной системе. Необходимо провести анализ каждой функции модели и убедиться, что она нашла свое отражение в формальном описании системы. В том случае, когда уравнения получены на основании анализа опытных данных, необходимо провести выборочную проверку согласия уравнений с исход-

ной информацией, по которой они получены. Для уравнений, полученных теоретическим путем, необходимо провести вычисления в нескольких контрольных точках с целью определения приемлемости результатов. Для дополнительной проверки уравнений желательно провести анализ размерностей и масштабов переменных системы.

Важно правильно выбрать вычислительные средства, которые обеспечивают быстроту логики программирования, минимальные затраты на моделирование, доступность выбранной ЭВМ, быстрое получение результатов. На практике начинающие исследователи либо из-за спешки, либо из-за организационных трудностей доступа к соответствующей ЭВМ зачастую недооценивают важность правильного выбора базовой ЭВМ, на которой им придется проводить эксперименты. Результатом являются огромные потери материальных ресурсов и труда исследователя из-за низкой технологической оснастки средств моделирования на выбранной ЭВМ. Конкретные рекомендации по выбору ЭВМ и математического обеспечения моделирования предоставить чрезвычайно трудно. В основу выбора ЭВМ необходимо положить наличие средств оперативного построения и испытания модели. Результатом этапа является проверенное формальное описание исследуемой системы на выбранном языке формализации.

Составление имитационной модели. Как только средства реализации и тип ЭВМ выбраны, исследователь приступает к этапу преобразования формального описания в описание имитационной модели. Многие специалисты по имитации не делают различия между этими этапами. И это оправдано, когда имитационная модель можно представить с помощью таких универсальных средств описания, как агрегативные схемы или системы массового обслуживания. При переходе к более сложным системам это различие проявляется. Прежде всего исследователя не удовлетворяет состав стандартной статистики моделирования, реализуемой соответствующими системами моделирования. Кроме того, появляется множество дополнительных трудностей, связанных с выбранным способом организации квазипараллелизма. Как правило, в этих случаях исследователю приходится решать множество дополнительных вопросов, не связанных с описанием поведения моделируемой системы. Сюда входят следующие вопросы реализации модели: декомпозиция объекта на составляющие и формирование элементов модели; отработка вопросов синхронизации частей компонент модели друг с другом модельного времени; организация сбора статистики; задание начальных условий моделирования; планирование процесса имитации отдельных вариантов системы; проверка окончания моделирования; обработка результатов имитации. Все эти действия являются чрезвычайно трудоемкими и ответственными, их успешное выполнение зависит прежде всего от опыта и интуиции исследователя. Результатом этапа является описание имитационной модели сложной системы.

Программирование. На этом этапе выполняются следующие действия. Составляется план создания и использования программной модели. Как правило, программа модели создается с помощью средств автоматизации моделирования на ЭВМ. Поэтому в плане указываются: тип ЭВМ; средства автоматизации моделирования; примерные затраты памяти ЭВМ на создание программы модели и ее рабочих массивов; затраты машинного времени на один цикл работы модели; оценки затрат на программирование и отладку программы модели.

Затем исследователь приступает к программированию модели. В качестве технического задания на программирование служит описание имитационной модели. Специфика работ по программированию модели зависит от средств автоматизации моделирования, которые доступны исследователю. Не существует значительных отличий создания программы модели от обычной автономной отладки программных модулей большой программы или пакета программ. В соответствии с текстом производится деление модели на блоки и подблоки. В отличие от обычной автономной отладки программных модулей, при автономной отладке блоков и подблоков программы модели объем работ существенно увеличивается, поскольку для каждого модуля необходимо создать и отладить еще имитатор внешнего окружения. Весьма существенно выверить реализацию функций модуля в модельном времени и оценить затраты машинного времени на один цикл работы модели как функцию от значений параметров модели. Завершаются работы при автономной отладке компонент модели подготовкой форм представления входных и выходных данных моделирования.

2.3 Этапы испытания имитационных моделей СС.

Вначале переходит ко второй проверке достоверности программы модели системы. В процессе этой проверки устанавливается соответствие операций в программе и описание модели. Для этого производится обратный перевод программы в схему модели (ручная «прокрутка» позволяет найти грубые ошибки статики модели).

После исключения грубых ошибок ряд блоков объединяется и начинается комплексная отладка модели с использованием тестов. Отладка по тестам начинается с нескольких блоков, затем в этот процесс вовлекается все большее число блоков модели. Отметим, что комплексная отладка программы модели намного сложнее отладки пакетов прикладных программ, поскольку ошибки динамики моделирования в этом случае найти значительно труднее вследствие квазипараллельной работы различных компонент модели. По завершении комплексной отладки программы модели необходимо вновь оценить затраты машинного времени на один цикл расчетов на модели. При этом полезно получить аппроксимацию времени моделирования на один цикл имитации.

Следующим действием является составление технической документации на модель сложной системы. Результатом этапа к моменту окончания комплексной отладки программы модели должны быть следующие документы:

- описание имитационной модели;
- описание программы модели с указанием системы программирования и принятых обозначений;
- полная схема программы модели;
- полная запись программы модели на языке моделирования;
- доказательство достоверности программы модели (результаты комплексной отладки программы модели);
- описание входных и выходных величин с необходимыми пояснениями (размерностей, масштабов, диапазонов изменения величин, обозначений);
- оценка затрат машинного времени на один цикл моделирования;
- инструкция по работе с программой модели.

Для проверки адекватности модели объекту исследования после составления формального описания системы исследователь составляет план проведения натурных экспериментов с прототипом системы. Если прототип системы отсутствует, то можно использовать систему вложенных ИМ, отличающихся друг от друга степенью детализации имитации одних и тех же явлений. Тогда более детальная модель служит в качестве прототипа для обобщенной ИМ. Если же построить такую последовательность невозможно либо из-за отсутствия ресурсов на выполнение этой работы, либо из-за недостаточности информации, то обходятся без проверки адекватности ИМ. Согласно этому плану параллельно с отладкой ИМ осуществляется серия натурных экспериментов на реальной системе, в ходе которых накапливаются контрольные результаты. Имея в своем распоряжении контрольные результаты и результаты испытаний ИМ, исследователь проверяет адекватность модели объекту.

При обнаружении ошибок на этапе отладки, устранимых только на предыдущих этапах, может иметь место возврат на предыдущий этап. Кроме технической документации к результатам этапа прилагается машинная реализация модели (программа, оттранслированная в машинном коде ЭВМ, на которой будет происходить имитация).

Испытание модели. Это важный этап создания модели. При этом необходимо выполнить следующее. Во-первых, убедиться в правильности динамики развития алгоритма моделирования объекта исследования в ходе имитации его функционирования (проводя *верификацию модели*). Во-вторых, определить степень адекватности модели и объекта исследования. Под *адекватностью* программной имитационной модели реальному объекту понимают совпадение с заданной точностью векторов характеристик поведения объекта и модели. При отсутствии адекватности проводят

либровку имитационной модели («подправляют характеристики алгоритмов компонент модели»).

Наличие ошибок во взаимодействии компонент модели возвращает исследователя к этапу создания имитационной модели. Возможно, что в ходе формализации исследователь слишком упростили физические явления, исключив из рассмотрения ряд важных сторон функционирования системы, что привело к недекватности модели объекту. В этом случае исследователь должен вернуться к этапу формализации системы. В тех случаях, когда выбор способа формализации оказался неудачным, исследователю необходимо повторить этап составления концептуальной модели с учетом новой информации и появившегося опыта. Наконец, когда у исследователя оказалось недостаточно информации об объекте, он должен вернуться к этапу составления содержательного описания системы и уточнить его с учетом результатов испытания предыдущей модели системы.

Исследование свойств имитационной модели. При этом оцениваются точность имитации явлений, устойчивость результатов моделирования, чувствительность критерия качества к изменению параметров модели. Получить эти оценки в ряде случаев бывает весьма сложно. Однако без успешных результатов этой работы доверия к модели не будет ни у разработчика, ни у заказчика ИМ. У разных исследователей в зависимости от вида ИМ сложились различные интерпретации понятий точности, устойчивости, стационарности, чувствительности ИМ. Пока не существует общепринятой теории имитации явлений на ЭВМ. Каждому исследователю приходится полагаться на свой опыт организации имитации и на свое понимание особенностей объекта моделирования. Ниже автор излагается личный опыт выполнения такой работы и не претендует на безусловность определений.

Точность имитации явлений представляет собой оценку влияния стохастических элементов на функционирование модели сложной системы.

Устойчивость результатов моделирования характеризуется сходимостью контролируемого параметра моделирования к определенной величине при увеличении времени моделирования варианта сложной системы (когда проверяется маловероятные ситуации в системе).

Стационарность режима моделирования характеризует собой некоторое установленвшееся равновесие процессов в модели системы, когда дальнейшая имитация бессмысльна, поскольку новой информации из модели исследователь не получит и продолжение имитации практически приводит только к увеличению затрат машинного времени. Такую возможность необходимо предусмотреть и разработать способ определения момента достижения стационарного режима моделирования.

Чувствительность ИМ представляется величиной минимального приращения выбранного критерия качества, вычисляемого по статистикам мо-

делирования, при последовательном варьировании параметров моделирования на всем диапазоне их изменений.

2.4 Этапы эксплуатации программ ИМ СС.

Эти этапы начинаются с составления плана эксперимента, позволяющего исследователю получить максимум информации при минимальных усилиях на вычисление. Обязательно статистическое обоснование плана эксперимента. Планирование эксперимента представляет собой процедуру выбора числа и условий проведения опытов, необходимых и достаточных для решения поставленной задачи с требуемой точностью. При этом существенно следующее: стремление к минимизации общего числа опытов, обеспечение возможности одновременного варьирования всеми переменными; использование математического аппарата, формализующего многие действия экспериментаторов; выбор четкой стратегии, позволяющей принимать обоснованные решения после каждой серии экспериментов на модели.

Затем исследователь приступает к этапу *проведения рабочих расчетов* на модели. Это весьма трудоемкий процесс, требующий больших затрат ресурсов ЭВМ и обилия канцелярской работы. Отметим, что уже на ранних этапах создания ИМ необходимо тщательно придумывать состав и объемы информации моделирования, чтобы существенно облегчить дальнейший анализ результатов имитации. Итогом работы являются результаты моделирования.

Анализ результатов моделирования. Данный этап завершает технологическую цепочку этапов создания и использования имитационных моделей. Получив результаты моделирования, исследователь приступает к интерпретации результатов. Здесь возможны следующие циклы имитации. В первом цикле имитационного эксперимента в ИМ заранее предусмотрен выбор вариантов исследуемой системы путем задания начальных условий имитации для машинной программы модели. Во втором цикле имитационного эксперимента модель монтируется на языке моделирования, и поэтому зачастую требуется повторная трансляция и редактирование программы.

Возможно, что в ходе интерпретации результатов исследователь установил наличие ошибок либо при создании модели, либо при формализации объекта моделирования. В этих случаях осуществляется возврат на этапы построения описания имитационной модели или на этап составления концептуальной модели системы соответственно.

Трудно дать общие рекомендации по реализации этапов эксплуатации программ ИМ СС поскольку каждая ИМ создается под конкретный объект исследования или проектирования. Однако в дальнейшем мы постараемся на типовых примерах описать технологию эксплуатации программ ИМ СС. В качестве примеров объекта имитации выступают: вычислительный про-

цесс в локальных вычислительных сетях ЭВМ (ВП ЛВС) и многомашинных вычислительных сетях ЭВМ (ВИЛ ВС); вероятностные технологические процессы производства (ВТИП); технологические процессы сборочно-разборочного производства ремонта изделий (ТП СРП). С нашей точки зрения, примером реализации технологии исследования и проектного моделирования семи типов объектов имитации достаточно для обучения технологии имитации сложных систем.

Результатом этапа интерпретации результатов моделирования являются рекомендации по проектированию системы или ее модификации. Имея в своем распоряжении рекомендации, исследователи приступают к принятию проектных решений. На интерпретацию результатов моделирования оказывают существенное влияние изобразительные возможности используемой ЭВМ и реализованной на ней система моделирования. Рекомендации по интерпретации результатов моделирования на имитационных моделях в общем случае давать довольно трудно. Исследователю можно по рекомендовать использовать графики как наиболее удобное изобразительное средство. В ряде случаев широко применяются дисплеи и средства для создания кинофильмов о моделируемом явлении. С их помощью можно помочь исследователю наблюдать и изучать явление в замедленном или в убыстренном темпе по сравнению с реальной скоростью протекания исследуемых процессов.

Например, при *проектном моделировании ЛВС* в ходе серии ИЭ исследователь может: получить оценки характеристик качества обслуживания запросов пользователей и загрузки компонентов оборудования; осуществить поиск «узких мест» в заданной структуре ВП ЛВС; оценить эффективность стратегий «спекулятивных» векторов откликов моделирования и критерии принятия решений из выбранного состава вариантов комбинаций параметров ЛВС и рабочей нагрузки (РН) на ЛВС; установить степень снижения функциональных характеристик компонентов ВП и РН из-за наличия отказов и восстановленный работоспособности основного оборудования ЛВС по сравнению с абсолютно надежной их работой; определить параметры регрессионных зависимостей откликов отказа появления работоспособности компонент оборудования узлов ЛВС.

При проектном моделировании вероятностных технологических процессов производства на ИМ ВТИП исследователь может: определить статические оценки критического времени реализации множества взаимосвязанных микротехнологических операций $\{\text{МТХО}_j\}$, образующих вероятностный технологический процесс (ВТИП), получить граф критических путей, образуемых при каждой реализации ИМ ВТИП согласно процедуре Монте-Карло, с помощью которых оценить наиболее вероятный состав $\{\text{МТХО}_j\}$, входящих в критический путь реализации вероятностного сетевого графика (ВСГР); определить наиболее опасные участки развития ВТИП; оценить общую пропускную способность вариантов организации

ВТИП при имеющемся составе ресурсов и оборудования для их реализации.

При проектном моделировании или при эксплуатации технологических процессов сборочно-разборочного производства ремонта изделий сложной структуры (ТП СРП) с помощью имитационной модели исследователь может: оценить пропускную способность ТП СРП ремонта изделий сложной структуры; найти «узкие места» в организации ТП СРП и исследоватьдинамику изменения «узких мест» при изменении параметров ТП СРП; оценить эффективность стратегий ремонта изделий на предприятии; определить допустимые диапазоны изменения откликов ИМ ТП СРП и общего показателя эффективности при росте интенсивности поступления на предприятие изделий, требующих ремонта согласно заданной технологии.

В конечном итоге после выполнения всех перечисленных выше итерационных этапов исследователь либо окажется удовлетворенным результатами моделирования и будет их учитывать при проектировании сложной системы, либо забракует проектируемую систему и сформирует техническое задание на разработку новой архитектуры системы.

2.5 Составление содержательного описания объекта имитации.

Создание имитационной модели сложной системы начинается с постановки задачи. Но зачастую заказчик ставит задачу недостаточно четко. Поэтому работа обычно начинается с поискового изучения системы. Оно порождает новую информацию, касающуюся ограниченной задач и возможных альтернативных вариантов. Отправной точкой при построении модели сложной системы будем считать исходную документацию на существующую систему или техническое задание на проектируемую систему. Составность сведений об объекте моделирования представляется в ней в виде схем, текстов, таблиц, экспериментальных данных, характеризующих предполагаемую структуру и функционирование системы. В исходной документации одновременно могут быть избыток информации об одних аспектах поведения системы и недостаток информации о других аспектах функционирования системы. Имеется также информация о внешних воздействиях и окружающей среды. Как правило, информации, представленной в исходной документации, бывает недостаточно для описания поведения сложной системы и изучения различных сторон ее поведения. Зачастую этого и не требуется. Задача может оказаться значительно уже. Поэтому необходимо сначала четко определить цель будущего исследования на модели, а затем в соответствии с этой целью переработать весь объем исходной информации и восполнить недостающую информацию. Такой процесс назовем составлением *содержательного описания* сложной системы.

Рекомендуется следующая последовательность действий при составлении содержательного описания сложной системы: выбор показателей каче-

ства, отражающих цели моделирования; определение управляющих переменных, выбор состава контролируемых характеристик объекта моделирования; детализация описания режимов функционирования системы; пополнение описания информацией о внешней среде. На этом этапе исследователь широко применяет опрос инженерно-технического персонала или специалистов, имеющих опыт работы с прототипом проектируемой или реально существующей системой, поведение которой необходимо исследовать.

Выбор показателей качества моделируемой системы. Этот достаточно важный шаг этапа определяется теми задачами, для решения которых и строится модель. Часто наблюдается плохая тенденция имитировать все, что касается поведения объекта исследования. Следует строить модель, ориентированную на решение лишь тех вопросов, на которые требуется найти ответы, а не имитировать реальную систему во всех подробностях. Можно сослаться на закон Парето, который гласит, что «в каждой группе или совокупности существует жизненно важное меньшинство и привильное большинство. Ничего действительно важного не происходит, пока не затронуто жизненно важное меньшинство» [2.1.3]. На первом этапе важно отделить главное от второстепенного. Например, если исследователи интересуются вопросами взаимодействия компонент в системе, то ему не следует изучать проблемы энергоснабжения и экономические аспекты функционирования системы. Выбор цели моделирования определяет характеристики, которые отражают поведение сложной системы. В дальнейшем вся работа сводится к выявлению и детализации тех аспектов функционирования системы, которые имеют отношение к выбранным показателям.

Определение управляющих переменных системы. Изучается техническая документация, по которой прослеживается информация, относящаяся к управлению системой, и анализируется на предмет общности и различия. В соответствии с описанием компонент реальной системы прототипа или согласно техническому заданию на проектируемую систему устанавливается состав управляемых и контролируемых характеристик объекта моделирования. Прежде всего, выделяются те характеристики управления системой и контроля за ее работой, которые имеют отношение к цели моделирования. Все составляющие функциональной зависимости, определяющие значение показателя качества системы, включаются в состав управляющих переменных и контролируемых характеристик моделирования.

Детализация описания режимов функционирования системы. Рассматривается и дополняется имеющаяся информация для возможного выделения алгоритмов функционирования в каждом из режимов работы системы. Любые неясности устраняются. Составляются временные диаграммы функционирования системы. Определяются наиболее неясные или

сложные моменты функционирования компонент системы, устанавливается последовательность действий этих компонент, выделяются вероятные места возникновения конфликтных ситуаций и приводится принятый порядок их разрешения в системе.

Составление описания внешней среды. На этом завершающем шаге этапа информация о поведении внешней среды берется из технического задания. В случае моделирования отдельных аспектов функционирования существующей системы проводится исследовательская работа, цель которой состоит в определении алгоритмов взаимодействия системы с внешней средой. Иногда возможны модификации или пополнение состава управляющих переменных системы из-за детализации алгоритмов взаимодействия между системой и внешней средой.

Таким образом, на каждом шаге данного этапа перерабатывается и дополняется имеющаяся информация о поведении системы в соответствии с поставленными целями моделирования. Результатом является содержательное описание сложной системы, выполненное на языке, принятом при составлении технических отчетов и понятном заказчику. Все лишнее, не относящееся к задаче моделирования, отбрасывается. Цель моделирования поставлена, и можно приступить к математической постановке задачи моделирования. Общего рецепта составления содержательного описания не существует. Успех зависит от интуиции и профессиональной подготовки разработчиков имитационной модели.

Таким образом, во всех этих случаях объекты имитации имеют вероятностную природу и для них (как правило) не выполняются предпосылки для использования аналитических моделей, поэтому приходится использовать имитацию.

Лекция 3 МЕТОД СТАТИСТИЧЕСКИХ ИСПЫТАНИЙ КАК ОСНОВА МОДЕЛИРОВАНИЯ ВЕРОЯТНОСТНЫХ СИСТЕМ

- 3.1 Особенности использования вероятностных моделей сложных систем.
- 3.2 Виды и алгоритмы реализации единичных жребьев на ЭВМ.
 - 3.3 Моделирование на ЭВМ псевдослучайных величин.
 - 3.4 Методы исключения и суперпозиции при моделировании случайных величин.
 - 3.5 Моделирование случайных величин, имеющих мультимодальную функцию плотности распределения.

3.1 Особенности использования вероятностных моделей сложных систем.

Когда параметры элементов СС имеют вероятностную природу и когда многие ограничения формализации СС не выполняются, оказывается, что аналитические модели (АНМ) трудно применять для исследования динамики функционирования СС. В этих случаях результаты исследования на АНМ зачастую противоречат друг другу. Объясняется это особенностями СС, имеющими место именно из-за вероятностного характера поведения компонентов в реальной СС. Выделим три основные особенности моделирования вероятностных СС: во-первых, о поведении СС в лучшем случае известен только алгоритм, которым развертывание расчетов зависит от жребия и результата которых заранее непредсказуем; во-вторых, имеет место зависимость результата моделирования от исходных данных; в-третьих, необходимо накапливать статистику результатов моделирования и затем уже проводить статистический анализ накопленной статистики: в-четвертых, возникает необходимость в использовании механизма получения в модели значений этих случайных чисел. Необходимо помнить, что любое моделирование случайного процесса в СС позволяет дать только ответ на вопросы, что же произойдет при конкретных значениях этих случайных величин. И чтобы узнать больше, необходимы повторные эксперименты, в ходе которых накапливается статистика поведения элементов системы. Эти особенности моделирования вероятностных СС приводят к тому, что исследователю проще использовать универсальную процедуру метода статистических испытаний, которую мы назовем "процедурой Монте-Карло" и суть которой сводится к следующему.

Вместо того, чтобы описывать СС с помощью АНМ проводится N "результативных" случайного явления в модели объекта имитации (заранее заданным числом N раз). При этом вектор параметров модели X не меняется (фиксирован) и на модели определяются значения компонентов вектора откликов Y . Так же, как в реальности, конкретная j -ая реализация случайного процесса (из-за случайного характера алгоритма поведения реального

44

явления) в модели всякий раз будет складываться по-разному. Таким образом, проведя N экспериментов с моделью СС при одинаковых значениях вектора параметров модели X из-за случайного характера алгоритма СС получает выборку значений откликов $\{Y_i\}_{i=1}^N$. Усредняя выборку значений $\{Y_i\}$, находя вектор математических ожиданий значений компонент этого вектора $\bar{Y} = (\bar{Y}_1, \bar{Y}_2, \dots, \bar{Y}_n)$ и дисперсию S_y^2 компонент вектора отклика при фиксированных значениях вектора параметров $\{Y_i\}$.

Число экспериментов N определяется из необходимой точности оценки \bar{Y} . Задавшись точностью вычислений ϵ , по таблицам нормального распределения для доверительной $\beta=1-\alpha$ находит требуемое число экспериментов N , обеспечивающее оценку среднего \bar{Y} с вероятностью ошибки α . При оценивании среднего значения компонентов вектора \bar{Y} наиболее часто встречаются следующие случаи:

a) $\{Y_i\}$ имеют нормальное распределение, но объем выборки мал ($N < 30$). Тогда составляется t -статистика:

$$t = \sqrt{N-1}(\bar{Y} - \mu)S_y^{-2}, \quad (3.1)$$

где S_y^2 и μ — оценка дисперсии и истинное математическое ожидание компонент вектора откликов Y ; \bar{Y} — среднее значение компонент вектора откликов. Как известно, t -статистика имеет распределение Стьюдента [1]. Поэтому по соответствующим таблицам при $N-1$ степенях свободы и заданном уровне значимости α оценивают доверительные интервалы для математических ожиданий вектора откликов:

$$\bar{Y} \pm d = \bar{Y} \pm t_{\alpha/2} S_y / \sqrt{N-1}, \quad (3.2)$$

откуда требуемый объем выборки для оценки отклика с точностью d будет равен:

$$N > \left(\frac{t_{\alpha/2} S_y}{d} \right)^2 + 1; \quad (3.3)$$

б) $\{Y_i\}$ имеют нормальное распределение; истинное математическое ожидание μ известно, объем выборки велик ($N \geq 30$). В этом случае используется двусторонняя нормальная статистика Z при заданном уровне значимости α . Доверительный интервал можно определить в виде соотношения

$$\bar{Y} \pm d = \bar{Y} \pm Z_{\alpha/2} \frac{S_y}{\sqrt{N}}, \quad (3.4)$$

и требуемый объем выборки

45

$$N \geq \left(\frac{S_{\alpha} - Z_{\alpha/2}}{d} \right)^2, \quad (3.5)$$

где $Z_{\alpha/2}$ – значение нормированного нормального распределения при уровне значимости α ; d – допустимая величина ошибки для оценки средних значений откликов;

в) нормальность распределений компонентов вектора Y предположить нельзя, но известны истинные значения математического ожидания μ и дисперсии σ^2 при большом объеме выборки ($N \geq 30$). В этом случае лучше всего использовать неравенство Чебышева

$$P\left\{ \left| \bar{Y} - \mu \right| \geq K \frac{\sigma}{N} \right\} \leq \frac{1}{K^2}. \quad (3.6)$$

Из этого неравенства доверительный интервал для оценки среднего значения можно записать в виде

$$\bar{Y} + d \in \bar{Y} \pm \frac{K\sigma}{\sqrt{N}} = \bar{Y} \pm \frac{\sigma}{\sqrt{N}(1-\alpha)}. \quad (3.7)$$

а требуемый объем выборки отклика

$$N \geq \frac{\sigma}{d^2(1-\alpha)} \quad (3.8)$$

Обычно задаются $d = \sigma/4, \alpha = 0.05$. Тогда $N > 320$.

При оценивании дисперсии σ^2 задача отыскания ее оценки S^2_Y с достоверностью $1-\alpha$ записывается в виде

$$P\left\{ (1-d)\sigma^2 \leq S^2_Y \leq (1+d)\sigma^2 \right\} = 1-\alpha \quad (3.9)$$

Эту задачу можно решать только при больших объемах выборки N . В этих случаях удобнее всего использовать χ^2 -статистику:

$$\chi^2 = \frac{(N-1)S^2}{\sigma^2} \text{ с } (N-1) \text{ степенями свободы.} \quad (3.10)$$

Из-за большого N эту статистику можно аппроксимировать нормальным распределением. Поэтому формула для расчета требуемого объема выборки отклика имеет вид:

$$N \geq 1 + 2 \left(\frac{Z_{\alpha/2}}{d} \right)^2, \quad (3.11)$$

где $Z_{\alpha/2}$ – значение нормально распределенной величины при уровне значимости α ; d – доверительный интервал оценки, чаще всего $d = \frac{\sigma_{\bar{Y}}}{4}, \sigma_{\bar{Y}}^2$, эмпирическая оценка дисперсии отклика.

Основной операцией, из совокупности которых складывается процесс моделирования СС, согласно процедуре Монте-Карло является некая I -ая реализация случайного процесса $F(X, \omega)$. Она представляет собой как бы один экземпляр реализации алгоритмов развития СС в дереве вариантов его функционирования. Как правило, она скапливается из последовательности, состоящей из двух типов составляющих. Первый тип составляющих представляет собой вычислительные процедуры F , а вторая составляющая означает разыгрывание значений ω , с помощью специального разработанного алгоритма, носящего название "бросание жребия". Как раз, когда в ход моделируемого явления вмешивается случайность, ее влияние в модели учитывается не расчетом, а "жребием". Далее расчетами F уже учитывается влияние исхода "жребия" на дальнейшее течение процесса. Учитываются при этом и условия разыгрывания следующего "жребия". Идею,ложенную в основе процедуры Монте-Карло, рассмотрим на следующем примере. Пусть необходимо вычислить некоторую скользящую величину C , заданную некоторым математическим выражением, сложным для аналитического сто исследования. На первом шаге определяют случайную величину ξ такую, что выполняются условия: $M\xi^2 < C$; $D\xi^2 \leq \infty$. На втором шаге моделируют N независимых реализаций этой случайной величины $(\xi_1, \xi_2, \dots, \xi_N)$. На третьем шаге по случайной выборке $(\xi_j) (j=1, N)$ строят выборочную оценку величины $C = \frac{1}{N} \sum_{j=1}^N \xi_j$.

Классическим примером применения процедуры Монте-Карло является задача вычисления площади области сложной формы A . Процедура Монте-Карло является задача вычисления площади области сложной формы A . Процедура Монте-Карло при решении данной задачи решается следующим образом.

1. Область A погружается в область G с известной площадью $g = mes(G) < \infty$.
2. Формируется случайная величина Бернулли ξ , математическое ожидание которой равно C :

$$\xi = gF_A(\eta) = \begin{cases} 1 & \text{если } \eta \in A \\ 0 & \text{если } \eta \notin A \end{cases} \quad (3.12)$$

где $F_A(\eta)$ – индикаторная функция области A ;
 η – случайный двумерный вектор равномерно распределенный в области G .
 Обозначим $\rho = P(\eta \in A)$ – вероятность события $\{\eta \in A\}$. Тогда

$$\begin{aligned} M(\xi) &= g \cdot p + \theta(l-p) = gp = C \\ D(\xi) &= g^2 p(l-p) - C(g \cdot C) < \infty \end{aligned} \quad (3.13)$$

Моделируется N независимых реализаций $(\xi_1, \xi_2, \dots, \xi_N)$. Строится случайная выборка $\xi_1, \xi_2, \dots, \xi_N$ с помощью функции распределения (3.12) и вычисляются оценка величины C :

$$\hat{C} = \frac{1}{N} \sum_{l=1}^N \xi_l = g + \frac{1}{n} \sum_{l=1}^N F_1(\eta_l) = g + \frac{h'}{N}, \quad (3.14)$$

где h' - число реализаций ξ_l , $l=1, N$, попавших в область A ; h'/N - относительная частота попаданий внутрь области, т.е. эффективная оценка вероятности p .

Для вычисления одной и той же величины C возможны разные варианты построения случайной величины ξ удовлетворяющие условию $M(\xi) = C$: $D(\xi) < \infty$. При этом предпочтение отдается варианту, наиболее полно удовлетворяющему следующим условиям:

- дисперсия $D(\xi)$ минимальна на множестве $\{\xi\}$;
- моделирование ξ осуществляется одним из известных способов одним из известных способов реализации крибя первого типа.

Рассмотрим еще один пример выполнения вероятностного технологического процесса производства (ВТПП), блок-схема реализации которого приведена на рис. 2. ВТПП представляет собой совместное выполнение множества алгоритмов $\{\text{АЛ}_j\}$, $j \in \overline{1, 11}$. Каждый из этих алгоритмов реализуется в течение постоянного времени t_j . При выполнении алгоритмов в трех местах ВТПП имеет место вероятностный выбор выполнения последующих алгоритмов технологического процесса. На рис. 2 выбор выполнения АЛ_j по одному из направлений осуществляется вероятностным образом и представлен операторами проверки PR_k ($k \in \overline{1, 3}$). Сумма вероятностей в каждой из этих операторов равна 1. Отметим, что в основной функции алгоритма АЛ_{11} является синхронизация выполнения ветвей ВТПП. Допустим имел место выбор в этих операторах проверки направления выполнения, помеченный "1". Это приведет к тому, что алгоритмы $\text{АЛ}_1, \text{АЛ}_5$ и АЛ_8 не будут выполняться. Кроме того, время выполнения ВТПП равно:

$$T_{A1} = \max \{(t_1 + t_4); (t_2 + t_4 + t_{10}); (t_3 + t_4 + t_{10}); (t_5 + t_6 + t_{10} + t_{11})\} \quad (3.15)$$

Если же выбор направления выполнения ВТПП совпал с пометкой "0" у операторов PR_k , то имеет место ситуация, когда алгоритмы $\text{АЛ}_6, \text{АЛ}_9$ и АЛ_{10} не будут выполняться. Время выполнения ВТПП при этом будет иметь другое значение:

$$T_{A2} = \max \{(t_1 + t_4 + t_8 + t_{10}); (t_2 + t_8 + t_{10}); (t_3 + t_8 + t_{10}); (t_5 + t_6 + t_{10} + t_{11})\} \quad (3.16)$$

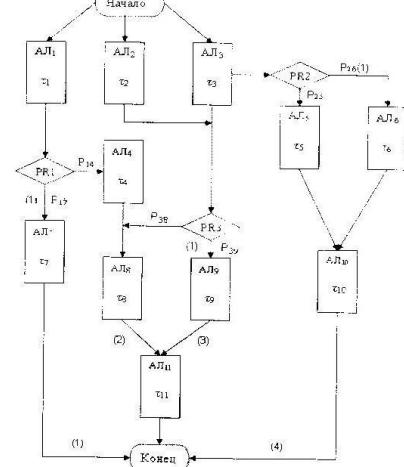


Рис. 2 - Блок-схема выполнения ВТПП

Таким образом, в реальном ВТПП могут быть не только рассмотренные случаи выполнения множества алгоритмов, но и различные времена выполнения ВТПП. В общем случае, если не делать никаких предположений, то результаты выполнения множества алгоритмов и время их реализации будут вероятностными. Задача эта усложняется, если считать, что времена выполнения алгоритмов t_j не являются детерминированными величинами. Например, если t_j определяются по функциям распределения $F_j(t)$ всякий раз могут быть различными. Этот пример показывает насколько возрастает сложность решения вероятностных задач и появляется неопределенность

результатом моделирования самого ВТПП и времени его выполнения. Очевидно, что эту задачу нужно решить с помощью процедуры Монте-Карло.

Для этого построим имитационную модель ВТПП. В подмоделях алгоритмов ИМ АЛ_j будем осуществлять в *i*-й реализации модели разыгрыш по жребию 3-го типа значений t_{ij} по соответствующим функциям распределения.

$$t_{ij} = F_j(t) \quad (3.17)$$

Сами операции проверки направления реализации ветвей ВТПП в операторах PR_i реализуются путем разыгрыша по жребию 2-го типа. При этом фиксируется статистика выполнения не только каждого алгоритма АЛ_j, но и всего ВТПП. В итоге каждой *i*-й реализации в ИМ фиксируются множество статистик выполнения АЛ_j (t_{ij}); $j = 1..11$ и общее время выполнения ВТПП (T_{ai}). После *N* реализаций ИМ ВТПП получаем выборки значений времени выполнения алгоритмов $\{t_{ij}\}$; $j = 1..11$, $i = 1..N$ и общего времени выполнения ВТПП $\{T_{ai}\}$; $i = 1..N$.

Далее согласно процедуре Монте-Карло по этим выборкам определяются оценки математического ожидания и дисперсии времени выполнения алгоритмов и всего времени реализации алгоритма:

$$\bar{t}_{ji} = S^2 t_{ji}, j = 1..11; \bar{T}_{ai} = S^2 T_{ai}. \quad (3.18)$$

Зная оценки математического ожидания откликов модели и их выборочные дисперсии по известным формулам вычисляется ошибка расчета откликов имитационной модели ВТПП.

Приведенных примеров достаточно для того, чтобы установить тот известный факт, что вероятностные СС в общем случае можно решить только с помощью процедуры Монте-Карло. При этом процедура Монте-Карло лежит в основе моделирования всех вероятностных ситуаций и требуется в ИМ разыгрыш жребиев различного типа для моделирования в модели тех вероятностных ситуаций, которые имели место в реальной СС.

3.2 Виды и алгоритмы реализации единичных жребиев на ЭВМ.

Условимся называть "единичным жребием" любой опыт со случайным исходом, в котором разыгрывается:

- 1) произошло или не произошло событие A , имеющее вероятность появления P_A ;
- 2) какое из возможных событий A_1, A_2, \dots, A_m произошло, если заданы вероятности их появления P_1, P_2, \dots, P_m ;
- 3) какое значение x принялла случайная величина X ;

4) какую совокупность значений (x_1, x_2, \dots, x_N) приняла система случайных чисел (X_1, X_2, \dots, X_N) .

Существует стандартный механизм, с помощью которого можно осуществить любой из перечисленных видов жребия. Он состоит в формировании случайного числа ξ , значения которого равновероятны на интервале [0,1], и последующем формульному преобразовании этого случайного числа ξ . Рассмотрим алгоритмы моделирования всех четырех видов жребия с помощью генератора случайных чисел ξ , равномерно распределенных на интервале [0,1].

Жребий 1^o. Моделирование события A. Моделируется очередное значение случайного числа ξ путем обращения к генератору случайных чисел ГСЧ. Если $\rho > \xi$, то $A :=$ "истина" (событие произошло), иначе $A :=$ "ложь" (событие не произошло).

Жребий 2^o. При заданном векторе вероятностей появления (P_1, P_2, \dots, P_m) моделирование одного из возможных событий (A_1, A_2, \dots, A_m). Делают интервал [0,1] на *m* участков, длины которых соответствуют вероятностям появления событий P_i , $i = 1..m$. Разыгрывается очередное значение величины ξ путем обращения к ГСЧ. Если ξ попадает в *i*-й участок, то считают, что появилось *i*-е событие (A_i = "истина", а остальные A_j = "ложь", $j \neq i$). Считается также, что все события A_i независимы.

Жребий 3^o. Какое значение x принялла случайная величина X.

Если X -дискретная величина и известно распределение вероятностей и ее значения:

$$\{(x_1, P_1), (x_2, P_2), \dots, (x_k, P_k)\};$$

где $\sum_{i=1}^k P_i = 1$, то формируют эмпирическую функцию дискретного распределения $\{(x_1, P_1), (x_2, P_2), \dots, (x_k, \sum_{i=1}^k P_i)\}$. Разыгрывается очередное значение ξ и проверяется выполнение неравенства $\sum_{i=1}^l P_i < \xi \leq \sum_{i=1}^{l+1} P_i$. Если это неравенство выполняется для *l*-го дискретного интервала, то считают, что $x = x_l$.

Если же X -непрерывная величина и известна функция плотности вероятности ее распределения $f(x)$, формируют функции распределения вероятностей появления величин x : $F(x) = \int_{-\infty}^x f(x) dx$. Разыгрывается очередное значение величины ξ путем обращения к ГСЧ. Далее вычисляется обратная функция Ψ от $F(x)$ при заданном значении $x^* = \xi$; тогда $x = \Psi(\xi)$ будет искомым значением случайной величины X . Предполагается при этом, что кривая $F(x)$ аналитически задана, поэтому и возможна процедура получения обратной функции Ψ в точке $x^* = \xi$. Если же $F(x)$ и $\Psi(x)$ трудно пред-

ставимы аналитически, то их заменяют (апроксимируют) линейно-кусочной (полигоном частот) или ступенчатой линией (гистограммой). А затем уже поступают аналогично предыдущему случаю для нахождения значений обратной функции в точке $x = \xi$.

Жребий #. *Какую совокупность значений (x_1, x_2, \dots, x_d) принялла система случайных чисел (X_1, X_2, \dots, X_d) . Возможны два случая: X_i независимы друг от друга; X_i представляют систему зависимых величин. В первом случае разыгрывая (x_1, x_2, \dots, x_d) сводится к k раз повторенному моделированию величин X по жребию Ξ^k . Гораздо хуже обстоит дело, когда x_1, x_2, \dots, x_d зависимы. В этом случае, разыграв одну из переменных, например $x_1 = \tilde{x}_1$, но уже согласно условному распределению двух величин и с учетом того, что $x_1 = \tilde{x}_1$. Третью величину, например x_2 разыгрывают по условным распределениям уже трех переменных: а именно: $x_3 = \tilde{x}_3$ при условии, что $x_2 = \tilde{x}_2$ и т.д.*

3.3 Моделирование на ЭВМ псевдослучайных величин.

В основе реализации четырех видов единичных жребьев лежит последовательное генерирование равномерно распределенных величин ξ_i , $i=0, 1, 2, \dots$. Во всех ГСЧ используется процедура формирования равномерно распределенных величин ξ_i на интервале $[0, 1]$. Затем уже производится функциональное преобразование ξ_i в x с заданным законом распределения $F(x)$.

Исторически возникло два подхода к моделированию величин ξ : с помощью специальных физических генераторов-приставок к ЭВМ и с помощью специально подобранного алгоритма, реализованного на ЭВМ в виде специальной процедуры. Первый подход позволяет получить действительно случайные величины ξ на любом интервале их значений. При этом никогда предыдущее значение ξ не повторится вновь. Моделирование ξ основано на известных физических явлениях (гамма-излучение, косметические лучи, белый шум в электрических цепях и т.д.) с заданными законами распределений. Основными достоинствами таких генераторов являются: действительно случайный характер их появления, неповторяемость и возможность получать одно число за одну специальную команду обращения к генератору-приставке к ЭВМ. При этом они обладают и существенными недостатками: нестабильностью работы, зависимостью от температуры, влажности, погоды, давления. Кроме того, неповторяемость ξ затрудняет организацию многовариантных исследований процессов, требуя для этого специальных запоминаний последовательностей $\{\xi_j\}$. Эти недостатки определили появление ГСЧ второго типа – программные ГСЧ. Это специальные процедуры, реализованные на ЭВМ, позволяющие

моделировать последовательность $\{\xi_j\}$. При этом ξ_j случайны на некотором интервале значений j , называемом периодом аperiодичности ($N_{\text{A}}=N$). Далее, начиная с некоторого N_{A} , все ξ_j начинают повторяться. Поэтому числа ξ_j и назвали псевдослучайными. Достоинства ГСЧ второго типа состоят в том, что они стабильно работают и не боятся перепадов температур, влажности и давления: достигается полная воспроизведимость всей последовательности чисел начиная с некоторого начального значения ξ_0 . Главным недостатком является большая их ресурсоемкость. На получение одного числа ξ_j необходимо выполнение целой программы (процедуры) генератора. Однако, несмотря на указанные недостатки, эти процедуры практически вытеснили из употребления физические датчики случайных чисел.

Базовый программный ГСЧ формирует последовательность псевдослучайных величин, равномерно распределенных на интервале $[0, 1]$: $\xi_1, \xi_2, \dots, \xi_n$. Практически в силу дискретности представления чисел в ЭВМ можно генерировать $M=2^k$ различных чисел (здесь k – разрядность кода ЭВМ, а M – максимальное число, представленное в ЭВМ). Вероятность появления очередного соседнего числа составляет $P(\xi_i = x_i) = \frac{1}{M}$. Обычно моделирование в ГСЧ второго типа ведется по итеративной формуле, т.е. очередное значение x_i^* является функцией всех предыдущих значений:

$$x_i^* = \phi(x_{i-1}^*, \dots, x_0^*) \bmod M \quad (3.19)$$

Нормировка по модулю M необходима для перевода значений x_i^* в интервал $[0, 1]$. Реализация x_i^* получаются внутри алгоритма путем подобранного преобразования ϕ из ранее имитированных чисел $x_0^*, x_1^*, \dots, x_{i-1}^*$. Существует множество процедур, реализующих в ГСЧ различные алгоритмы преобразований ϕ . Наиболее распространен контрунтный метод генерации псевдослучайных чисел, сводящийся к формальному преобразованию:

$$x_i^* = (\lambda x_{i-1}^* + C) \bmod M. \quad (3.20)$$

Работу этого алгоритма представим для случая $A=2$ и $C=0$, т.е. $x_i^* = 2x_{i-1}^* \bmod M$. Графически моделирование по этому алгоритму представлено на рис. 3 и 4. В первом случае начальное значение x_0 подобрано очень неудачно, что приводит к очень короткому периоду аperiодичности ГСЧ ($N_{\text{A}}=2$). Этот ГСЧ позволил получить всего 2 различных значения, и

далее началось повторение этой пары чисел. Здесь диагональ играет роль подстановки предыдущего значения x в качестве x_{i-1} для расчета x_i . Во втором случае x_0 подобрано гораздо лучше, что позволило существенно увеличить период аperiодичности. И, как видно из рис. 4, все шесть x , действительно отличаются друг от друга и распределены относительно равномерно по всему интервалу значений (от 0 до 1). Именно переходной режим в ГСЧ и позволяет получать псевдослучайные последовательности чисел x_i .

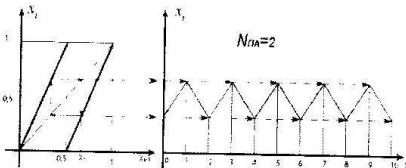


Рис. 3 - Диаграмма моделирования псевдослучайных последовательностей по конгруэнтному методу, когда $x_i^* = 2x_{i-1} \text{ mod } M$ и $N_{da}=2$.

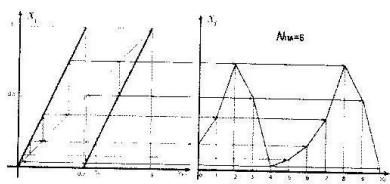


Рис. 4 - Диаграмма моделирования псевдослучайных последовательностей по конгруэнтному методу, когда $x_i^* = 2x_{i-1} \text{ mod } M$ и $N_{da}=6$.

Кроме рассмотренного, существует еще один способ, называемый "способом серебряных квадратов". Суть этого способа заключается в следующем. Берут произвольное m -разрядное число x_0 , возводят его в квадрат, и результат уже имеет $2m$ разрядов. Из него выбирают средние m цифр, получая таким способом новое число x_1 . Далее процедура повторяется, где в качестве начального значения используют ранее полученное x_1 . Его возводят в квадрат. Из него выбирают средние m цифр и получают новое число x_2 .

На рис. 3 приведена диаграмма моделирования псевдослучайных последовательностей по конгруэнтному методу, когда $x_i^* = 2x_{i-1} \text{ mod } M$, начальное значение x_0 . На рис. 4 приведена диаграмма моделирования псевдослучайных последовательностей по конгруэнтному методу $[x_i^* = 2x_{i-1}]$, с другим начальным значением x_0 .

Третий способ представляет собой совмещение двух конгруэнтных способов. Он предполагает собой совмещение двух конгруэнтных способов. Имеет место двойное обращение к ГСЧ, работающему по конгруэнтному способу, а затем происходит формульное преобразование

$$x_{i+1}^* = (x_i - x_{i-1}) \text{ mod } M. \quad (3.21)$$

Эта простейшая процедура увеличивает существенно период первоначальности ГСЧ. Наиболее эффективна процедура возмущения основного алгоритма ГСЧ. Суть этой процедуры состоит в следующем. Одновременно используются две процедуры ГСЧ. Основная процедура моделирует последовательность случайных чисел по алгоритму:

$$x_d^* = \phi_1(x_{d-1}) \text{ mod } M \quad (3.22)$$

Для этой процедуры через каждые K чисел меняется начальное значение x_0 моделируемое по алгоритму второй процедуры:

$$x_{d2}^* = \phi_2(x_{d2-1}) \text{ mod } M \quad (3.23)$$

Комбинация обращений к двум процедурам ГСЧ удается увеличить период аperiодичности на несколько порядков.

Более подробно с алгоритмами формирования базовых генераторов псевдослучайных величин $\{x_i\}$, равномерно распределенных на интервале $[0,1]$.

Моделирование псевдослучайных величин, имеющих распределение вероятностей, отличающееся от равномерного, осуществляется путем

функционального преобразования величин ξ_i . Рассмотрим наиболее употребительные алгоритмы такого функционального преобразования.

Нормальное распределение можно моделировать множеством способов. Рассмотрим два алгоритма. **Первый алгоритм** моделирования нормально распределенной величины η с параметрами $M=0$ и $\sigma^2=1$ основан на центральной предельной теореме теории вероятностей. Согласно этой теореме при сложении достаточно большого числа независимых случайных величин, сравнимых между собой по своим дисперсиям, получаем новую случайную величину n , распределенную по нормированному нормальному закону. Чем больше складывается значений ξ_i , тем ближе распределение $\sum \xi_i$ оказывается к нормальному. Поскольку ξ_i имеют одну и ту же дисперсию и считаются независимыми друг от друга, то для практики достаточно от 6 до 12 сложений ξ_i :

$$\eta = \sqrt{\frac{12}{n}} \sum_{i=1}^n \xi_i - \frac{n}{2}, \quad (3.24)$$

где n -число обращений к ГСЧ.

Второй алгоритм является более экономичным, поскольку позволяет по двум обращениям к базовому ГСЧ (ξ_1, ξ_2) получить пару псевдослучайных чисел (η_1, η_2), распределенных по нормальному закону. Расчеты ведутся по формулам

$$\eta_1 = \sqrt{-2I_{n\xi_1}} \cos 2\pi \xi_2; \quad \eta_2 = \sqrt{-2I_{n\xi_1}} \sin 2\pi \xi_2. \quad (3.25)$$

Экспоненциальное распределение широко используется при моделировании СТС. Наиболее распространены следующие два способа имитации псевдослучайных величин, которые имеют одну и ту же функцию плотности распределения $f(x) = \lambda e^{-\lambda x}$, где λ -параметр экспоненциального распределения.

Первый способ основан на методе обратной функции:

$$F(x) = 1 - e^{-\lambda x}. \quad (3.26)$$

Обратной к ней будет функция $e^{-\lambda x}$. Тогда экспоненциально распределенная величина с параметром λ формируется путем следующего функционального преобразования величины ξ :

$$x = -\frac{1}{\lambda} \ln(1 - \xi). \quad (3.27)$$

Второй способ основан на методе композиции. Опытным путем задаются величины t и θ такими, чтобы характеристики величины x были как

можно ближе к экспоненте. С помощью ГСЧ моделируется θ штук значений $\xi_1, \xi_2, \dots, \xi_\theta$. Затем осуществляется пересчет по формуле

$$t = \text{так}\{\xi_1, \xi_2, \dots, \xi_\theta\}. \quad (3.28)$$

Для практики достаточно задавать $t=4$ и $\theta=4$, что позволяет получать экспоненциально распределенную величину x с параметром $\lambda=1$.

Дискретные распределения также моделируются из базовых значений ξ_i . При схеме независимых испытаний часто необходимо моделировать псевдослучайные числа, распределенные по биноминальному закону. Так, например, выборка из биноминального распределения получается прямым подсчетом числа единиц в n -разрядном случайном числе ξ .

Если необходимо моделировать дискретные случайные величины, имеющие геометрическое распределение

$$P(y = y) = P(1 - p)^{y-1}, \text{ где } y = 1, 2, \dots \quad (3.29)$$

то находят номер испытания (y), которое принесет первый успех для появления события A с вероятностью P . Поэтому последовательно обращаются к ГСЧ и моделируют очередное ξ_i . Проверяют выполнимость жребия первого типа. Номер обращения (y) к ГСЧ, обеспечивающего A_i ="истина", и дает очередное значение величины y , распределенной по геометрическому распределению с параметром P .

В технике широко распространено распределение Пуассона

$$P(k) = \frac{\lambda^k}{k!} e^{-\lambda}, \quad (3.30)$$

где λ – параметр распределения Пуассона для дискретной переменной k .

Для моделирования величин k последовательно обращаются к ГСИ, получают каждый раз очередное значение ξ_i и проверяют выполнимость неравенства

$$\prod_{i=1}^k \xi_i \geq t^{-\lambda} > \prod_{i=1}^{k+1} \xi_i. \quad (3.31)$$

Как только это неравенство выполняется, то сумма ξ_i будет иметь распределение Пуассона с параметром λ .

3.4 Методы исключения и суперпозиции при моделировании случайных величин

В тех случаях, когда плотность распределения $f_0(x)$ моделируемой случайной величины x имеет сложный аналитический вид и метод обратной функции неприменим, используется метод исключения. Обозначим функ-

цию распределения $F_0 = f(x,y); \theta \leq f_0(x), x \in R$ - область, ограниченная кривой $y=f_0(x)$, и общее (см. рис. 5.а). Определяется мажорирующая функция $y=g(x)$; где $g(y) \geq f_0(x) \geq 0$ и область $G=f(x,y); \theta \leq g(x); x \in R > F_0$ (см. рис. 5.а). При этом мажорирующая функция $g(x)$ должна иметь более простой вид, чем $f_0(x)$.

Как видим, G охватывает (мажорирует) область F_0 и имеет более равномерно распределенный в области G . Моделирующий алгоритм включает следующие этапы:

- 1) подбор мажорирующей функции $g(x)$;
- 2) моделирование случайного вектора (ξ^*, η^*) , равномерно распределено в области G ;
- 3) если (x,y) - реализация вектора (ξ^*, η^*) и $y \leq f_0(x)$ (т.е. точка с координатами (x,y) попадает в область F_0), то x принимается в качестве реализации случайной ξ . Если же $y > f_0(x)$, что означает то, что точка (x,y) лежит за пределами F_0 , то реализация (x,y) исключается и этап 2 повторяется снова.

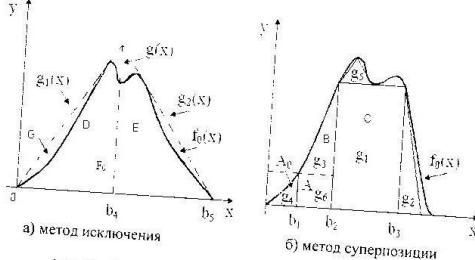


Рис. 5 - Вид функций плотности распределений, моделируемых методами исключения и суперпозиции

В тех случаях, когда ни метод обратной функции, ни метод исключения ввиду сложности функции плотности $f_0(x)$ трудно осуществить, моделирование случайной величины x с плотностью распределения $f_0(x)$ можно свести к моделированию случайного вектора (ξ^*, η^*) , равномерно распределенного в области F_0 (образованной кривой $y=f_0(x)$ и осью абсцисс) поскольку компонента ξ этого вектора имеет плотность распределения $f_0(x)$. Из-за сложного вида $f_0(x)$ моделирование равномерного распределения в области F_0 трудно осуществимо. Поэтому согласно методу суперпозиции область

F_0 разбивается на фиксированное число простых областей, моделирование равномерного распределения в которых осуществляется просто. При этом вероятности попадания в эти области равны их площадям. Используя эти вероятности вначале с помощью ярбия 2 лица поочередно "разыгрывают" номер области, в которой осуществляется моделирование с помощью типового ГСЧ.

Метод суперпозиции основан на формуле полной вероятности для плотностей. Пусть ξ^* и v -случайные величины, заданные на одном и том же вероятностном пространстве: $F_v(Z)$ -функция распределения

$P_{\xi^*, v}(x, z) -$ условная плотность распределены ξ^* при условии $v=z$.

Тогда безусловная плотность распределения ξ^* равна:

$$f_0(x) = \sum_{z=0}^{\theta} P_z^* / v(x, z) F_v(z) \quad (3.32)$$

Если v - дискретная величина со множеством значений $\{C_1, \dots, C_N\}$ и вероятностями $P(v=c_i) = P_i$ ($i=1, N; N < \infty\}$; $P_i = x/v_i / \sum_{j=1}^N x_j/v_j = f(x)$)

то функция безусловной плотности распределения принимает вид:

$$f_0(x) = \sum_{i=1}^N P_i \cdot f_i(x) \quad (3.33)$$

Моделирующий алгоритм включает следующие этапы:

- 1) Определение вспомогательной случайной величины v , удовлетворяющей условиям (3.32) и (3.33).
- 2) Моделирование v и считают, что Z есть реализация v .
- 3) Моделирование реализации случайной величины x с плотностью $f_z(x)$ (т.е. получение x - реализации ξ^*).

Графически эти операции можно проиллюстрировать примером на рис. 5.б. Из рис. 5.б видно, что область $F_0 = \{(x,y)\}; 0 \leq y \leq f_0(x); 6 \leq x \leq c$ разбивается на 6 непересекающихся частей $\{g_i\}_{i=1}^6$ с площадями $\{P_i\}$. Основной принцип разбиения заключается в том, что части g_i , имеющие наибольшую площадь и (наибольшую вероятность P_i), должны соответствовать просто и быстро моделируемым плотностям $f_i(x)$. На рис. 5.б число областей равно $n=6$, причем плотности $f_i(x)$ $i=1, 5$. Это распределения либо треугольного, либо ступенчатого вида. Они легко моделируются.

Остаточную плотность можно моделировать методом исключения:

$$f_0(x) = (f_0/x) - \sum_{i=1}^5 P_i f_i(x) / P_6, P_6 = (1 - \sum_{i=1}^5 P_i) \quad (3.34)$$

При моделировании случайных величин с функцией распределения сложной формы (как показано на рис. 5.6) нужны алгоритмы формирования псевдослучайных величин, у которых функция плотности имеет вид прямоугольного треугольника (распределение g_1 и g_2), обычного треугольника (распределение g_3), прямоугольника (распределение g_4) и прямоугольной трапеции (распределение g_5). На рис. 5.6 функция $g(x)=g_1(x)+g_2(x)$. Это означает, что нужно моделировать нормированное распределение (на интервале от 0 до 1) имеющее форму прямоугольника и прямоугольной трапеции. А далее осуществлялось обычное преобразование от нормированной величины к распределению, заданному на интервале $[b_1, b_2]$ или на интервале $[b_3, b_4]$.

Если распределение имеет форму прямоугольного треугольника (см. рис. 5.6 на интервале $[0, b_1]$ и площадь которого его равна $A_0=P_1$, то используя конгруэнтный метод и поэтому расчетная формула будет иметь

$$\xi_i = \frac{1}{2} A_0 \xi_{i-1}, \quad (3.35)$$

где $A_0 = 1/2 - b_1$; вероятность попадания ξ в область $[0, b_1]$.

Когда распределение имеет форму прямоугольника представлению на рис. 5.6 на интервале $[b_2, b_3]$ и площадь его равна вероятность попадания ξ в этот интервал, то согласно конгруэнтному способу расчетная формула имеет вид:

$$\xi_i = 2 \cdot C \cdot \xi_{i-1}. \quad (3.36)$$

Для случая, когда распределение имеет форму прямоугольной трапеции и площадь ее равна $(A+B)$, составные части которой представляют собой вероятность P_2 и P_3 попадания в интервал $[b_1, b_2]$, то согласно суперпозиции двух конгруэнтных методов расчетная формула примет вид:

$$\xi_i = (A+2B)\xi_{i-1}. \quad (3.37)$$

Наконец, при моделировании треугольного распределения, имеющего площадь равную $(D+E)$, где D и E – вероятности попадания, соответственно в области $[0, b_1]$ и $[b_3, b_4]$ (рис. 5).

Расчетная формула примет вид:

$$\xi_{i+1} = \frac{1}{2} D \cdot \xi_{i-1} + \frac{1}{2} E \xi_i. \quad (3.38)$$

Как видим, для формирования случайной величины, имеющей сложную форму функции плотности распределения, нужно несколько обраще-

ния

ний к ГСЧ и осуществить формульное преобразование. Во всех случаях разыгрывается жребий вероятности выбора необходимой части сложной функции плотности распределения, а затем уже проводится формальное преобразование значений ξ_i , полученных от ГСЧ. В тех случаях, когда это необходимо, имеет место обычный переход от нормированной случайной величины ξ_i к необходимому значению случайной величины, исходя из границы области нахождения этой случайной величины.

3.5 Моделирование случайных величин, имеющих мультимодальную функцию плотности распределения.

Все приведенные выше способы моделирования случайной величины x , имеющей функцию плотности, аналитически трудно представимую, являются весьма ресурсоемкими. В случае мультимодальной плотности распределения у случайной величины эти методы весьма громоздки и поэтому на практике гораздо проще использовать следующую технологию имитации случайной величины, которая оказывается при многократном имитационном эксперименте менее ресурсоемкой, чем рассмотренные ранее способы ее имитации.

Для изложения технологии имитации рассмотрим пример, когда случайная величина имеет мультимодальную функцию плотности распределения, представленную на рис. 6. В начале аппроксимируют функцию плотности $f(x)$ эмпирической ступенчатой функцией плотности $F(x)$. Затем формируют эмпирическую ступенчатую функцию плотности $F'(x)$ по известному алгоритму накопления $F'_k(x) = \sum f'_k(x)$, значения которой стремятся к единице.

На рис. 6 приведен также вид функции распределения $F(x)$. Используем рассмотренный ранее алгоритм формирования случайной величины по жребию третьего типа (случай с эмпирической функцией распределения). Допустим в k -й реализации ГСЧ сформировал значение величины ξ (см. рис. 3.5.). Тогда искомое значение случайной величины x в k -й реализации будет равно ξ_k^* . Как видим из рис. 6 значение ξ_k значением $F'(x)$ на каждом интервале группирования. Как только выполняется неравенство $F'_k(\xi_k) < \xi_k \leq F'_{k+1}(\xi_k)$, определяется искомая величина ξ_k^* , которая равна середине интервала $(k+1)$ группирования x .

В качестве примера использования процедуры Монте-Карло и генераторов псевдослучайных чисел рассмотрим моделирование поражения ракетами площадной цели со сложной структурой. В расположении нападающей стороны имеется n ракет. Зона разрушения одной ракеты представляет собой круг радиуса r . В результате n выстрелов ракеты будет поражена какая-то часть (S_n) общей площади S_0 . Чтобы увеличить долю по-

поражения цели $U = S_n / S_c$ и избежать ненужных перекрытий зон поражения, на площадной цели устанавливают n точек прицеливания (O_1, O_2, \dots, O_n) . Известны характеристики баллистического рассеяния ракет (по осям координат соответственно σ_x^2, σ_y^2). Предполагается, что систематические ошибки при попадании ракет на цель отсутствуют. Рассеяние ракет подчиняется двумерному нормальному закону распределения. Координаты точек попадания $(x_j, y_j), j = 1, n$ независимы друг от друга и от координат других точек попадания.

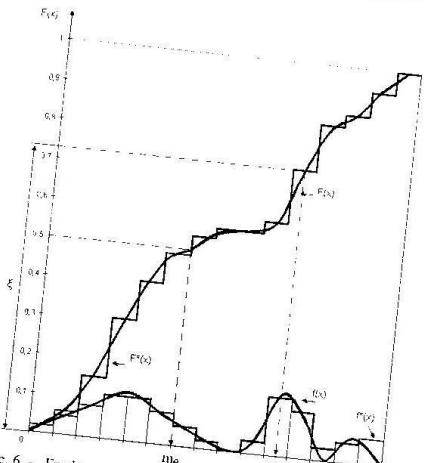


Рис. 6 - Графическая интерпретация моделирования случайных величин мультиомодального распределения $F(x)$

Требуется: при заданном расположении точек прицеливания O_1, O_2, \dots, O_n вычислить следующие характеристики обстрела: $m_u = M[U]$ - среднюю долю пораженной площади цели; D_u - дисперсию доли пораженияплощадной цели; $P(U > \bar{U})$ - вероятность того, что будет поражено не ме-

нее заданной доли площади \bar{U} ; r_b - математическое ожидание числа "промахов" ракет.

Если не делать никаких упрощений о форме цели и зонах поражения, то аналитически эту задачу решить трудно. Проще ее решить с помощью процедуры Монте-Карло. Каждая реализация представляет собой "обстрел" цели n ракетами, при котором все точки попадания (x_j, y_j) будут разыграны по жребию. Моделирование каждой i -й реализации будет состоять из единичных жребьев плюс расчет площади пораженной цели S_i и производных от нее характеристик поражения цели. На рис. 7 представлен пример попадания ракет наплощадную цель для i -й реализации по методу Монте-Карло.

Координаты попадания ракеты j распределены по двумерному нормальному закону с параметрами $(m_{xj}, m_{yj}, \sigma_x, \sigma_y)$, где m_{xj} и m_{yj} - координаты точек прицеливания $O_j, j = 1, n$. По условию независимости друг от друга x_j и y_j полагаем, что коэффициент корреляции $Z_{xy} = 0$.

Тогда координаты точек попадания определяются по формулам:

$$x_j = \sigma_x \sqrt{2} \left(\sum_{k=1}^{12} \xi_k - 3 \right) + m_{xj}; \quad (3.39)$$

$$y_j = \sigma_y \sqrt{2} \left(\sum_{k=7}^{12} \xi_k - 3 \right) + m_{yj}, \quad (3.40)$$

где ξ_k - 12 экземпляров случайных равномерно распределенных величин.

Для моделирования из ξ_k нормально распределенных величин используем способ сумм. Допустим, что этот обстрел смоделирован и мы получили n точек попадания ракет, которые при заданном распределении точек прицеливания O_j попали так, как показано на рис. 3.6. Вокруг каждой точки попадания опишем круг радиуса r и подсчитаем площадь той части цели, которая покрыта хотя бы одним из кругов. Всю цель разделим на ряд элементарных площадок dS_j . Для каждой площадки j определяется расстояние ρ_j от точки попадания j -й ракеты до центра площадки dS_j . Если хотя бы для одной из точек попадания j это расстояние оказалось равным или меньше $r(\rho_j \leq r)$, то считается dS_j пораженной, т.е. некоторая переменная-индикатор поражения площадки x_{ij} , которая в начале проверки площадок была равна 0, становится равной 1, если $\rho_j \leq r$. После чего производится суммирование пораженных площадок по всей цели ($S_p = \sum_{i,j} dS_j \cdot x_{ij}$).

Далее для i -й реализации процесса обстрела площади n ракетами ($i = 1, N$) долю S_{ik} пораженной площади U_i находят как отношение числа площадок dS_{ik} к общему числу элементарных площадок в цели:

$$U_i = S_{ik} / S_c. \quad (3.41)$$

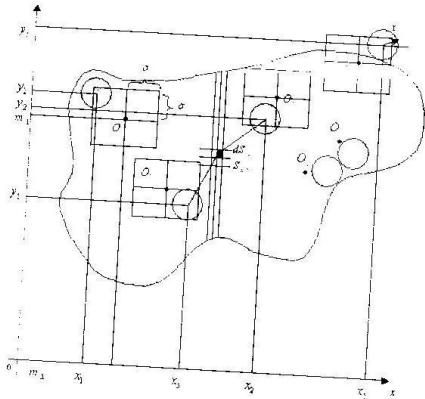


Рис. 7 – Пример попадания n ракет по площадной цели для i -й реализации по методу Монте-Карло

Одновременно для i -й реализации вычисляют количество ракет, расстояние которых от точек попадания больше, чем границы площадки S_n (т.е. находит число промахов r_{ik}). Математическое ожидание и дисперсия доли пораженной площади определяются из соотношений:

$$M_{ii} = \frac{1}{N} \sum_{k=1}^N U_{ik}; D_{ii} = \frac{1}{N} \sum_{k=1}^N U_{ik}^2 - M_{ii}^2. \quad (3.42)$$

Вероятность того, что доля пораженной площади будет не меньше, чем \bar{U} , определяют следующим образом. С каждой i -й реализацией формируется

$$Z_i = \frac{\text{Impat}_i \geq \bar{U}}{\text{Impat}_i > \bar{U}}, \quad (3.43)$$

Тогда $P(U \geq \bar{U}) = \frac{1}{N} \sum_{i=1}^N Z_i$. Наконец, определяют среднее число промахов ракет по цели:

$$\bar{r}_0 = \frac{1}{N} \sum_{k=1}^N r_{ik}. \quad (3.44)$$

Само же число реализаций N по методу Монте-Карло рассчитывается исходя из заданной доверительной вероятности β . Находит такое число элементов выборки, чтобы была обеспечена требуемая доверительная вероятность β . Расчетная формула имеет вид

$$N \geq S^2 / (\beta) \Delta^2, \quad (3.45)$$

где $\Delta = k S_x$;

$$S_x = S / \sqrt{N};$$

$k = 1, 2, 3, \dots$;

S – выборочное среднеквадратическое отклонение;

Δ – допустимая (требуемая) толщина оценки дисперсии вычислений.

Лекция 4 СПОСОБЫ ОРГАНИЗАЦИИ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ СЛОЖНЫХ СИСТЕМ

- 4.1 Понятие о модельном времени.
- 4.2 Структурная схема имитационных моделей сложных систем.
- 4.3 Организация квазипараллелизма в имитационных моделях путем просмотра активностей.
- 4.4 Организация квазипараллелизма в имитационных моделях способом составления расписания событий.
- 4.5 Способы организации квазипараллелизма в имитационных моделях.

4.1 Понятие о модельном времени.

Для того чтобы понять механизм имитации функционирования СС на ЭВМ, рассмотрим два примера.

Пример 1. Объектом имитации является движение i -й космической ракеты. Можно выделить ряд стадий ее движения, в ходе которых выполняется последовательность функциональных действий (ФД). Пусть каждая стадия движения ракеты характеризуется своим номером j . При $j=1$ осуществляется запуск ракеты i на стартовой площадке. На этой стадии выполняется функциональное действие ΦD_1 . Коррекция движения и сброс первой ступени i -й ракеты на стадии 2 определяют ΦD_2 . Вторая коррекция движения и сброс второй ступени i -й ракеты означают выполнение ΦD_3 . Движение множества ракет представляет собой СС, и каждая i -я ракета является компонентой K_i .

Пример 2. Объектом имитации является последовательная обработка i -й детали на нескольких станках. Каждый j -й станок обеспечивает выполнение своего набора операций обработки детали i , реализуя таким образом функциональное действие ΦD_j . Обработку множества деталей можно рассматривать как СС, в которой каждая i -я деталь является компонентой K_i .

В обоих случаях функционирование компоненты K_i СС представляет собой последовательность ΦD_j . Будет говорить, что в результате выполнения ΦD_j в СС происходит событие C_j . Каждое из событий в реальной СС связано, как правило, с соответствующей компонентой K_i . При этом любое действие ΦD_j выполняется на некотором временном интервале t_j . Для каждой K_i можно, однако, характер этих изменений различен и определяется последовательностью временных интервалов $\{t_j\}$.

При построении ИМ СС ΦD_j аппроксимируются некоторыми упрощенными функциональными действиями $\Phi D'_j$. Степень этого упрощения определяет уровень детализации ИМ. Отличия $\Phi D'_j$ от ΦD_j порождают ошибки имитации реальной СС. В ИМ ΦD_j представляется парой $(\Phi D'_j, t_j)$, которая выполняется следующим образом. Вначале реализуется $\Phi D'_j$ при

неизменном значении t_i , а затем уже отображается изменение t_i на величину t_j , инициируя таким образом появление события C_j .

На рис. 8 представлен пример развития действий K_i в системе координат $(\Phi D'_j, t_i)$. Конечно, подобная система координат носит условный характер, поскольку по оси ординат нельзя отложить «значение» функционального действия $\Phi D'_j$. Подобное представление используется только для изображения того факта, что с изменением «локального времени» t_i некоторая K_i выполняет несколько различных ΦD_j . Так, согласно рис. 8, K_i последовательно выполняет $\Phi D_{11}, \Phi D_{12}, \Phi D_{13}$, а в СС соответственно происходят события C_{11}, C_{12}, C_{13} . Причем с появлением каждого нового C_{ij} происходит изменение ΦD_j и увеличение его временной координаты t_i , соответственно на величину t_{ij} . Условие на рис. 8 появление событий в K_i при выполнении ΦD_{11} показано штрихпунктирной линией. В ИМ появление реализуется ступенчатой линией (0, a, C₁₁, b, C₁₂, d, C₁₃). Это означает, что вначале выполняется $\Phi D'_{11}$ при неизменном t_i , а затем уже отображается изменение t_i на величину t_{11} , инициируя появление события C_{11} . Затем реализуется $\Phi D'_{12}$ при неизменном времени t_i , на величину t_{12} , инициируя в K_i свершения события C_{12} . Далее в два приема (реализация $\Phi D'_{13}$ и последующее изменение t_i на величину t_{13}) аналогичным образом инициируется событие C_{13} . Отметим при этом, что порядок появления событий может быть и обратным: сначала изменяется t_i , а затем уже выполняется $\Phi D'_{11}$.

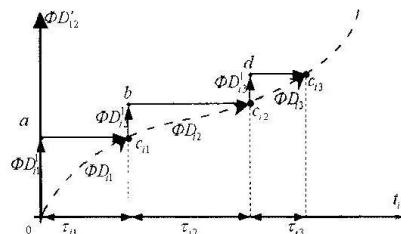


Рис. 8 - Пример аппроксимации функциональных действий i -й компоненты системы (K_i) в имитационной модели СС

В ИМ каждое $\Phi\mathcal{U}_i$ описывается в общем случае некоторым алгоритмом АЛ_i . В ходе имитации происходит реализация $\Phi\mathcal{U}_i$ на соответствующим алгоритмом АЛ_i и последующее изменение t_i на величину t_{ij} . Таким образом, любая ИМ описывается набором некоторых «молекул», каждая из которых содержит в себе описание алгоритма выполнения АЛ_i, соответствующего $\Phi\mathcal{U}_i$ и оператора Мт_i, осуществляющего изменение временной координаты t_i на величину t_{ij} . Пару ($\text{АЛ}_i, t_{ij}$) обычно называют *ij-й активностью* ИМ и обозначают АК_{ij}. Иногда активности называют «рабочими» [60]. Любая ij-я активность представляет собой запись поведения компоненты K_i имитационной модели СС. Реализация этой активности в ИМ приводит к появлению в модели СС события С_{ij}.

Если бы на ЭВМ имитировалось поведение только одной компоненты системы, то выполнение активностей в ИМ можно было бы осуществить строго последовательно, и дело свелось бы к пересчету временной координаты t_i после очередного выполнения алгоритма АЛ_i. В действительности СС состоит из нескольких компонент. Все эти компоненты K_i функционируют одновременно. Это должна отражать ИМ. В большинстве современных ЭВМ в каждый момент времени может реализовываться алгоритм только из компонент модели.

Чтобы обеспечить имитацию параллельных событий реальной системы, вводят некоторую глобальную переменную t_0 , которую называют модельным (системным) временем. С помощью этой переменной организуются синхронизация всех событий С_{ij} в модели и выполнение алгоритмов АЛ_i компонент K_i модели системы.

При реализации ИМ используются обычно три представления времени: t_K - реальное время системы, работа которой имитируется в ИМ; t_i - модельное время, по которому организуется синхронизация событий в системе; t_0 - машинное время имитации, отражающее затраты ресурса времени ЭВМ на организацию имитации.

С помощью модельного времени t_0 реализуется квазипараллельная разделятельный характер обслуживания событий в ИМ, одновременно возникающих в разных компонентах реальной системы.

Корректировка временных координат t_i нескольких K_i ИМ осуществляется с помощью модельного времени t_0 следующим образом. Если изменения t_i при выполнении АЛ_i нескольких K_i совпадают (это означает, что в реальной системе происходит одновременно несколько событий С_{ij}), то последовательно обслуживаются АЛ_i, совпадающие по времени выполнения, т.е. имеющие одинаковые значения t_0 . Здесь и далее под t_0 будем понимать конкретное значение t_0 , при котором происходит событие С_{ij}. При павших по времени реализации алгоритмов АЛ_i. Таким способом последовательно выполняются соответствующие $\Phi\mathcal{U}_i$ при неизменном значении t_0 .

После каждой реализации АЛ_i, обеспечивающей выполнение в ИМ $\Phi\mathcal{U}_i$, выполняется оператор корректировки временной координаты Мт_i. Чаще всего эта корректировка сводится к вычислению нового значения t_i по формуле:

$$t_{ij} = t_0 + t_{ij} \quad (4.1)$$

Это значение временной координаты t_i запоминается и используется в дальнейшем для определения момента новой активизации в ИМ компоненты K_i. Под активизацией компоненты модели K_i будем понимать начало выполнения следующей ее активности (выполнение алгоритма АЛ_i и оператора корректировки временной координаты t_i).

Когда имитация одновременно появившихся событий С_{ij} завершена, выполнены соответствующие алгоритмы активностей АЛ_i и проведены корректировки временных координат t_i , меняется значение глобальной переменной модели t_0 . Существуют два способа изменения t_0 : с помощью фиксированных и переменных интервалов изменения модельного времени. Часто их называют соответственно способами фиксированного шага и шагов до следующего события.

Для того чтобы легче было представить оба способа организации изменения модельного времени, рассмотрим следующий пример (рис. 9). Пусть в системе функционируют три компоненты K_i ($i = 1, 3$). При функционировании K_i последовательно происходят четыре события (С₁₁, С₁₂, С₁₃, С₁₄) соответственно четырем моментам изменения t_1 ($t_{11}, t_{12}, t_{13}, t_{14}$). Между этими моментами K₁ выполняет четыре различных функциональных действия (ФД₁₁, ФД₁₂, ФД₁₃, ФД₁₄). Каждое из указанных функциональных действий выполняется в течение соответствующих интервалов времени ($t_{11}, t_{12}, t_{13}, t_{14}$). Апроксимация ФД₁ осуществляется последовательностью $\{\Phi\mathcal{U}_{1j}\}$, $j = 1, 4$. Аналогичным образом K₂ последовательно выполняет три функциональных действия $\{\Phi\mathcal{U}_{2j}\}$, $j = 1, 3$ в течение трех интервалов времени (t_{21}, t_{22}, t_{23}), которые в ИМ аппроксимируются соответственно $\{\Phi\mathcal{U}_{2j}\}$. Наконец, K₃ также последовательно выполняет три функциональных действия $\{\Phi\mathcal{U}_{3j}\}$, $j = 1, 3$ соответственно в течение трех интервалов времени (t_{31}, t_{32}, t_{33}), аппроксимируемых в ИМ соответственно тремя $\{\Phi\mathcal{U}_{3j}\}$. Отметим, что каждое $\Phi\mathcal{U}_i$ описывается соответствующим алгоритмом АЛ_i и реализуется соответствующей активностью, представляющей собой выполнение этого алгоритма при неизменном значении t_i и модификацию временной координаты t_i на величину t_{ij} . Алгоритм функционирования K_i в координатах

(Φ_{D_i}, t_i) представлен на рис. 9. В ходе выполнения активностей (AL_j, τ_j) в модели К, происходят последовательно события C_{ij} .

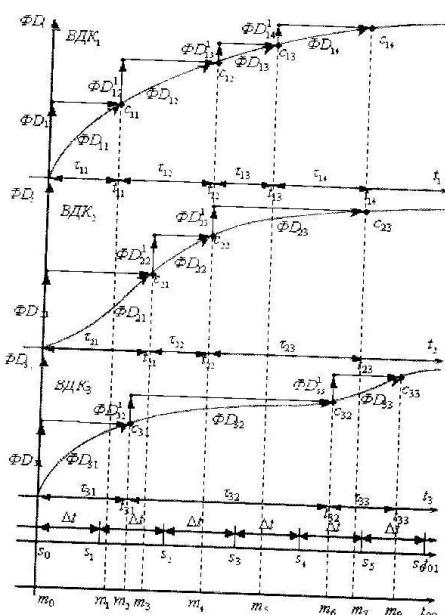


Рис. 9 - Временная диаграмма моделирования событий в СС, состоящей из трех компонент

Рассмотрим имитацию событий C_3 в системе при каждом из способов изменения t_0 . При имитации по способу фиксированного шага модельное время, обозначенное на рис. 9, как t'_0 , меняется каждый раз на величину шага Δt . Тогда в моменты времени $0, \Delta t, 2\Delta t, 3\Delta t$ и т.д. в ИМ происходят соответственно события S_0, S_1, S_2 и т.д., означающие переход на обслуживание управляющей программой моделирования тех событий C_{ij} , которые попадают внутрь очередного интервала Δt . Как видно из рис. 9, в действительности события C_{11}, C_{31}, C_{21} происходят на интервале (S_0, S_1) . Считается, что они происходят одновременно в момент $t' = \Delta t$. Управляющая программа моделирования (УПМ) последовательно обслуживает эти события, что сводится к инициализации и выполнению соответствующих активностей $(AL_{11}, \tau_{11}; AL_{31}, \tau_{31}; AL_{21}, \tau_{21})$. Затем в момент $t'_0 = \Delta t$ УПМ инициирует событие S_2 . Считается, что события C_{12} и C_{22} также происходят одновременно, поэтому УПМ последовательно инициирует выполнение активностей (AL_{12}, τ_{12}) и (AL_{22}, τ_{22}) . Время t'_0 получает приращение на заранее выбранную величину Δt , а моменты реагирования УПМ на появление событий S_1 не связаны с моментами t_y появления событий в реальной системе. Таким образом, точность моделирования событий C_3 определяется шагом Δt изменения модельного времени t'_0 , и, как правило, все события обслуживаются в точке, соответствующей верхней границе интервала изменения t'_0 .

При имитации по способу шагов до следующего события время t'_0 меняется в моменты t_y , которые соответствуют моментам t_y появления событий в реальной системе. Обработку событий C_{ij} , одновременно появляющихся в реальной системе, УПМ осуществляет последовательно при неизменном модельном времени t'_0 . Например, события C_{12} и C_{22} в реальной системе происходят одновременно. В модели вначале реализуется, например, алгоритм AL_{12} и модифицируется координата t_1 на величину t_{12} , а затем выполняется алгоритм AL_{22} и модифицируется координата t_2 на величину t_{22} . Таким образом, t'_0 каждый раз сдвигается на величину, равную минимальному значению приращения модельного времени t_y , которое получают координаты t_i .

Независимо от способа изменения t_0 механизм регламентации изменения модельного времени обычно предусматривает выполнение следующих действий:

- выбор событий в модели, которые необходимо обслужить при одном и том же модельном времени t_0 ;
- обслуживание событий (инициализация активностей), которые имеют одинаковое время инициализации;
- по окончании обслуживания всех одновременных (в пределах шага) событий определение очередного значения модельного времени;

- корректировка временной координаты модели t_0 ;
- проверка условий окончания моделирования либо по времени завершения имитации, либо по выполнению других событий в системе.

Все эти действия выполняет УПМ. Каждое событие C_j в модели системы, являющееся результатом выполнения активности AK_i , обслуживается УПМ и требует для реализации некоторого ресурса времени работы ЭВМ. Поэтому чем чаще обслуживаются события УПМ, тем больше возрастают машинное время t_s . С этой точки зрения способ фиксированного шага изображения t_s является более экономичным, поскольку имеет место групповое обслуживание всех событий C_j , которые попали внутрь очередного шага изменения Δt модельного времени.

На практике предпочтение способу фиксированного шага отдаётся в двух случаях. Во-первых, когда события C_j распределены равномерно на всем интервале моделирования и исследователь может подобрать интервал времени изменения временной координаты Δt , обеспечивающий минимальную погрешность имитации. Во-вторых, когда событий очень много и они являются группами. Во всех остальных случаях способ задания шага до следующего события более предпочтителен. В тех случаях, когда события C_j не равномерны и появляются они через значительные временные интервалы t_{ij} , способ задания шага до следующего события экономичнее и точнее способа фиксированного изменения t_s . С учетом достоинств и недостатков каждого из способов изменения t_s наметилась специализация их применения при моделировании СС.

На практике наибольшее распространение получил способ шага до следующего события. Поэтому в дальнейшем будем рассматривать все изменения модельного времени.

4.2 Структурная схема имитационных моделей сложных систем.

Как уже установлено, в ИМ происходит аппроксимация каждого ФД_{ij} парой действий: упрощенным функциональным действием Φ''_{ij} и модифицией временной координаты i-й компоненты модели t_i . Эта пара действий представляет собой «молекулу» ИМ и называется активностью AK_{ij} . Любая AK_{ij} состоит из описания алгоритма AL_{ij} и оператора модификации временной координаты компоненты модели M_{ij} . Для синхронизации выполнения компонент ИМ на ЭВМ используется глобальная переменная, называемая модельным временем t_0 . Кроме модификации модельного времени t_0 и выполнения операторов M_{ij} в функции УПМ входят: запуск на гом в ходе имитации и проверка условий окончания имитации. Таким образом, любая ИМ представляет собой совокупность набора «молекул», отражающих поведение объекта имитации, и УПМ, организующей взаимодействия этих «молекул» друг с другом.

Для задания входных условий, начальных значений параметров и запуска ИМ в состав модели включается подпрограмма (ПП), организующая начало имитации. Им создается для изучения на основе статистики, характеризующей поведение компонент К, реальной системы в различных режимах ее функционирования. Для сбора статистики моделирования в ИМ вводится соответствующая программа, которая присутствует в ИМ либо в явном виде, либо рассредоточена по всем компонентам ИМ. Наконец, по достижении условий окончания имитации УПМ передает управление на подпрограмму окончания имитации, которая зачастую и обеспечивает расчет характеристик поведения модели системы и выдает исследователю результаты моделирования. Все эти подпрограммы взаимодействуют только с УПМ.

Таким образом, описание ИМ существенно разрастается в объеме по сравнению с собственно описанием объекта имитации. Можно говорить о двух частях ИМ. Первая часть ИМ является переменной и создается исследователем системы с помощью имеющихся средств автоматизации моделирования. Вторая часть ИМ представляет собой реализацию этих средств автоматизации моделирования в виде: процедур, обеспечивающих выполнение операторов синхронизации модели (например, M_{ij}), подпрограмм запуска и завершения имитации УПМ и подпрограмм сбора статистики моделирования.

На рис. 10 представлена схема перевода объекта моделирования в его имитационную модель. Из рисунка видно, что любое ФД_{ij} в ИМ описывается соответствующей активностью AK_{ij} . Каждая AK_{ij} представляет собой попарное сочетание описания алгоритма AL_{ij} и оператора модификации временной координаты компоненты модели M_{ij} . УПМ передает управление на начало программы, реализующей алгоритмы AL_{ij} . Возврат на УПМ имеет место при выполнении операторов M_{ij} . УПМ и служебные подпрограммы обычно являются универсальными и не изменяются при переходе от одной модели к другой. Однако принципы их построения и способ управления выполнением активностей зависят от класса объектов моделирования. Вопросы организации квазипараллельного выполнения активностей интересны для специалистов по созданию средств автоматизации моделирования. Составление же текста описания ИМ и использование системных средств при разработке AL_{ij} являются областью интересов исследователей функционирования СС.

В моделях сложной систем состав активностей в разных компонентах и характер взаимодействия друг с другом могут быть различными. В одних системах все ФД_{ij} различны и для реализации каждого алгоритма AL_{ij} активности в модели требуется выполнение своих условий. В других системах некоторые ФД_{ij} в разных компонентах аналогичны, совпадают по времени и приводят к одному и тому же событию C_{ij} . Для реализации в ИМ

РЕПОЗИТОРИЙ ГРУШИ

каждой такой группы ΦD_i одним алгоритмом AL_i требуется выполнение своих определенных условий.

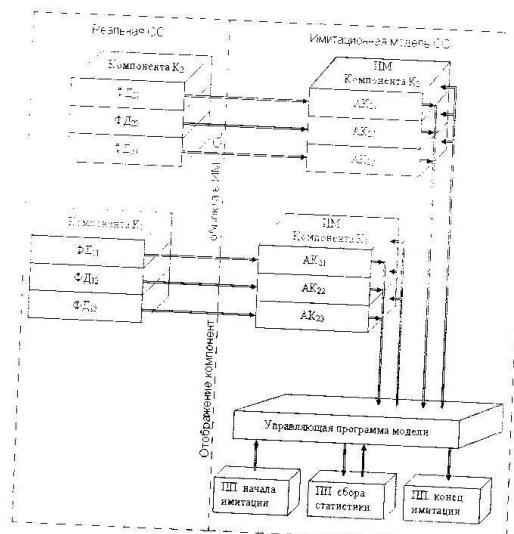


Рис. 10 - Различие между объектом моделирования и его имитационной моделью

Для некоторых систем существует сильное взаимодействие между ΦD_i , которые можно аппроксимировать явно задаваемыми математическими зависимостями при создании ИМ. В зависимости от состава алгоритмов AL_i , наличия связей между компонентами, целей и задач моделирования выбирается тот или иной способ представления K_i и реализации активностей в имитационных моделях. Чаще всего составление описание ИМ называют

формализацией объекта моделирования. Наибольшее распространение получили пять способов описания ИМ: непосредственно активностями, аппаратом событий, транзактами, агрегатами, процессами. Каждому способу формализации объекта моделирования соответствует свой способ организации квазипараллелизма обслуживания УПМ активностей, из которых составлена ИМ. Поэтому различают соответственно следующие способы расписания событий, просмотр активностей, составление расписаний событий, управление обслуживанием транзактов, управление агрегатами, синхронизация процессов.

Поэтому различают следующие способы организации квазипараллелизма в ИМ: путем просмотра активностей; составления расписания событий; имитации с помощью потоков транзактов; представления СС набором агрегатов; путем просмотра процессов; взаимодействия транзактов с процессами; организация взаимодействия агрегатов с процессами.

Одну и ту же СС принципиально можно представить любым из перечисленных способов формализации. Однако ИМ, построенные на их основе, будут отличаться размерами и качеством ресурсов, затрачиваемых на их создание, испытание и использование.

Однажды различают особенности и принципы организации квазипараллелизма в ИМ каждым из перечисленных способов рассмотрим на примере, в котором в качестве объекта моделирования выбрана система, состоящая из трех компонент K_i (рис. 11). В соответствии с временной диаграммой перехода K_i в состояния C_i в системе координат ΦD_i время его выполнения (см. рис. 9) установлено следующее. Компонента K_1 выполняет четыре функциональные лягости ($\Phi D_{11}, \Phi D_{12}, \Phi D_{13}, \Phi D_{14}$) во времени t_1 , которые в ИМ описываются соответственно четырьмя активностями. Аналогичным образом компоненты K_2 и K_3 выполняют по три функциональных действия ($\Phi D_{21}, \Phi D_{22}, \Phi D_{23}$) во временах соответственно t_2 и t_3 . Отпуская для простоты некоторые детали, рассмотрим основные функции УПМ и принципы ее взаимодействия с компонентами модели системы K , при каждом из способов организации квазипараллелизма.

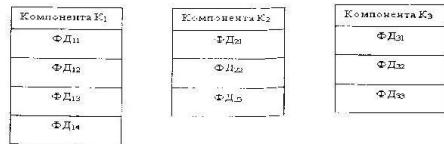


Рис. 11 - Структура обмена, используемого для иллюстрации способов имитации

4.3 Организация квазипараллелизма в имитационных моделях путем просмотра активностей.

Данный способ организации квазипараллелизма используется при моделировании реальных систем, характеризующихся следующим. Все ФД из них требуется выполнение своих условий. Эти условия конкретны, известны заранее исследователю и могут быть представлены алгоритмически. В результате выполнения ФД, в системе происходят различные события С. Связи между ФД отсутствуют, все ФД функционируют независимо друг от друга. В таких случаях исследователь описывает ИМ в виде двух частей: множества активностей $\{AK_i\}$ и набора процедур проверки выполнимости условий инициализации активностей. Пол инициализацией АК понимают передачу управление от УПМ на выполнение АД, данной активности. В процедуре проверки выполнимости условий инициализации АК реализуется зависимость выполнения Φ/Γ_i от конкретной ситуации на ЭВМ алгоритма АД_i данной активности назовем обслуживанием АК_i. Завершается обслуживание АК_i выполнением оператора модификации временных координат компоненты M_{ij} . По этому оператору управление возвращается УПМ. Таким образом, имитация на ЭВМ представляет собой чередование выполнения алгоритмов активностей, операторов модификации временных координат t_i и алгоритма УПМ. Вся ИМ представляет собой набор активностей AK_i , каждая из которых после выполнения на ЭВМ возвращает управление УПМ. Схема взаимодействия активностей с УПМ представлена на рис. 12.

Перед началом имитации соответствующая подпрограмма устанавливает значения начальных состояний компонент модели К, и параметров, определяющих условия инициализации активностей AK_i . Обычно проверка выполнимости условия инициализации AK_i состоит либо в определении значений параметров модели, либо в вычислении моментов t_i , в которые должно начаться выполнение соответствующего ФД_i, либо в проверке значений переменных модели. С помощью алгоритма реализуется своя последовательность проверок выполнимости условия инициализации AK_i . Для простоты предполагаем, что каждая AK_i имеет только одну проверку условия ее инициализации (см. рис.12). Отметим, что пользователь описывает не только алгоритмы активностей AK_i и операторы модификации временных координат M_{ij} , но и алгоритмы проверки условий инициализации AK_i .

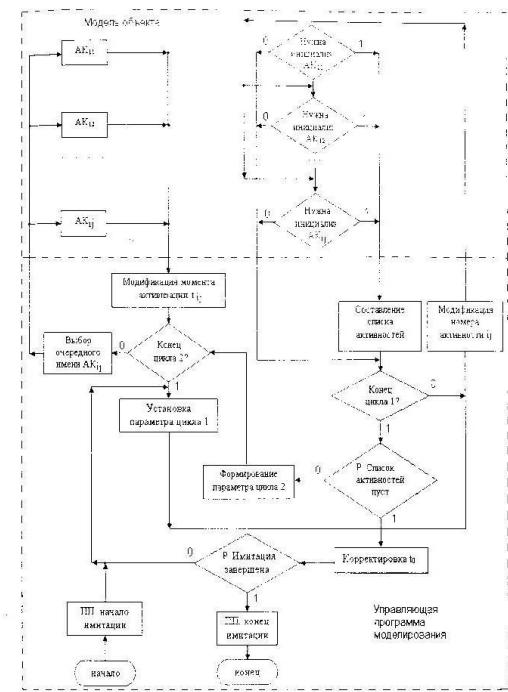


Рис. 12 - Блок схема ИМ объекта, состоящей из активностей

Алгоритм инициализации УПМ AK_j состоит в следующем. По первому циклу (параметр цикла равен максимальному числу AK_i в ИМ) управление последовательно передается на проверку выполнимости условий активизации AK_i . Имена тех активностей, для которых эти условия выполнены, заносятся в список инициируемых AK_j . По завершении этого цикла начинается второй цикл (параметр цикла равен числу инициализируемых AK_j), в котором осуществляется последовательная передача управления на выполнение АЛ_j (обслуживание AK_j). Каждая AK_j завершается оператором M_{t_j} , который вычисляет новый момент t_i инициализации AK_j и возвращает управление на продолжение просмотра второго цикла. Может оказаться, что при выполнении некоторых AK_i инициализированных на втором цикле, вновь устанавливаются соответствующие значения параметров и переменных, указывающие на то, что выполнились условия инициализации некоторых AK_i . Поэтому управление передается на начало первого цикла для повторного просмотра выполнимости условий инициализации AK_j . Только в случае, когда после первого цикла список инициализируемых AK_j пуст, УПМ корректирует t_i минимальным значением момента инициализации активности t_{j_1} . Все t_i заносятся в упорядоченный по их значениям список моментов инициализации AK_j . Когда завершается обслуживание некоторой AK_j , в этот список заносится значение t_i и имя соответствующей AK_j . После корректировки t_i проверяется выполнимость условия завершения имитации. Если это условие не выполнено, то имитация продолжается и управление передается на начало первого цикла. Когда условие завершения имитации выполнено, имитация прекращается и управление передается соответствующей подпрограмме, которая формирует результаты моделирования.

Поскольку выполнение алгоритмов одних активностей АЛ_j может привести к инициализации других AK_i , то в УПМ возможны повторные циклы проверки выполнимости условий инициализации AK_j . При этом события C_{ij} в ИМ не регламентированы, а лишь указываются условия, при которых они могут произойти. Отметим, что инициализация AK_j и сдвиг ее временной координаты оператором M_{t_j} разрешаются только тогда, когда выполняются все условия ее инициализации. Если хотя бы одно из этих условий остается невыполненным, то соответствующая AK_j не попадает в список инициализируемых активностей.

Иногда удельный вес безуспешных проверок очень велик УПМ. Данный способ организации квазипараллелизма AK_j выделен только при наличии достаточно простых алгоритмов проверки выполнимости условий инициализации активностей.

4.4 Организация квазипараллелизма в имитационных моделях способом составления расписания событий.

Данный способ организации квазипараллелизма используется при моделировании реальных систем, характеризующихся следующим. Различные компоненты К, выполняют один и те же функциональные действия Φ_{D_i} . Начало выполнения этих Φ_{D_i} определяется одними и теми же условиями, которые также заранее известны исследователю и могут быть представлена алгоритмически. В результате выполнения одних и тех же Φ_{D_i} в системе происходит одинаковые события C_{ij} независимо друг от друга. Связи между различными Φ_{D_i} отсутствуют, каждое Φ_{D_i} выполняется независимо друг от друга.

Пусть в нашем примере (см. рис. 11) у компонент системы совпадают следующие Φ_{D_i} , приводя к одним и тем же событиям:

- $\Phi_{D_1}, \Phi_{D_2}, \Phi_{D_3} -$ к событию C_{11} ;
- $\Phi_{D_1}, \Phi_{D_2}, \Phi_{D_3} -$ к событию C_{12} ;
- $\Phi_{D_1}, \Phi_{D_2}, \Phi_{D_3} -$ к событию C_{13} ;
- $\Phi_{D_1} -$ к событию C_{14} .

В таких случаях исследователь также описывает ИМ в виде двух частей: множества активностей $\{AK_i\}$ и набор процедур проверки появления событий и инициализации соответствующих активностей. При этом каждая AK_i имитирует выполнение группы связанных функциональных действий Φ_{D_i} у различных компонент К системы. В нашем случае AK_{11} инициализируется при появлении события C_{11} и имитирует выполнение $\Phi_{D_{11}}$, $\Phi_{D_{12}}, \Phi_{D_{13}}$. Аналогичным образом инициируются: AK_{12} при появлении события C_{12} , AK_{13} при появлении события C_{13} и AK_{14} при появлении события C_{14} . Инициализация AK_i имеет тот же смысл, что и для предыдущего способа организации квазипараллелизма, и выполняется УПМ.

В процедурах проверки появления событий C_{ij} реализуется зависимость выполнения Φ_{D_i} от конкретной ситуации, имеющей место в реальной системе. Выполнение алгоритма АЛ_j также называется обслуживанием AK_j , которое завершается оператором модификации временной координаты M_{t_j} активности, обслуживающей группу одинаковых Φ_{D_i} у разных компонент реальной системы. По этому оператору управление возвращается на УПМ. Часто такие групповые активности называют процедурами обслуживания событий C_{ij} . Таким образом, ИМ состоит из двух типов процедур: проверки выполнимости событий C_{ij} . Таким образом, ИМ состоит из двух типов процедур: проверки выполнимости событий C_{ij} в модели системы и обслуживания события C_{ij} .

Выполнение этих процедур синхронизируется в модельном времени списковым механизмом планирования УПМ. Каждый элемент этого списка определяет момент t_j свершения события C_{ij} , а также имя или номер той процедуры обслуживания событий, которая должна выполняться после завершения этого события.

Схема взаимодействия процедур, составляющих описание ИМ с УПМ представлена на рис.13. Перед началом имитации соответствующая подпрограмма устанавливает начальные состояния компонент модели системы и задает исходные значения параметрам, определяющим условия инициализации групповых активностей АК_g. Принципы проверки выполнимости условий появления ИМ событий С_i те же, что и при имитации активностями. Здесь также с помощью соответствующих алгоритмов пользователь составляет описание процедур проверки условий появления событий С_i.

Алгоритм инициализации УПМ процедур обслуживания событий состоит в следующем. По первому циклу (параметр цикла равен максимальному числу различных событий С_i в модели системы) управление последовательно передается на проверку выполнимости условий появления событий. Имена тех событий, для которых эти условия выполнены, заносятся в список инициализируемых событий. По завершении этого цикла проверяется, имеется ли в этом списке хотя бы один элемент. Если список событий не пустой, то начинается второй цикл (параметр цикла равен числу свершившихся событий С_i), в котором осуществляется последовательная передача управления на выполнение процедур обслуживания событий. Каждая такая процедура реализует алгоритм соответствующей групповой активности и завершается оператором модификации временной координаты М_{tj}, который модифицирует значение момента свершения данного события в будущем и возвращает управление на продолжение второго цикла.

Для того чтобы учсть появление новых событий во время обслуживания списка свершившихся событий, по окончании цикла 2 устанавливается начальное значение параметра цикла 1 и повторяется просмотр условий свершения событий. Как только список свершившихся событий окажется пустым, УПМ корректирует модельное время. Корректировка осуществляется подпрограммой-календарем, которая ищет в списке запланированных событий минимальное значение момента инициализации t₀, которое становится новым значением модельного времени t₀. После корректировки t₀ проверяется выполнимость условия завершения имитации. При невыполнении этого условия имитация продолжается и управление передается на установку параметра цикла 1. Когда условие завершения имитации выполнено, она прекращается и управление передается соответствующей подпрограмме, которая вычисляет результаты моделирования и выдает их исследователю.

Из приведенного примера видно, что объединение нескольких активностей в группы существенно сокращает размеры циклов 1 и 2, что значительно уменьшает накладные расходы на организацию имитации УПМ. Поэтому для некоторых случаев (условия применения способа оговорены в начале данного параграфа) линейный способ имитации более экономичен, чем способ имитации активностями. Однако у рассматриваемого способа имеется существенный недостаток, который заключается в следующем.

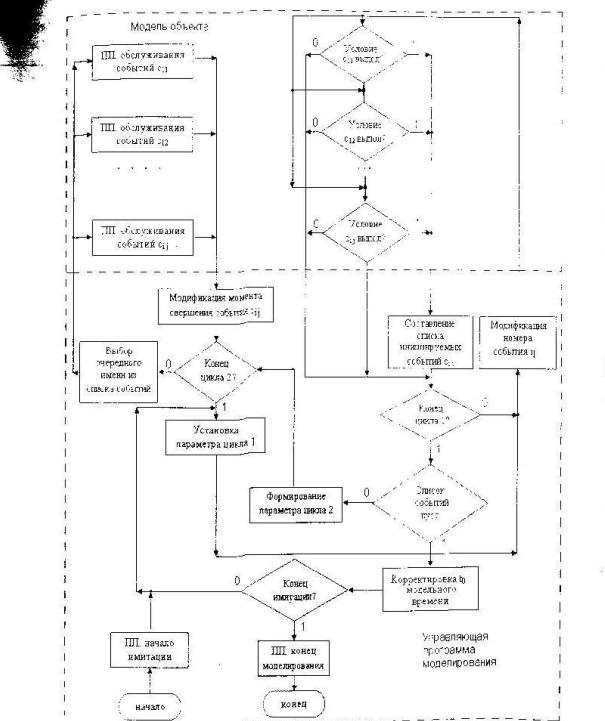


Рис. 13 - Блок-схема имитации способом обслуживания событий

Из-за того, что происходит объединение активностей различных компонент в составе процедур обслуживания событий, описание ИМ может потерять сходство со структурой реальной системы. Зачастую в рамках одной процедуры обслуживания событий ИМ могут обслуживаться даже логически не связанные друг с другом активности, но приводящие к одиличим и тем же событиям. Это обстоятельство может затруднить анализ результатов моделирования и модификацию ИМ.

4.5 Способы организации квазипараллелизма в имитационных моделях.

а) транзактный способ.

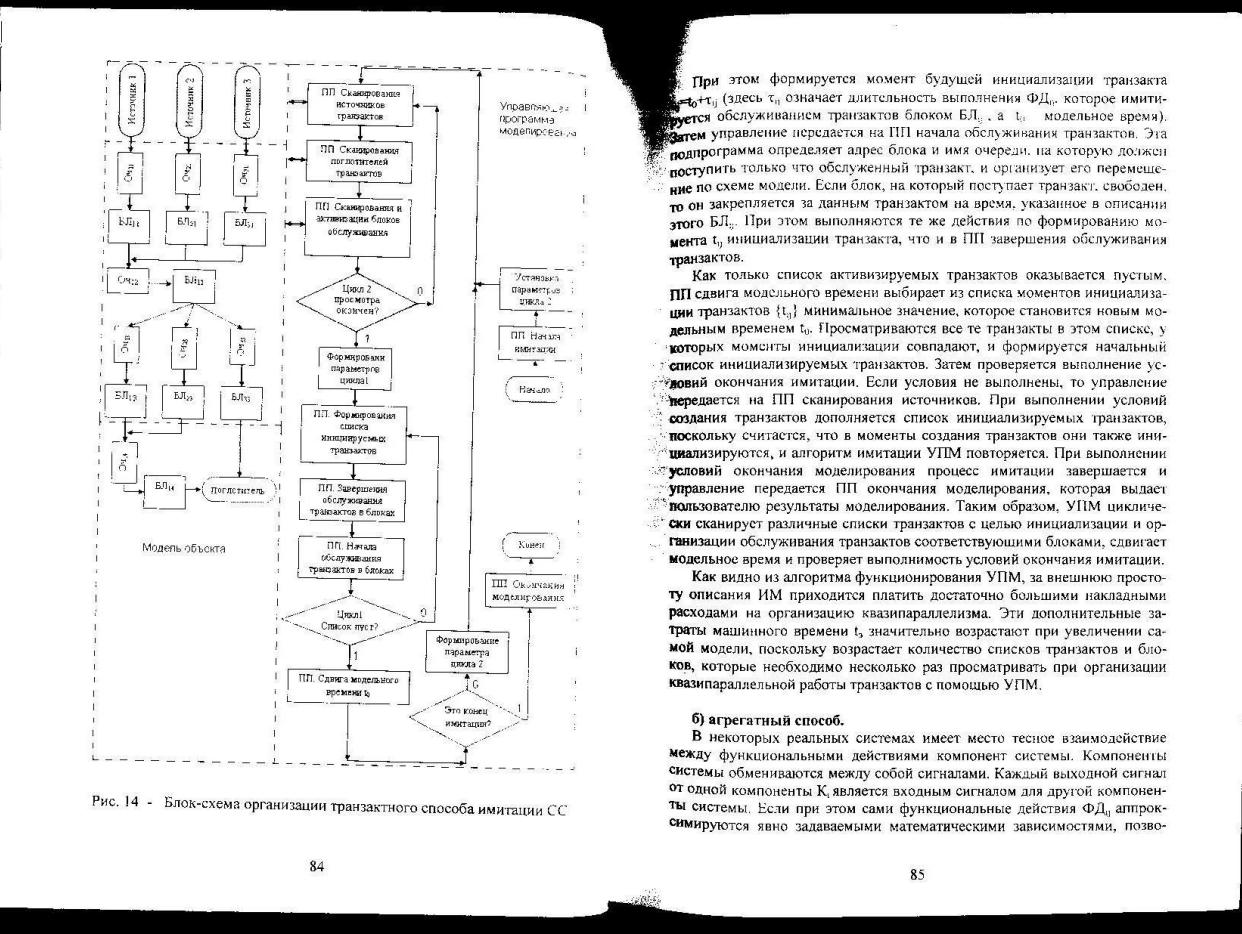
В ряде случаев функциональные действия Φ_D , представляет собой набор простейших операций, и его можно аппроксимировать активностями, алгоритмы выполнения которых лишь корректируют значения временных координат t_i , компонент системы. Важно и то, что существует зависимость выполнения Φ_D , друг от друга, которую удобно представить в виде схемы. Взаимодействие такого вида активностей аналогично работе систем массового обслуживания (СМО). Для имитации поведения реальной системы в таких случаях используется транзактный способ организации квазипараллелизма в ИМ. Однотипные активности пользователем объединяются и называются приборами массового обслуживания. Инициаторами появления событий C_i в ИМ становятся заявки (транзакты) на обслуживание этими приборами массового обслуживания. Связь между обслуживающими приборами устанавливается с помощью системы очередей, выбранных дисциплин поступления и способов извлечения из них транзактов.

Пусть в нашем примере функциональные действия Φ_{D_1} , Φ_{D_2} , Φ_{D_3} совпадают. Кроме того, в любой данный момент выполняется только одно из них и исследователь интересует влияние этих Φ_D на поведение всей системы. Взаимосвязь между Φ_D , исследуемой системой можно представить в виде схемы. Программа-модель создается в два этапа. Вначале ИМ представляется в виде схемы, отображающей рождение транзактов, их пространственное перемещение по схеме и уничтожение уже обслуженных транзактов. Для описания ИМ создается достаточно широкий, но фиксированный набор стандартных блоков обслуживания транзактов. С их помощью представляются действия по образованию и исключению транзактов из модели, управлению движением транзактов, занятию и освобождению различных типов ресурсов системы, имитации задержек в продвижении транзактов. В общей сложности используется несколько десятков графически различных блоков, позволяющих в наглядной форме представлять алгоритмы функционирования сложных систем с дискретными событиями. На втором этапе осуществляется символическое кодирование схемы, при котором каждому блоку ставится в соответствие определенный оператор соответствующего языка моделирования образуя текст программы-модели.

Программа-модель в этом случае реализуется в режиме интерпретации. На рис. 14 изображена схема взаимодействия УПМ с моделью объекта при транзактном способе имитации реальной системы, представленной на рис. 11.

В источниках (*ИСТ*) транзакты создаются, а в блоке ПОГЛОТИТЕЛЬ все добравшиеся до него транзакты уничтожаются. УПМ просматривает списки источников транзактов (сканирует) и создает новые транзакты с заданной интенсивностью их поступления в системе. Таким образом, все поступающие в систему транзакты имитируют собой внешнее окружение реальной системы. Событием в ИМ, построенной по транзактному способу имитации, является момент инициализации любого транзакта. В результате транзакт в такой ИМ выступает в роли активности. УПМ назначает транзактам новые моменты инициализации t_i и до наступления этих моментов закрепляет за транзактами соответствующие блоки обслуживания ($БЛ_i$). Алгоритм работы УПМ состоит в следующем. ПП сканирования источников просматривает условия создания новых транзактов каждым из источников и передает управление ПП создания транзактов при совпадении этих условий. ПП создания транзактов формирует новые транзакты и помещает их в очередь к блокам, в нашем примере к $БЛ_{11}$, $БЛ_{12}$ и $БЛ_{31}$. После создания всех транзактов, для которых выполнены условия их создания УПМ приступает к просмотру поглотителей. ПП сканирования поглотителей просматривает очереди транзактов, завершивших свое пребывание в модели системы, и передает управление ПП уничтожения транзактов. Последняя при каждом уничтожении транзакта формирует статистики пребывания транзакта в модели системы. Когда просмотрены все поглотители ИМ, управление передается ПП формирования списка инициализируемых транзактов.

Под инициализацией транзакта понимают завершение пребывания транзакта в каком-либо блоке модели или поступление транзакта в соответствующие очереди к блокам модели системы. ПП формирования списка инициализируемых транзактов просматривает списки транзактов (очереди), поступивших на входы к блокам $БЛ_{ij}$ и выбирает из них те, у которых время инициализации совпадает с t_0 , образуя список активизируемых транзактов. По окончании просмотра всех блоков УПМ проверяет условие «Список инициализируемых транзактов пуст». Если в этом списке имеется хотя бы один транзакт, управление передается на ПП завершения обслуживания транзактов. Эта подпрограмма выбирает адрес блока, на котором завершено обслуживание транзакта, определяет имя очереди ($ОЧ_i$) к блоку $БЛ_{ip}$, завершившему обслуживание, выбирает из очереди $ОЧ_i$ новый транзакт и закрепляет за ним освободившийся транзакт.



При этом формируется момент будущей инициализации транзакта $t_0 + t_0$ (здесь t_0 означает длительность выполнения ФД₀, которое имитируется обслуживанием транзактов блоком БЛ₀, а t_0 - модельное время). Затем управление передается на ПП начала обслуживания транзактов. Эта подпрограмма определяет адрес блока и имя очереди, на которую должен поступить только что обслуженный транзакт, и организует его перемещение по схеме модели. Если блок, на который поступает транзакт, свободен, то он закрепляется за данным транзактом на время, указанное в описании этого БЛ₀. При этом выполняются те же действия по формированию момента t_0 инициализации транзакта, что и в ПП завершения обслуживания транзактов.

Как только список активизируемых транзактов оказывается пустым, ПП сдвига модельного времени выбирает из списка моментов инициализации транзактов $\{t_i\}$ минимальное значение, которое становится новым модельным временем t_0 . Просматривается все те транзакты в этом списке, у которых моменты инициализации совпадают, и формируется начальный список инициализируемых транзактов. Затем проверяется выполнение условий окончания имитации. Если условия не выполнены, то управление передается на ПП сканирования источников. При выполнении условий создания транзактов дополняется список инициализируемых транзактов, поскольку считается, что в моменты создания транзактов они также инициализируются, и алгоритм имитации УПМ повторяется. При выполнении условий окончания моделирования процесс имитации завершается и управление передается ПП окончания моделирования, которая выдает пользователю результаты моделирования. Таким образом, УПМ циклически сканирует различные списки транзактов с целью инициализации и организации обслуживания транзактов соответствующими блоками, сдвигает модельное время и проверяет выполнимость условий окончания имитации.

Как видно из алгоритма функционирования УПМ, за внешнею простоту описания ИМ приходится платить достаточно большими накладными расходами на организацию квазипараллелизма. Эти дополнительные затраты машинного времени t , значительно возрастают при увеличении самой модели, поскольку возрастает количество списков транзактов и блоков, которые необходимо несколько раз просматривать при организации квазипараллельной работы транзактов с помощью УПМ.

б) агрегатный способ.

В некоторых реальных системах имеет место тесное взаимодействие между функциональными действиями компонент системы. Компоненты системы обмениваются между собой сигналами. Каждый выходной сигнал от одной компоненты К, является входным сигналом для другой компоненты системы. Если при этом сами функциональные действия ФД₀ аппроксимируются явно задаваемыми математическими зависимостями, позво-

лиющими определять момент появления выходных сигналов компонент К, при наличии входных сигналов, поступающих от других компонент, то создаются подходящие условия для построения ИМ по модульному принципу. В этом случае каждый из модулей ИМ строится по унифицированной структуре и называется агрегатом.

Агрегат является математической схемой, с помощью которой возможно описание достаточно большого круга реальных процессов. В любой момент t_i агрегат может находиться в одном из возможных состояний. Состояния агрегата $z(t)$ называют фазовыми траекториями. Переход агрегата из состояния в состояние описывается с помощью некоторого оператора перехода H , который позволяет по предыдущему определить очередное состояние агрегата. Агрегат имеет входы, куда поступают входные сигналы X_i (от других агрегатов) и выходы, на которых формируются выходные сигналы $Y_j(t)$. Здесь i и j обозначают номера соответственно входных и выходных сигналов. Кроме того, у агрегата имеются дополнительные входы, на которые поступают управляющие сигналы $g(t)$. Выходные сигналы $Y_j(t)$ формируются из выходных $X_i(t)$ и управляющих $g(t)$ сигналов оператором выхода H в результате его взаимодействия с оператором H . Значения операторов H и G задаются исследователем при аппроксимации агрегатами выполнения ФД₀ реальной системы.

Заметим, что квазипараллельная работа агрегатных систем может быть реализована различными способами (активностями, планирование событий, взаимодействием транзактов, процессами). Поэтому выделение агрегатного способа организации квазипараллелизма в ИМ является условным и обуславливается наличием ряда специфических функций, которые должны выполнять УПМ для реализации такой универсальной математической модели. Среди этих функций необходимо выделить проверку условий перехода агрегата в особые состояния и формирование выходных сигналов, изменение координат состояний агрегата, передачу сигналов, изменение координат состояний агрегата, передачу сигналов от одного агрегата к другому. Моделирование поведения агрегата представляет собой последовательную цепь переходов из одного особого состояния в другое. Когда законы поступления управляющих и выходных сигналов заданы, то в функции УПМ входит проверка условий перехода агрегата в особые состояния и возможность выработки выходных сигналов. Если входные сигналы не могут быть определены до момента их имитации, УПМ предусматривает процедуры проверки факта поступления внешних сигналов за некоторый интервал модельного времени Δt_0 . УПМ анализирует возможность моделирования выходных сигналов у агрегата.

Частным случаем агрегата является кусочно-линейный агрегат (КЛА), у которого отсутствуют управляющие сигналы. Обычно КЛА представляется в виде многополюсника, у которого имеются N входных и m выходных полюсов. На входные полюсы поступают элементарные сигналы $X_i (i = \overline{1, N})$.

Аналогичную структуру имеют выходные сигналы $Y_r (r = \overline{1, m})$. Связи между выходами одних агрегатов и входами других называются каналами. Существует следующее ограничение: по каждому каналу может передаваться только один сигнал, который является входным для одного агрегата и выходным для другого. Ниже рассматриваются особенности организации квазипараллельного обслуживания агрегатов УПМ для случая, когда каждое функциональное действие компонент реальной системы ФД₀ аппроксимируется в ИМ отдельным КЛА. Конечно, привязка отдельных ФД₀ к различным агрегатам не всегда целесообразна и рассматривается нами только для простоты изложения специфики организации УПМ квазипараллельной работы агрегатов.

На рис. 15 представлена схема взаимодействия УПМ с моделью объекта, состоящей из КЛА. Каждый КЛА аппроксимирует соответствующее ФД₀ реальной системы, представленной на рис. 11. Агрегатная система является замкнутой, и возникновение выходных сигналов в агрегате A_0 можно лишь вследствие возникновения внутренних событий. Поэтому A_0 представляет собой модель поведения внешней среды, которая воздействует на исследуемую систему. Каждому ФД₀ в ИМ соответствует свой агрегат A_0 . Процесс эволюции состояний в агрегативной ИМ и появление C_n обеспечивает УПМ. Допустим, на выходе A_{12} появляется выходной сигнал Y_{32} . Производится пассивизация агрегата A_{32} , назначение момента будущей активизации агрегата A_{32} и формирование списка выходных сигналов. Под пассивацией агрегата будем понимать занесение его в список агрегатов, ожидающих выполнения либо временных, либо логических условий, согласно которым УПМ возвращается к обслуживанию агрегата назовем активизацией агрегата. Момент возврата УПМ к обслуживанию агрегата назовем активизацией агрегата. Далее проверяется наличие у агрегата A_{32} выходных сигналов.

Для того чтобы УПМ могла организовать взаимосвязь между агрегатами с помощью сигналов, пользователь при формировании реальной системы составляет матрицу коммутации, указывающую коммутацию агрегатов соответствующими каналами. При наличии выходного сигнала Y_{33} по матрице коммутации агрегатов УПМ определяет имя входного сигнала (в данном случае это будет X_{33}) и организует активизацию агрегата A_3 .

Когда все выходные сигналы Y_i агрегатов A_0 обслужлены, проверяется условие окончания моделирования и при его невыполнении управление передается на подпрограмму активизации агрегатов по времени имитации. Это означает выбор минимального значения в упорядоченном списке моментов времени изменения состояний агрегатов t_{ij} .

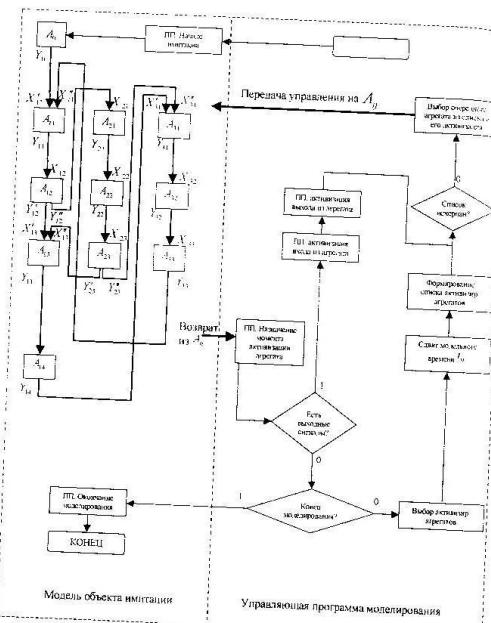


Рис. 15 - Блок схема организации агрегатного способа имитации СС

Для простоты предполагаем, что все A_j обладают только одной координатой изменения состояний и оператор переходов соответствует простейшему алгоритму вычисления момента $t_{ij}=t_0+t_{ij}$ появления следующего события C_{ij} . При выборе минимального значения t_0 определяется точка входа (адрес) или имя того агрегата, который в данный момент должен быть активирован. Это имя агрегата становится начальным значением

списка активизируемых по времени агрегатов. Одновременно с этим происходит сдвиг модельного времени ($t_0=\min t_{ij}$). Затем УПМ формирует список активизируемых агрегатов, т.е. тех агрегатов у которых момент активизации совпадает с новым значением t_0 . По окончании формирования списка активизируемых агрегатов УПМ приступает к последовательной активизации алгоритма тех A_{ij} , которые попали в этот список. Активизация любого A_{ij} состоит в передаче управления на A_{ij} и реализации последовательных операторов H и G , в результате чего агрегаты переходят в новые состояния $Z(t)$ и формируются выходные сигналы. Далее процесс обслуживания УПМ агрегатов аналогичен ранее рассмотренному.

Итак, обобщая рассмотренный пример, можно выделить пять шагов эволюции агрегативной системы, состоящей из k агрегатов и УПМ:

- начальная установка состояний агрегатов;
- определение момента времени, в который произойдет следующее событие в агрегатах, и номера агрегата, в котором произойдет это событие;
- определение нового состояния агрегата, формирование множества выходных сигналов в результате сопряжения внутренних событий и фиксация новых состояний всех агрегатов;
- проверка множества выходных сигналов и, если необходимо, по соответствующей матрице коммутаций определение номеров каналов, по которым должен быть передан выходной сигнал, при этом определяется также номер входного полюса, на который поступает входной сигнал;
- определение нового состояния агрегата в результате прихода входного сигнала и формирование выходного сигнала.

Описание поведения каждого агрегата реализуется с помощью соответствующего языка моделирования. При этом последовательно указывает способ выдачи и характеристики операторов переходов и операторов выдачи выходных сигналов G , составляет матрица коммутации полюсов агрегатов. Далее система моделирования, реализующая соответствующий язык моделирования, обеспечивает перевод описаний в программные модули, а УПМ уже стандартным образом организует собственно имитацию событий в агрегативной системе, сбор статистики и управление ходом имитации. По выполнении условий окончания моделирования управление передается подпрограмме окончания моделирования, которая и выдает исследователю результаты моделирования в принятой для данной системы моделирования стандартной форме.

Как видно из приведенного примера, агрегативный способ организации квазипараллелизма в ИМ является достаточно удобным с точки зрения описания сложной системы. Однако необходимость коммутации и обслуживания сигналов требует дополнительных расходов ресурсов машинного времени t_0 . За удобство описания и математическую строгость модели приходится расплачиваться дополнительным расходом t_0 , и это обстоятельство является одним из факторов, сдерживающих использование агрегат-

тивного способа организации квазипараллелизма при моделировании сложных систем.

в) процессный способ.

При моделировании сложных систем исследователи зачастую встречаются со следующими ограничениями. Все функциональные действия ФД компонент реальной системы различны. Условия появления событий C_i приводящие к выполнению ΦD_i , также индивидуальны. У каждой компоненты К_i существует определенная последовательность выполнения ΦD_i . В любой момент времени в данной компоненте может выполняться только одно ΦD_i . В любой момент времени в данной компоненте может выполняться только одно ΦD_i . Перечисленные ограничения определяют выбор исследователем процессного способа организации квазипараллелизма в ИМ. При процессном подходе краткость описания активностей объединяется с эффективностью событийного представления имитации.

Процессный способом можно организовать имитацию любых сложных систем. Однако процессный способ имитации особенно эффективен в тех случаях, когда требуется высокий уровень детализации выполнения ΦD_i при аппроксимации с помощью АЛ_и и сама ИМ используется для поиска узких мест в системе. При такой постановке задачи на моделирование очень важно соблюдать сходства структуры модели и объекта исследования, что обеспечивается процессным способом имитации. В нашем примере (см. рис. 11) реальную систему удобнее всего имитировать процессным способом, когда в один и тот же момент времени не может реализоваться более одного ΦD_i данной i-й компоненты реальной системы. Тогда исследователю удобнее рассматривать функционирование компонент модели К_i как единое целое. Вся имитационная модель можно представить в виде набора описаний процессов, каждое из которых описывает один класс процессов, например компоненты К_i для нашего случая. При этом могут иметь место информационные и управляющие связи не только между компонентами К_i, но даже и между отдельными алгоритмами их функционирования.

На рис.16 представлена схема взаимодействия УПМ с моделью объекта, состоящей из описаний процессов. Алгоритм функционирования ИМ представляется последовательным взаимодействием процессов и УПМ. Причем в процессы обединяются связанные между собой активности, которые определяют функционирование одной и той же компоненты модели К_i. Таким образом, имеет место полное соответствие компонент реальной системы и ее ИМ. Каждой компоненте объекта моделирования соответствует свой процесс. Переход от выполнения одной активности к другой активности того же процесса считают изменением его состояния и называют активизацией процесса. Обычно под состоянием процесса понимают номер (j) той активности, которая входит в состав i-го процесса и на которую

УПМ передает управление при свершении события C_j в К_i. Следовательно, изменения состояний реальной системы соответствуют изменениям соответствующих состояний процессов и появлению событий C_j .

Вся ИМ представляет собой набор процессов, реализованных на соответствующем языке моделирования. Процессы связаны с УПМ с помощью некоторых операторов этого языка, по которым имеет место обращение к УПМ при завершении активности данного процесса, означающее переход процесса в другое состояние. Отметим при этом, что проверка выполнимости условий активизации процесса и появление событий C_j осуществляются самим процессом. Будучи активизированным, выполнение процесса (некоторой его активности) может начаться немедленно или задержаться до появления определенных условий или до изменения состояний других процессов. Процессы могут переходить в новые состояния как по своей инициативе, так и в результате действий, выполняемых активностями других процессов.

В качестве примера реализации квазипараллелизма на основе процессного способа выберем алгоритм функционирования УПМ, реализованный в моделирующем комплексе (МК) PLSIM. Для простоты изложения опустим некоторые детали функционирования этого алгоритма и будем считать, что взаимодействия между активностями, из которых состоят процессы, нет и все активности обращаются к УПМ с помощью операторов синхронизации WAIT (t_{ij}). Каждый такой оператор означает, что данному i-му процессу по окончанию выполнения алгоритма АЛ_j активности АК_{ij} назначается момент следующей активизации t_{ij} по окончании ожидания процесса в модельном времени длительностью t_j . Началу выполнения АЛ_j процессов соответствуют адреса a_j в подпрограммах, реализующих выполнение активностей, объединенных в процессы. УПМ работает с массивом состояний процессов (МС) и таблицей состояний процессов (ТС). В МС каждый элемент представляет собой пару значений (i, t_j), где i – номер процесса, t_j – момент j-й активизации процесса в будущем и появления события C_j . Для выбора процессов, которые необходимо активизировать в момент t_0 , УПМ использует ТС. Строками этой таблицы являются списки параметров процессов.

Так, i-му процессу соответствует строка, в которой указаны следующие параметры: a_j – адрес передачи управления на выполнение j-й активности в i-м процессе; t_{0ji} – время, которое осталось i-му процессу находиться в состоянии ожидания, когда он останавливается другим процессом; π – приоритет i-го процесса, согласно которому осуществляется последовательное обслуживание УПМ двух одновременно активизируемых процессов.

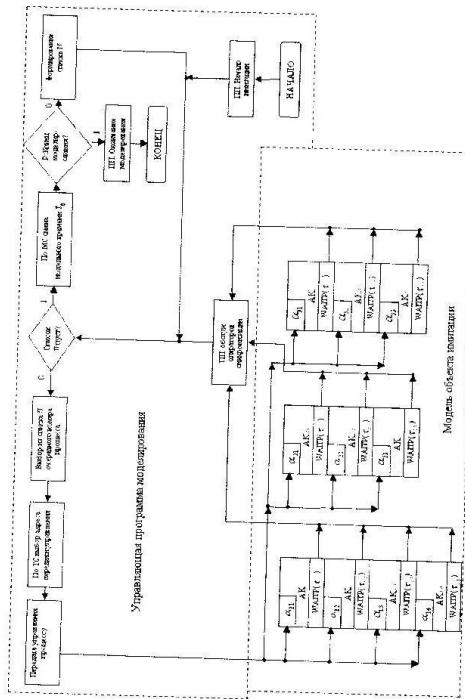


Рис. 16 - Блок схема организации процессного способа имитации СС

Все элементы МС упорядочены по возрастанию значения t_{ii} . Если какой-либо i -й процесс остановлен (хотя бы даже для обслуживания его **УПМ** в моменты активизации), то время последующей его активизации в МС бесконечно велико (t_{ii}^∞) и соответствующий ему элемент находится в конце этого массива.

Определяем понятие «конфликтная ситуация в системе дискретных событий»: несколько событий C_i в различных процессах происходят одновременно и требуют немедленного обслуживания со стороны УПМ. Такая ситуация имеет место, например, при наличии в МС нескольких элементов с одним и тем же значением t_{ij} . Учет конфликтных ситуаций производится с помощью списка одновременно активизируемых процессов (список jj). В списке jj находятся номера всех процессов, которые необходимо активизировать в один и то же модельное время t_b . Запись номеров процессов i в списке jj и выбор их из этого списка УПМ осуществляется согласно приоритетам процессов π_i . Обслуживание УПМ очередного элемента списка jj состоит в передаче управления на выполнение алгоритма АЛ _{i} -го процесса. Выполнение этого алгоритма продолжается до появления в нем очередного оператора синхронизации процесса WAIT (t_{ij}). С помощью операторов ожидания типа WAIT (t_b) имитируется изменение временных координат активностей, входящих в состав i -го процесса.

В нашем примере (см. рис.11) реальный объект состоит из трех компонент, и поэтому его ИМ состоит также из трех процессов ($i=1, 2, 3$). Время задано время окончания моделирования $T_{\text{ок}}$, а условием окончания моделирования является достижение модельным временем t_0 значения $T_{\text{ок}}$. УПМ обслугивает процессы согласно алгоритму, представленному на рис. 16. Пусть в начальный момент моделирования $t_0=t_0^{\text{ст}}$ список активизации процесса jj пуст и процессам назначены соответственно моменты активизации t_{1j}, t_{2j}, t_{3j} в МС.

Шаг 1. Поскольку список j пуст, то осуществляется процедура смены t_0 , представляющая собой выбор минимального значения из МС ($t_0 = \min\{t_{ij}\}$).

Шаг 2. Формируется список JJ следующим образом. Все процессы, у которых $t_i \leq t_0$, выбираются из МС и список JJ заносится согласно приоритетам процессов τ . Приоритеты процессов определяются по таблице ТС.

тетом процессов π . Приоритеты процессов определяются по таблице 3.

Шаг 3. Выбирается первый элемент списка j , который необходимо в момент $t_0 = 0$ активизировать. По ТС определяется адрес a_{ij} и управление от УПМ передается на выполнение алгоритма соответствующей активности. По окончании выполнения алгоритма активности появляется оператор ожидания i-го процесса до следующей активизации WAIT (t_1). В ходе реализации алгоритма активности A_i , происходит вычисление или задание значения t_1 для операторов синхронизации процессов. Появления в алгоритме i-го процесса операторов синхронизации WAIT (t_1) возвращают управление УПМ, которая выполняет в дальнейшем следующие действия:

- формирование нового элемента МС и занесение его в этот массив согласно значению $t_0=t_0+t_{ij}$;
- модификацию текущего состояния процесса и изменение при этом адреса последующей передачи управления a_{ij} для очередной активизации процесса i в ТС.

Шаг 4. Проверяется, исчерпан ли список jj . Если в списке jj есть еще элементы, то УПМ переходит к шагу 2. Иначе выполняется шаг 5.

Шаг 5. Проверяется момент окончания моделирования ($t_0 \geq T_{ok}$). Если это неравенство не выполняется, то это означает необходимость продолжения имитации, начиная с шага 1. В противном случае управление передается программе окончания моделирования.

Продолжим рассмотрение приведенного примера, используя временную диаграмму перехода компонент реальной системы (процессов модели системы) из состояния в состояние (см. рис. 9) и схему взаимодействия УПМ с процессами (см. рис.16). Итак, первоначально в момент $t_0=m_0$ в списке jj находятся все три процесса, которые расположены соответственно их приоритетам, например $\pi_1 > \pi_2 > \pi_3$. Допустим, что время окончания моделирования T_{ok} равно моменту m_8 на оси изменения модельного времени t_0 (см. рис. 9).

Управляющая программа моделирования вначале передает управление по адресу a_{11} . Выполнение алгоритма активности AK_{11} завершается появлением оператора WAIT (t_{11}), и управление передается на ПП обслуживания операторов синхронизации. Эта подпрограмма выполняет следующие действия:

- формирует новый адрес продолжения процесса 1 в ТС (a_{12}); назначает момент будущей активизации процесса 1 ($t_{11}=m_7+t_{12}$);
- формирует новую строку МС (1, t_{11}) и заносит ее в этот массив.

Далее по алгоритму УПМ (поскольку jj не пуст) происходит выбор процесса $j=2$, и управление передается по адресу a_{21} для выполнения алгоритма активности 21 до появления оператора WAIT (t_{21}) и возврата управления на ПП обслуживания операторов синхронизации. Формируется новый момент активизации процесса 2 ($t_{21}=m_6+t_{22}$). В ТС модифицируется адрес продолжения выполнения процесса 2 (a_{22}) и формируется новая строка МС (2, t_{21}), которая заносится в соответствии со значением t_{21} (после строки для процесса 1) в этот массив. Аналогично обслугивается процесс $i=3$ и управление передается по адресу a_{31} для выполнения алгоритма активности 31 до появления оператора WAIT (t_{31}) и возврата на ПП обслуживания операторов синхронизации. Формируется новый момент активизации процесса 3 ($t_{31}=m_0+t_{32}$). В ТС модифицируется адрес продолжения выполнения процесса (3 a_{32}) и формируется новая строка МС (3, t_{31}), которая в соответствии со значением t_{31} заносится в этот массив между строками для первого и второго процессов (см. рис. 9).

Когда список jj исчерпан, происходит смена модельного времени. Новым значением модельного времени становится минимальное значение

$$t_0=\min(t_{11}, t_{21}, t_{31})-t_{11}-m_1. \quad (4.2)$$

При этом $i=1$ заносится в список jj , и поскольку $t_{11} < m_8$, то управление передается на формирование списка jj . Так как момент $t_0=m_1$ активизируется только один процесс, то в списке jj будет находиться всего один процесс. Согласно алгоритму УПМ управление передается по адресу a_{12} на выполнение алгоритма активности 12 до появления оператора WAIT (t_{12}). Процессу 1 назначается новый момент активизации $t_{12}=t_0+t_{12}-m_0+t_{12}$, и в ТС модифицируется адрес последующей активизации процесса a_{13} . В соответствии со значением t_{12} строка (1, t_{12}) заносится в МС после строк для процессов 2 и 3. Далее аналогичным образом продолжаются обслуживание УПМ процессов и смена модельного времени согласно временной диаграмме рис. 9 до момента, когда модельное время $t_0=m_8$. В этот момент по окончании обслуживания процессов 1 и 2 выполняется условие $t_0 \geq T_{ok}$ и процесс моделирования завершается, а управление передается на программу окончания моделирования.

При этом способе имитации близость модели к отображаемой системе облегчает обозрение принятых предположений и упрощений. Это необходимо в задачах моделирования проектируемых систем, когда в ходе проектирования или исследования реальной системы модель приходится вносить частные изменения. Процессный способ имитации обладает хорошими изобразительными возможностями при осуществлении многоуровневого модульного подхода к моделированию систем. Эти преимущества возрастают по мере роста размеров модели.

г) транзактно-процессный способ.

При моделировании сложных систем исследователям приходится учитьывать особенности системы. Такими особенностями могут быть: все функциональные действия различны, но они принадлежат к разным группам Φ_{D_i} , которые можно объединить по принадлежности к компонентам K_i . У каждой компоненты K_i только одно из Φ_{D_i} может выполняться в конкретный момент времени. Причем, имеет место последовательный характер выполнения Φ_{D_i} друг за другом. Поэтому можно считать, что факт выполнения Φ_{D_i} зависит от состояния S_j компоненты K_i . Кроме того, алгоритмы выполнения Φ_{D_i} состоят в выполнении некоторого набора запросов на ресурсы моделируемого объекта. Алгоритмы предоставления алгоритмом Φ_{D_i} ресурсов системы на каждый к-й запрос. Это означает, что алгоритмы Φ_{D_i} могут обслуживать группу этих запросов в соответствии с приоритетом этих запросов π_k и дисциплиной выбора этих запросов из группы. В таких случаях запросы ресурсов Z_k удобно представлять в виде динамиче-

ских элементов ИМ, которые назовем транзактами (TR_k). Сами по себе Z_k ских элементов k -го типа к определенной группе обслуживаемой в соответствии с их приоритетом π_k : величина запроса ресурсов V_{rk} , представляющая собой случайную величину, номер Φ_D , алгоритм выполнения которого имитирует выделение ресурса, указатель места накопления статистики обслуживания динамических элементов модели. Переисчисленные особенности статистических и динамических элементов ИМ определяют выбор исследователем транзактно-процессного способа организации квазипараллелизма в ИМ. При этом все Φ_D , относящиеся к компоненте K , объединяются в один статистический элемент модели, называемый в дальнейшем процессом. Номера j Φ_D , относятся к одному и тому же процессу номера i . Сами Z_k , являясь динамическими элементами, представлены транзактами TR_k сложной структуры. TR_k в отличие от традиционного понимания обладает: идентификатором k , отображающим принадлежность к определенной группе запросов Z_k ; приоритетом π_k , величиной запроса ресурса r -го типа V_{rk} ; номер состояния процесса j , определяющий номер Φ_D , которое обеспечивает выделение ресурса r -го типа; номер процесса j , который обслуживает TR_k после процесса j (номер процесса-предшественника обработки TR_k).

Как видим, транзактно-процессным способом можно организовать имитацию сложных систем, которые требуют на высоком уровне детализации обслуживания TR_k сложной структуры отдельными Φ_D . При этом имеет место сходство структуры модели и объекта исследования. Удобство рассмотрения K как единого целого, состоящее из нескольких Φ_D и объединенных в единое целое, называемые процессом. Процессы обслуживаются динамическими элементами (TR_k) и для синхронизации различных процессов на входе процессов используются очереди TR_k .

Все ИМ системы можно представить состоящей из четырех частей: группа моделей динамических элементов транзактов $\{\Delta EM_k\}$; группа моделей статистических элементов-процессов $\{CTEM\}$; база данных линейных элементов ($BDDEM$); база данных статистических элементов ($BDCSTEM$). При этом существуют управляющие и информационные связи между ΔEM_k и $CTEM$. Синхронизация взаимодействия процессов друг с другом и транзактами сложной структуры осуществляется через систему очередей OQ_i . На входе ИМ имеется множество генераторов, создающих объекты исследования. В OQ_i TR_k могут поступать от $CTEM_i$ и обслуживаются $CTEM_{i+1}$.

Группа $\{\Delta EM_k\}$ описывается алгоритмами генераторов транзактов, начинающимися по адресу ρ_k и заканчивающиеся операторами ожидания $WAIT$ Δ . Множество массивов переменных линейных элементов $\{\Delta EM_k\}$, расположенные в базе данных линейных элементов

$BDDEM$ по адресам S_{ik} , содержит в себе следующую информацию: $F(V_{rk})$, j, j_1 (здесь $F(V_{rk})$ – функция распределения размеров запросов ресурсов r -го типа Z_k динамическим элементом). Кроме информационной части транзакт сложной структуры может выполнять алгоритмическую обработку информации, находящейся в массивах переменных $MDEM_k$. Это обстоятельство означает, что динамические элементы ΔEM_k мало отличаются от статистических элементов $CTEM$. Но модели от одной очереди OQ_i к другой очереди OQ_j , движется триада транзакта: идентификатор (k), приоритет π_k , адрес «тела» транзакта (S_{ik}). При традиционной структуре модель динамического элемента (TR_k) не имеет «тела» транзакта и поэтому $S_{ik} = 0$. Статистические элементы ИМ $CTEM$, могут содержать несколько моделей Φ_D (активности AK_{ip}). Алгоритм каждой AK_{ip} начинающийся адресом a_{ij} и завершившийся оператором синхронизации $WAIT_{Cj}$, обозначим как набор алгоритмических операций $CTEM_i$, которые используют содержащее массивов переменных $MCTEM_i$ и формируют набор статистик имитации в виде множества массивов статистик выполнения $CTEM_i$ ($STST_i$). На рис. 17 приведена блок-схема организации транзактно-процессного способа имитации.

Как видно из рис. 17, модель объекта имитации состоит из двух групп: элементов модели $\{\Delta EM_k\}$ и $\{CTEM\}$, каждая из которых использует собственную базу данных моделирования системы: $\{\Delta DEM_k\}$, $\{STD_E_i\}$, $\{MCTDEM_i\}$, $\{STST_i\}$. Вторую часть ИМ объекта (см. рис. 17) представляет стандартная управляющая программа моделирования (УМП). Алгоритм функционирования ИМ ДСС представляется последовательным взаимодействием ΔEM , $CTEM$ и УМП. Процессы объединяются в связанные между собой активными ΔEM_k и $CTEM$, которые определяют функционирование одной и той же компоненты модели K и обслуживание множества динамических элементов (транзактов сложной структуры). Переход к выполнению активностей ΔEM и $CTEM$ считается состоянием соответствующими динамическими и статистическими элементами модели и означает активизацией соответствующего транзакта TR_k или процесса j . Под состоянием транзакта или процесса понимают номер (k) или процесса (j) на которые УМП передает управление при сворачивании событий C_i и K_j . Таким образом, изменением состояний реальной ДСС соответствует изменениям соответствующих состояний транзактов и процессов и появления соответствующих событий C_i и C_j .

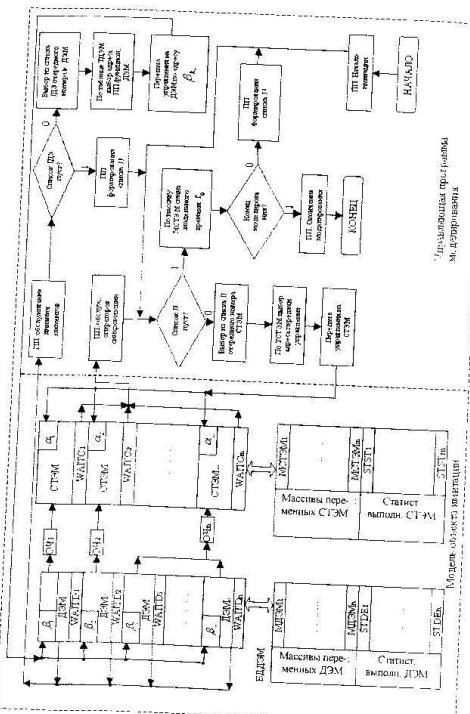


Рис. 17 - Блок схема организации транзактно-процессного способа имитации СС

98

Процессы и транзакты связаны с УПМ с помощью соответствующих операторов WAITD_i и WAITC_j, по которым имеет место обращение у УПМ. Проверка выполнимости условий активизации осуществляется самими транзактами и процессами. Все переходы от программ ДЭМ, на УПМ осуществляются с помощью программы обслуживания динамических элементов.

Момент активации ДЭМ_k определяется по формуле:

$$t_{\text{ДЭМ}k} = t_0 + t_k \quad (4.3)$$

где t_0 – модельное время; t_k – время выполнения алгоритма k -го динамического элемента.

Все моменты активации ДЭМ_k упорядочены в списке моментов активации СМАД по возрастанию $t_{\text{ДЭМ}}$. Аналогичным образом все переходы от активностей процессов (СТЭМ) на УПМ осуществляются с помощью программы обслуживания операторов синхронизации СТЭМ. Момент активации СТЭМ определяется по формуле:

$$t_{\text{СТЭМ}} = t_0 + t_{ij} \quad (4.4)$$

где t_{ij} – время выполнения алгоритма ij -ой активности.

Все моменты активации СТЭМ, упорядочены в списке моментов активации МСТЭМ по возрастанию $t_{\text{СТЭМ}}$.

Блок-схема алгоритма выбора УПМ номера активизируемого динамического и статистического элементов представлена на рис. 7.10.

Выбор очередного активизируемого ДЭМ_k осуществляется следующим образом. После установки в массиве СМАД момента $t_{\text{ДЭМ}k}$ проверяется условие: список IDЭ пуст? Таким способом УПМ пытается завершить запуск тех ДЭМ_k, которые ожидали своей активизации. Если список IDЭ не пустой, то выполняется следующая последовательность действий: выбор из списка IDЭ отредного номера (k_0) ДЭМ. По таблице состояний динамических элементов (ДЭМ) определяется адрес β_0 активности ДЭМ_{k0} и затем управление передается по адресу β_0 на выполнение активности ДЭМ_{k0}.

По окончании выполнения активности ДЭМ_{k0} с помощью оператора синхронизации WAITD_{k0} управление возвращается на ПП обслуживания динамических элементов для назначения нового момента активизации активности ДЭМ_{k0}. Если же предыдущая активизация ДЭМ_k завершила, то это означает, что список IDЭ пустой. В этом случае формируется новый список IDЭ только тех ДЭМ_k у которых

$$t_{\text{ДЭМ}k} < \min t_{\text{СТЭМ}} \quad (4.5)$$

Поэтому управление в УПМ передается на оператор проверки пустой ли список процессов (список JJ).

99

Если список JJ не пустой, то выполняется следующая последовательность действий: выбирается из списка JJ очередного номера процесса j ; по номеру j из таблицы статических элементов (СТЭМ) определяется адрес активности $\{A_j\}$, которая должна быть активизирована именно в момент модельного времени t_0 ; передается управление по адресу A_j на выполнение активности СТЭМ $_j$. По окончании алгоритма выполнения активности СТЭМ $_j$ с помощью оператора синхронизации WAIT $_j$ управление возвращается на ПП обсуждения оператором синхронизации СТЭМ, которая назначает новый момент активизации данной активности по формуле (7.4). Снова проверяется условие: Список JJ пустой? Если все активности СТЭМ $_j$ активизированы в момент модельного времени t_0 (список JJ пустой), то по массиву МСТЭМ осуществляется смена модельного времени t_0 . Проверяется условие окончания имитации. В случае необходимости продолжения имитации управление передается на ПП формирования списка JJ . В списке JJ формируются номера тех активностей СТЭМ $_j$, которые необходимо активизировать в новый момент модельного времени t_0 . По окончании формирования списка JJ управление передается на проверку содержимого списка JJ . Если же выполняются условия окончания моделирования, то управление передается на программу окончания моделирования. В самом начале имитации с помощью ПП «начало имитации» устанавливаются начальные состояния таблиц УПМ и управление передается на проверку «Пуст ли список JJ ?». Далее алгоритм УПМ выполняется аналогичным образом.

а) агрегатно-процессный способ.

В ряде случаев СС может состоять из компонентов трех типов: одна группа компонент $\{K_{ij}\}$ имеет типовой состав и алгоритмы выполнения ФД $_{ij}$; компоненты K_{ij} другой группы имеют уникальный состав ФД $_{ij}$; третья группа компонент $\{K_{ij}\}$ также имеет типовой состав, в их функции входит наблюдение за завершением функционированием $\{K_{ij}\}$ и управлением выполнения следующих $\{K_{ij}\}$. Примером такого совместного выполнения компонент сложной системы являются технологические процессы производства (ТПП). Выполнение ТПП представляется последовательностью технологических операций $\{\text{TXO}_i\}$. Каждая TXO $_i$, в свою очередь, может состоять из последовательности взаимосвязанных друг с другом микротехнологических операций $\{\text{MTXO}_{ij}\}$. При этом одна группа МТХО $_{ij}$ не может начать выполняться до тех пор, пока не будет выполнена предшествующая группа МТХО $_{ij}$.

В таких ситуациях параллельно-последовательный характер выполнения $\{\text{MTXO}_{ij}\}$ удобно представлять в виде агрегатов. Причем, в отличие от классического представления кусочно-линейных агрегатов (КЛА), эти агрегаты могут иметь сложную структуру и состоять из нескольких взаимосвязанных активностей (АКА $_{ij}$). Каждая активность АКА $_{ij}$ представляет

собой модель ФД $_{ij}$. Внутри МТХО $_{ij}$ группа ФД $_{ij}$ может быть взаимосвязана и их поведение определяется алгоритмом МТХО $_{ij}$. Каждую МТХО $_{ij}$ в таких случаях можно представить агрегатами первого типа AGR $_{ij,1}$, которые являются моделями МТХО $_{ij}$ и могут состоять из одной или нескольких активностей АКА $_{ij}$. Начало выполнения одной группы $\{\text{MTXO}_{ij}\}$ зависит от завершения предшествующей группы $\{\text{MTXO}_{ij}\}$. Агрегаты AGR $_{ij,1}$ взаимодействуют друг с другом информационно с помощью множества сигналов. После завершения алгоритма выполнения AGR $_{ij,1}$ на его выходе формируется SIGN $_{ij}$ сигнал, который поступает на один из агрегатов второго типа AGR $_{ij,2}$. В функции AGR $_{ij,2}$ входит контроль выполнения предыдущих МТХО $_{ij}$ и запуска на выполнение алгоритмов последующих МТХО $_{ij}$. С приходом последнего сигнала от AGR $_{ij,1}$ на входы AGR $_{ij,2}$ срабатывает спусковая функция, которая формирует с выходов AGR $_{ij,2}$ выходные сигналы на входы агрегатов AGR $_{ij,1}$.

Число входных и выходных сигналов у AGR $_{ij}$ может быть различным. В таких случаях компоненты 3-го типа K $_{ij}$ реальной СС в ИМ представляются в виде агрегатов AGR $_{ij,3}$. Наконец, компоненты 2-го типа K $_{ij}$ можно представить в виде множества взаимосвязанных активностей и объединить их в единий процесс (PROC $_{ij}$) номера m . Связь между PROC $_{ij}$ в ИМ организуется с помощью команд запуска (PUSKC $_{ij,m}$) и остановки (STOPC $_{ij,m}$). Поэтому каждая активность АКР $_{km}$ процессов может оканчиваться либо командами PUSKC $_{ij,m}$ и STOPC $_{ij,m}$, либо оператором синхронизации процессов WAITP ($t_{ij,m}$). Аналогичным образом AGR $_{ij,3}$ состоят из набора активностей АКА $_{ij,3}$ (где k -номер активности в агрегате номера ij), которые могут оканчиваться: либо операторами формирования сигналов агрегата (SIGNA $_{ij,3}$), либо командами PUSKA $_{ij,3}$ и STOPA $_{ij,3}$, либо операторами синхронизации агрегатов WAITA ($t_{ij,3}$). Агрегаты AGR $_{ij,3}$ также состоят из набора активностей АКР $_{kj}$, которые могут оканчиваться: либо операторами формирования сигналов агрегата SIGNB $_{ij,3}$, либо командами PUSKB $_{ij,3}$ и STOPB $_{ij,3}$. Самы по себе эти сигналы могут иметь сложную структуру и быть «поликрашенными» (по аналогии с ДЭМ $_i$ при транзактно-процессном способе имитации). В «телах» сигнала может находиться самая разнообразная информация, по которой последующие агрегаты AGR $_{ij,1}$ информируются об операционной обстановке в ИМ объекта имитации (например, имели ли место отказы оборудования при выполнении предыдущего агрегата). В отличие от транзактно-процессного способа имитации, сигналы по ИМ передаются только от агрегата-источника до агрегата - адресата их движения и далее по модели не распространяются. Поэтому между агрегатами нет.

На рис. 18 приведена блок-схема организации агрегатно-процессного способа имитации СС. Как видно из рис. 18, модель объекта имитации состоит из трех групп элементов модели: агрегата универсальной структуры (AGR $_{ij}$), универсального агрегата (AGR $_{ij,3}$) и множества процессов (PROC $_{ij}$). Каждый элемент этих агрегатов и процессов состоит из своего набора активностей. Таблицы информационного взаимодействия агрегатов

тивностей. Таблицы информационного взаимодействия агрегатов первого типа и рабочие массивы агрегатов составляют информационную базу данных агрегатов первого типа (IBD₁). Агрегаты второго типа используют свои таблицы информационного взаимодействия, в которых рассматриваются приходы и отправления сигналов от агрегатов и для агрегатов первого типа. В совокупности эти таблицы и рабочие массивы составляют информационную базу данных второго типа (IBD₂). Процессы также используют таблицы и массивы их для информационной связи друг с другом, образуя в совокупности информационную базу данных третьего типа (IBD₃). Поскольку алгоритмы выполнение всех агрегатов первого типа одинаковы, то выполнение агрегатов номера (ij) первого типа представляется одной реинтегральной программой. Каждый номер AGR_{ij} имеет в IBD₁ свою, область что позволяет одной и той же реинтегральной программе обслуживать параллельным образом в модельном времени все множество агрегатов первого типа.

Аналогичным образом из-за типового характера алгоритмов второго типа, выполнение агрегатов AGK₂ также представляется одной реинтегральной программой. Здесь также каждый номер AGR_{2j} имеет в IBD₂ свою область, позволяя программе модели агрегата 2-го типа обслуживать параллельно в модельном времени все множество агрегатов 2-го типа.

Для организации квазипараллелизма выполнения агрегатов и процессов используется управляющая программа моделирования (УПМ). От ИМ объекта имитации на УПМ возможны следующие группы выходов:

- на выполнение сигналов агрегатов (по операторам SIGNA_{ij} и SINGNB_j) инициируется ПП обслуживания сигналов;
- на выполнение операторов управления агрегатами по операторам PUSKA_{ij} и STOPA_{ij}, инициирующим ПП обслуживания операторов пуска-останова агрегатов;
- на выполнение операторов управления процессами по операторам PUSCP_m и STOPC_m, инициирующий ПП обслуживания операторов пуска-останова процессов;
- на выполнение операторов синхронизации агрегатов WAITA (t_0, ij) и процессов WAITP (t_0, m), инициирующими ПП обслуживания операторов синхронизации агрегатов и процессов.

Отметим, что три подпрограммы (ПП обслуживания сигналов агрегатов, ПП обслуживания операторов пуска-останова агрегатов, ПП обслуживания операторов пуска-останова процессов) представляют собой обычные процедуры, обеспечивающие обратный возврат от УПМ на следующую активность агрегата или процесса. Это обстоятельство определяет динамику взаимодействия УПМ и модели объекта имитации мгновенно в модельном времени. Поэтому выбор для активизации очередной активности агрегаты первого типа или процесса осуществляется УПМ после возврата

управления от модели объекта на ПП обслуживания операторов синхронизации.

Эта подпрограмма либо выполняет оператор WAITA (t₀, ij), назначая момент активизации активности агрегата, либо выполняет оператор WAITP (t₀, m), назначая момент активизации процедуры номера m. В этих операторах указывается (ij) номер агрегата первого типа (AGR_{ij}), или же номер (m) процесса PROC_m.

Назначение момента активизации агрегата ij или процесса m осуществляется по формуле:

$$t_{AKA2} = t_0 + \tau_{ij} \quad \text{и} \quad t_{AKPm} = t_0 + \tau'_m. \quad (4.5)$$

Значения t_{AKA2} и t_{AKPm} заносятся соответственно в массивы активизации {ACTAGR} и {ACTPROC}. В этих массивах все агрегаты и процессы упорядочены по величине возрастания моментов их активизации (t_{AKA2} и t_{AKPm}). После выполнения операторов синхронизации УПМ проверяет условие: список (JAG) агрегатов. Активизируемых в момент t₀ (JAG), пуст? Если имеются номера агрегатов в этом списке. Выбор из списка JAG очередной по приоритету номер первого типа (ij) и затем по таблице агрегатов первого типа определяется адрес начала активности a_{ij} и передает управление на выполнение этой активности AKTA_{ij}. Если в списке JAG нет номеров агрегатов первого типа, выбирается из списка JAG очередной по приоритету номер агрегата второго типа (j) и затем по таблице агрегатов второго типа определяется адрес начала активности β_j и управление передается на выполнение активности AKT_j.

Если список JAG пуст, то проверяется список JPR номеров активизируемых процессов пустой. Когда в списке JPR имеются номера процессов (m), которые нужно активизировать в момент времени t₀, управление передается на программу выбора из списка JPR очередного активизируемого процесса (m). Далее по этому номеру m ПП. Начала активностей процессов по таблице процессов определяет адрес начала активности γ_m и управление передается на выполнение активности PROC_m.

В тех случаях, когда список JPR пуст, проверяется условие: условие окончания моделирования выполнено. Если моделирование нужно продолжить, осуществляется смена модельного времени по формуле:

$$t' = \min \{t_{JPR}, t_{JAG}\}, \quad (4.6)$$

где t_{JAC} и t_{JAS} соответственно минимальные значения времен активизации соответственно процессов и агрегатов.

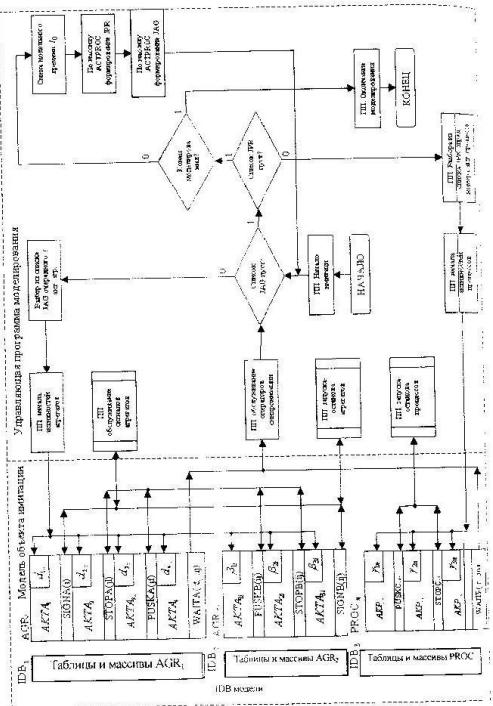


Рис. 18 - Блок схема организации агрегатно-процессного способа имитации СС

Далее с помощью ИП формирования списка процессов с помощью массива ACSTRPROC составляется новый список JPR, к который входит множество номеров (m_1) процессов у которых

$$T_{\text{AK}} \tau_{\text{PPM}}^{-1/2} f_{\text{ext}}' \quad (4.7)$$

Затем с помощью ПП формирования списка аргументов с помощью массива ACTAGR составляется новый список JAG, в который входит множество всех номеров $\{ij\}$, у которых $TAKAGR[i]=1$. Далее управление в УПМ передается на оператор проверки условия «список JAG пустой».

В случае окончания моделирования управления передается на полпрограмму окончания моделирования и имитации завершается. В самом начале имитации подпрограмма «начало имитации» устанавливает начальные значения в таблицах и массивах имитационной модели и передается управление на проверку условия «пустой ли список JAG».

ЛИТЕРАТУРА

- 1 Максимей, И.В. Имитационное моделирование на ЭВМ / И. В. Максимей. – М.: Радио и связь, 1989.
- 2 Максимей, И.В. Модели задач исследования операций : В 3 частях. Часть 3. Технология ИЭ и принятие решений / И. В. Максимей, С. И. Жоголь, С. П. Жоголь, В. Д. Левчук [и др.]. – Гомель: БелГУТ-ГГУ, 1999.
- 3 Айвазян, С.А. Прикладная статистика. Основы моделирования и первичная обработка данных / С. А. Айвазян, И. С. Еноков, Л. Д. Мешалкин. – С. П. Жоголь, В. Д. Левчук [и др.]. – Гомель: БелГУТ-ГГУ, 1999.
- 4 Айвазян, С.А. Прикладная статистика. Исследование зависимостей / С. А. Айвазян, И. С. Еноков, Л. Д. Мешалкин. – М.: Финансы и статистика, 1985.
- 5 Айвазян, С.А. Прикладная статистика. Классификация и снижение размерности / С. А. Айвазян, И. С. Еноков, Л. Д. Мешалкин. – М.: Финансы и статистика, 1989.
- 6 Афффи, А. Статистический анализ. Подход с использованием ЭВМ / А. Афффи, С. Эйзен. – М.: Мир, 1982.
- 7 Мелник, М. Основы прикладной статистики / М. Мелник. – М.: Энергоатомиздат, 1983.
- 8 Мандель, И.Д. Кластерный анализ / И. Д. Мандель. – М.: Финансы и статистика, 1982.
- 9 Растигин, Л.А. Коллектив решений правил / Л. А. Растигин, Р. Х. Эренштейн. – М.: Радио и связь, 1983.
- 10 Горев, А. Эффективная работа с СУБД / А. Горев, Р. Ахаян, С. Машаририов. – СПб.: Питер, 1997.
- 11 Вейскас, Д. Эффективная работа с Microsoft Access 2 / Д. Вейскас. – СПб.: Питер, 1997.
- 12 Грабер, М. Введение в SQL / М. Грабер. – М.: Изд-во «Лори», 1996.
- 13 Армстронг, Т. ActiveX: создание Web-приложений / Т. Армстронг. – Киев: BHV, 1998.
- 14 Волц, А. Основы программирования на Java для Word Wide Web / А. Волц. – Киев: Диалектика, 1996.
- 15 Ломакс, П. Изучаем VBScript / П. Ломакс. – Киев: BHV, 1998.
- 16 Хиллер, С. Microsoft Visual Basic, Scripting Edition в действии / С. Хиллер. – СПб.: Питер, 1997.
- 17 Архитектура среды для разработки приложений. – Киев: Диалектика, 1992.
- 18 Буч, Г. Объектно-ориентированное проектирование с примерами применения / Г. Буч; пер. с англ. – М.: Конкорд, 1992.
- 19 Задков, В.И. Компьютер в эксперименте : Архитектура и программные средства систем автоматизации / В. И. Задков, Ю. В. Пономарев. – М.: Наука, 1988.
- 20 Нортон, П. Персональный компьютер изнутри / Питер Нортон, Корри Сандлер, Том Баджет; пер. с англ. – М.: БИНОМ, 1995.
- 21 Аладьев, В.З. Компьютерная хрестоматия. Программное обеспечение персональных компьютеров : справочное руководство. – Киев: Думка, 1993.
- 22 Turbo Vision для языка Pascal. Описание. – М.: И.В.К.-Софт, 1992.
- 23 Климов, Ю.С. Программирование с среде Turbo Pascal 6.0 / Ю. С. Климов, А. И. Касаткин, С. М. Мороз. – Мн.: Вышэйшая школа, 1992.
- 24 Лебедев, А.Н. Моделирование в научно-технических исследованиях / А. Н. Лебедев. – М.: Радио и связь, 1989.
- 25 Объектно-ориентированное программирование с использованием C++ / пер. с англ. – Киев: НИПФ ДиаСофЛтд, 1995.
- 26 Фигурнов, В. Э. IBM PC для пользователя / В. Э. Фигурнов. – СПб.: АО «Корун», НПО «Информатика и компьютеры», 1994.
- 27 Толковый словарь по искусственному интеллекту. – М.: Радио и связь, 1992.
- 28 Техническое обеспечение компьютерных сетей / под ред. А. Н. Морозевича. – Мн.: УЦПНК АНБ, 1996.

100-02-02
"Гомельский национальный университет"
им Франциска Скорины"

БІБЛІЯГЭКА

Учебное издание

Максимей Иван Васильевич

**ОСНОВЫ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ**

Тексты лекций
по разделу «Технология имитационного
эксперимента на ЭВМ»
для аспирантов и магистрантов

В авторской редакции

Подписано в печать 07.06.2007 г. (45). Формат 60x84 1/16 Бумага
писчая № 1. Гарнитура Таймс. Усл. п.л. 6,1. Уч.- изд. л. 4,7. Ти-
раж 25 экз.

Б/Ч

Отпечатано с оригинала-макета на ризографе
учреждения образования
«Гомельский государственный университет
имени Франциска Скорины»
246 019, г. Гомель, ул. Советская, 104

108

РЕПОЗИТОРИЙ ГУИМЕНИ Ф. СКОРИНЫ